Advocating the use of Imprecisely Observed Data in Genetic Fuzzy Systems

Luciano Sánchez  
Computer Science Department  
Universidad de Oviedo  
Campus de Viesques  
33271 Gijón, Asturias, Spain  
E-mail: luciano@uniovi.es

Inés Couso  
Statistics Department  
Universidad de Oviedo  
Campus de Viesques  
33271 Gijón, Asturias, Spain  
E-mail: couso@pinon.ccu.uniovi.es

Abstract—To our opinion, and in accordance with the current literature, the precise contribution of Genetic Fuzzy Systems to the corpus of the machine learning theory has not been clearly stated yet. In particular, we question the existence of a set of problems for which the use of fuzzy rules, in combination with genetic algorithms, produces classifiers inherently better than those arising from the bayesian point of view.

We will show that this set of problems actually exists (and comprises intervalar and fuzzy valued datasets) but it is not being exploited. Current GFSs deal with crisp classification problems, where the role of fuzzy sets is reduced to give a parametric definition of a set of discriminant functions, with a convenient statistical test. Provided that the customary use of fuzzy set is to reduce the cost function is the mean number of errors. This rule is called "minimum error Bayes rule".

II. STATISTICAL CLASSIFIERS AND GFSs

A. Classification systems based on discriminant functions

Let us suppose we have a set \( \Omega \) that contains objects \( \omega \), and let us admit also that each one of them is assigned to a class \( c_i \), \( i = 1 \ldots N_c \). We are given a set of measurements \( X(\omega) = (X_1(\omega), \ldots, X_N(\omega)) \) over every object. 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.

For example, let \( \Omega \) be a set of fruits; apples \( (c_1) \), pears \( (c_2) \) or bananas \( (c_3) \). We observe the weight and the colour of a randomly selected fruit, for example, \( X(\omega) = (\text{yellow}, 150) \).

Our classification system relates the pair \( (\text{yellow}, 150) \) to the class \( c_3 \), and we wish this relation to be true most of times (i.e., most of the yellow fruits that weight 150 g. are bananas).

Since we do 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 can not be defined for this problem. But an optimum classifier can be defined with respect to the average number of errors.

To define the concept “average number of errors” we need to assume that the mapping \( X \) fulfills all necessary conditions to be a random variable. Let us also define a new random variable that quantifies the cost of assigning the class \( c_i \) to an object when it belongs to class \( c_j \), cost\( (i, j) \). If we choose 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”.

For the problem stated, if the classifier is a decision rule, \( D(X) \), and “class\( (\omega) \)” is the class of the object \( \omega \), then the merit value of a classifier can be numerically quantified as

\[
\text{err}(D) = \int_\Omega \text{cost}(D(X(\omega)), \text{class(\omega)}) \, dP
\]
where the error function is integrated with respect to a probability measure \( P \) defined over \( \Omega \). It is well known that this error is optimized by a classifier defined as follows: [16]

\[
D(x) = \arg \max_{i=1,...,N_c} P(\text{class}(\omega) = c_i | X = x). \tag{2}
\]

In practical designs, a monotonic transformation \( M(\cdot) \) of the conditional probabilities \( P(c_i | X) \) does not alter the classification but simplifies computations. We define \( N_c \) functions \( g_i(X) = M(P(c_i | X)) \), taking as the decision rule that \( X \) belongs to class \( i \) for which \( g_i(X) \) is maximum. \( g \)'s are called "discriminant functions" and the optimal classifier is written

\[
D(x) = \arg \max_{i=1,...,N_c} g_i(x). \tag{3}
\]

B. GFS should be learnt and evaluated with fuzzy data

It is important to note that this last approach is followed, up to our knowledge, by all Genetic Fuzzy Systems [6]. The random nature of the problem is clearly assumed by all GFS’s authors, because current standard experimental designs (leave one out, cross validation, etc.) are unbiased estimations of the classification error over the whole population \( \Omega \) (see eq. 1), and therefore the optimal classifier, no matter the learning technique, is defined by eq. 2, or by one of its transformations, as defined in eq. 3. Moreover, when an input is applied to a fuzzy rule base, the inference process eventually computes \( N_c \) truth values [11] or \( N_c \) number of votes [12] for the set of assertions "the input matches class \( c_i \)" and the defuzzification, in classification problems, consists in choosing the class maximizing the corresponding set of votes. This process is not different from the depicted in eq. 3.

As a consequence of this, the term "fuzzy" does not mean in GFSs that a classification problem different from the crisp one is being solved. "Fuzzy" means here that the parameterising functions has a linguistic interpretation compatible with the fuzzy logic postulates. This does not mean that a fuzzy classifier cannot be fed with fuzzy data; obviously, it can. We mean that neither learning algorithms nor statistical tests take into account the fuzzy nature of the output of the classifier. For example: we know that a random piece of fruit is yellow and weights "about 150 g". Now imagine that we want to compare two classifiers, A and B. Classifier A outputs "pear" with confidence 0.1 and "apple" with confidence 0.2. Classifier B outputs confidences 0.8 and 0.9. Which one is better? Under all statistical tests we are aware of, the two are assigned the same error, because they both eventually will classify the fruit as being an apple.

The experimental designs of GFS that are focused on imprecisely observed data are not being actively studied by the GFS community. Contrary to this, and according to the fuzzy statistics community, the customary use of fuzzy sets in classification and regression problems is the treatment of vague data [10], [15]. We think that this last point should not be understressed. If we admitted that the classification problem being solved by GFSs is not different to the crisp one, it may follow that

- GFSs are not a different machine learning technique than bayesian classifiers. They are a numerical method able to obtain discriminant functions with intuitive meanings.
- There are no reasons different from linguistic interpreatability that favour fuzzy rule based classifiers. Therefore, the usefulness of approximate classifiers [5], where linguistic concerns are secondary, is compromised.

By the contrary, if imprecise datasets were used to train and test GFSs, and specific statistical tests were devised to compare classifiers with fuzzy outputs, we would be able to compare different fuzzy classifiers over the set of problems where we expect that fuzzy classifiers make a difference over crisp classifiers, namely datasets with intervalar or fuzzy valued data.

III. AN EXTENDED DEFINITION OF THE CLASSIFICATION PROBLEM

We will consider that a fuzzy valued dataset is a sample of a fuzzy random variable, as defined in [13], whose \( \alpha \)-cuts are random sets. We will extend first the definition of the classification problem to the intervalar case, and then apply the results to all cuts of the fuzzy random variable sample.

A. Intervalar data

Recall eq. 2: to succeed, a learning algorithm should be able to estimate the values \( P(\text{class}(x) | x) \) from a sample of measures taken over a subset of \( \Omega \). To simplify the notation, in this section we assume that \( X \) takes values in \( \mathbb{R} \). When \( X \) has absolutely continuous distribution, the standard technique consists in making a transform

\[
D(x) = \arg \max_{i=1,...,N_c} \frac{f(x|c_i)P(c_i)}{f(x)}. \tag{4}
\]

where \( f \) is the density function induced by the random variable \( X \). The denominator can be removed without affecting the result:

\[
D(x) = \arg \max_{i=1,...,N_c} f(x|c_i)P(c_i) \tag{5}
\]

and we obtain a well known result: from an statistical point of view, learning a classifier is the same problem as estimating a density function from a sample of a random variable.

Now we are presented with a sample from a random set, and need to know how can we estimate the density function of the underlying, imprecisely observed random variable (so called original random variable in random sets literature [19]). Rephrasining the problem, we need to generalise the concept of density function to the random set case. Our primary thought was to define an "upper" density function as

\[
f^*(x) := \lim_{h \to 0} \frac{P^*((x-h,x+h))}{h},
\]

provided that this limit exists. For instance, if the random set \( \Gamma : \Omega \to P(\mathbb{R}) \) is a random interval of the form \( \Gamma(\omega) = [X(\omega) - \epsilon, X(\omega)] \), \( \forall \omega \), where \( X \) is a random variable with absolutely continuous distribution, this limit exists almost everywhere, but it is \( \infty \). Observe that \( P^*((x-h,x+h)) = \]
is the optimistic estimation of the error, where (cost is a pessimistic estimation of the error). For each class $c_i$, the probability we have that
\[
P(c_i \mid X = x) = \frac{P(c_i \cap \{\omega \in \Omega \mid X(\omega) \in (x-h, x+h)\})}{P(\{\omega \in \Omega \mid X(\omega) \in (x-h, x+h)\})}.
\]

The denominator is, again, the same for all classes, therefore we only need to compare the numerators for the different classes, and give lower and upper bounds for the value $P(c_i \mid X \in (x-h, x+h))$. Following [1], the limit when $h$ tends to 0 of these quantities is the value we need, $P(c_i \mid X = x)$. Applying the definition of conditional probability, we have that
\[
P(c_i \cap \{\omega \in \Omega \mid X(\omega) \in (x-h, x+h)\})
\]
and
\[
P(\{\omega \in \Omega \mid X(\omega) \in (x-h, x+h)\})
\]

Since we do not know the value of $P_i$, but a set that contains it, it is clear that, unless the intervals $[P_i, \bar{P}_i]$ do not overlap, we can not know if $P_i > P_j$ for all pairs of classes, thus the decision rule $D$ is not completely defined. This is graphically illustrated in Figure 1: the decision rule $D$ is not longer a point function, but a set valued function, where
\[
\mathcal{D}(x) = \{i \mid \bar{P}_j < P_i\}
\]
and $h$ is assigned a value small enough for the problem.

Given that $\mathcal{D}$ is a set valued function, the average error of the classifier is no longer known (or, alternatively, we could say that the average error is a set valued statistic.) Anyway, we can find upper and lower bounds for it (see eq. 1). Let us define a pair of functions
\[
\text{cost}(\omega) = 0 \text{ if } \text{class}(\omega) \in \mathcal{D}(\Gamma(\omega)), 1 \text{ else}
\]
\[
\overline{\text{cost}}(\omega) = 0 \text{ if } \mathcal{D}(\Gamma(\omega)) = \{\text{class}(\omega)\}, 1 \text{ else}.
\]

In words, cost is the optimistic estimation of the error, where we admit that an object is correctly classified if its class number is included in the output, and $\overline{\text{cost}}$ is a pessimistic estimation, where we suppose that an object is misclassified unless its class number is the only output of the classifier. Therefore, the average classification error is contained in the interval
\[
\text{err}(\mathcal{D}) = \left[ \int_{\Omega} \text{cost}(\omega) \, d\Omega, \int_{\Omega} \overline{\text{cost}}(\omega) \, d\Omega \right]
\]

\[
\mathcal{D}(x) = \{i \mid \bar{P}_j < P_i\}
\]

$\alpha$-cuts of $\tilde{\Gamma}$ are random sets (for example, the 0.5-cut of the value ‘high’ can be the interval [100,160]). Therefore, for every value of $\alpha$ we can build an interval classifier, as shown in the preceding section, whose output is a discrete set of class labels (“if the weight is [100,160], then the object is compatible with both pear and apple”). It is intuitive to conclude that the output of the classifier, if presented a fuzzy input, will be a fuzzy set defined over the set of class labels (“if the weight is high, then the object is 0.1/apple+0.6/pear”). The same can be said about the average error of the fuzzy classifier; it will be a fuzzy set.

To obtain this last value, it suffices to admit that the best description we can make about the probability $P(c_i \cap \{\omega \in \Omega \mid X(\omega) \in (x-h, x+h)\})$, given that the original random variable $X$ is contained in the fuzzy random variable $\tilde{\Gamma}$, is a fuzzy set $\tilde{P}_i$, whose $\alpha$-cuts are intervals $[P^{\alpha}_i, \bar{P}^{\alpha}_i]$ defined as follows:
\[
P^{\alpha}_i = P(c_i \cap \{\omega \in \Omega \mid \tilde{\Gamma}(\omega)_\alpha \subseteq (x-h, x+h)\})
\]
Therefore, the fuzzy output of the classifier will be the set

\[ \tilde{D}(x) = \{ i \mid \tilde{f}(j) \neq i \text{ with } [P_i]^a > [P_j]^a \} \]

and its average error is another fuzzy set,

\[ [\text{err}(D)]_\alpha = \left[ \int_{[\Omega]} \cos_{\alpha}(\omega) \, dP, \int_{[\Omega]} -\cos_{\alpha}(\omega) \, dP \right] \]

where

\[ \cos_{\alpha}(\omega) = 0 \text{ if class}(\omega) \in [D(\Gamma(\omega))]_\alpha, 1 \text{ else} \]

\[ \cos_{\alpha}(\omega) = 0 \text{ if } [D(\Gamma(\omega))]_\alpha = \{ \text{class}(\omega) \}, 1 \text{ else}. \]

C. Computer-friendly definition

In the preceding subsection we have stated that the average error of a classifier, when its input comprises fuzzy sets, should also be a fuzzy set. Therefore, the fitness functions in GFSS will return a fuzzy value. This value can be numerically estimated by means of eqs. 16, 17 and 18. Since these equations are expressed in terms of a family of \( \alpha \)-cuts, we give a rewriting of them that is easier to codify in a computer.

Let \( \tilde{D} = \sum_{i=1}^{N} h_i / i \) be the fuzzy output of the classifier, \( p \) the class label with maximum membership value in \( \tilde{D} \), \( p = \arg \max_{i=1,...,N}(\mu_i) \), and \( q \) the second maximum membership, \( q = \arg \max_{i=1,...,N,q \neq p}(\mu_i) \). Let the height of \( \tilde{D} \), \( \mu_p = 1 \). Then, the contribution of the object \( \omega \) to the total error is

\[ \text{err}(\omega) = \begin{cases} 1/0 + \mu_q / 1 & \text{if class}(\omega) = p \\ \mu_{\text{class}(\omega)}/0 + 1/1 & \text{otherwise} \end{cases} \]

For example, suppose that, in a problem with three classes, the output of the classifier is the fuzzy set \{0.2/pear+1/apple+0.8/banana\}. If the object was a pear, the accumulated error of the classifier would be increased by the fuzzy amount \{0.2/0+1/1\}. If it was an apple, the new error would be \{1/0+0.8/1\} higher, or \{0.8/0+1/1\} if it was a banana.

IV. EXAMPLE OF FITNESS EVALUATION IN THE EXTENDED CLASSIFIER

In this section we will numerically evaluate a fuzzy classifier over a small problem, to illustrate the ideas introduced in the preceding section.

Let us suppose that we have to discriminate between three classes (apple, pear, banana), given the weight of a piece of fruit. To design the classifier, we are given a sample comprising five pieces, whose weights and classes are given in table I. Weights are triangular fuzzy numbers, designated by three numbers: leftmost, center and rightmost values.

Let us also suppose that the GFS has to evaluate the fitness of the rule base that follows:

- if weight is small then banana
- if weight is medium then apple
- if weight is high then pear

Therefore, the cost of this classifier is 2 (in other words, we estimate that it is wrong 40% of times).

If we apply an interval input to the same classifier (the support of the fuzzy examples,) its output is a crisp subset of the class labels. For example, the interval [88, 112] has associated the crisp subset \( D(\Gamma(x)) = D([88, 112]) = \{ \text{banana}, \text{apple} \} \), because, if we classify all the points in [88, 112], we observe that points in [88, 92.5] are assigned the class “banana”, and points in [92.5, 112] are assigned the class “apple.” To calculate the cost of the classifier we operate as follows:
error still is between 20% and 80%, but a genetic algorithm with confidence 0.5625 and 80% with confidence 0.15. The is 20% with confidence 0.875, 40% with confidence 1, 60%
numbers, and the data (i.e., when data is intervalar, all we can say without assuming a random distribution of the observation error is that it is wrong between 20% and 80% of times.)

Finally, if the classifier is applied a fuzzy input, its outputs and costs are shown in table II. The inputs are fuzzy triangular numbers, and the data \((x, y, z)\) are left, center and right point. The cost of the classifier is \(\{1/1\} \oplus \{1/0 + 0.5625/1\} \oplus \{0.875/0+1/1\} \oplus \{1/0\} \oplus \{0.875/0+0.15/1\} = \{0.875/1+1/2+0.5625/3 + 0.15/4\}\), or in words, the error of the classifier is 20% with confidence 0.875, 40% with confidence 1, 60% with confidence 0.5625 and 80% with confidence 0.15. The error still is between 20% and 80%, but a genetic algorithm could prefer this result over a different classifier that has, say \(\{0.875/1 + 1/2 + 0.5625/3 + 0.25/4\}\), even in the punctual estimations of the classification error of either are the same, because the differences in the fuzzy errors state that the former classifier is less affected by imprecision in the input data (it is less likely to obtain an 80% error.) Observe also that, if input data are fuzzy triangular fuzzy sets, the punctual error of the classifier is given by the value with membership 1 in the fuzzy cost, 2 (or \(\%40\) of errors) in this example.

It is remarked that the algorithm used here to calculate the output, and the error of the classifier, given a fuzzy input, does not produce the same results that we would have obtained by means of the direct use of fuzzy inference. For example, if we apply max-min inference to compute the output of the rule base when its input is the fuzzy set \(\{0.4/banana+0.9048/apple+0.3095/pear\}\). By the contrary, the procedure proposed in this paper produces the set \(\{0.15/banana+1/apple+0.0294/pear\}\). In other words, we have proposed to use fuzzy logic to assign a class to a crisp input, but a fuzzy statistics-based interpretation of the observation error to extend the classification to imprecise data.

V. CONCLUDING REMARKS AND OPEN PROBLEMS

We have suggested in this paper that the kind of problems where GFSs are inherently better than their stochastic counterparts is composed by those problems including imprecisely observed data. To deal with imprecise data, stochastic algorithms have to introduce additional hypotheses, as a probability distribution (uniform, gaussian, etc.) over the measurement errors, that fuzzy algorithms do not need.

When the classification problem is extended to fuzzy data, there are some changes that have to be done in the numerical algorithms. We enumerate these changes and briefly comment on the problems that these modifications leave open:

1) Since the error of a classifier will not be a scalar, but a fuzzy set, the fitness function in the extended GFS must return a fuzzy value. Genetic Algorithms must do fitness-based orderings of individuals, therefore we must a) use a fuzzy ranking [3] to induce a total order over fuzzy parts of \([0, 1]\) or b) induce only a partial ordering and use multicriteria genetic algorithms instead [14]. The selection of the best fuzzy ranking, or, from a more general point of view, processing fuzzy values when evaluating the fitness function in genetic algorithms, is a problem that can not yet be considered as generally solved [18].

2) New statistical tests have to be designed in order to judge the relevance of the differences of two fuzzy algorithms. There are some works in fuzzy statistical inference [4], [8], [9], but more work needs to be done in order to do practical comparisons between fuzzy valued algorithms. It is not clear yet how the comparison between fuzzy and crisp data should be done (and we need that, to compare extended GFSs to other algorithms,) and this poses other open problems: the definition of statistical contrasts about fuzzy-valued parameters in fuzzy random variables, the very definition of parametric families of random sets or fuzzy random variables, and the design of their corresponding tests.

3) The benchmarks most commonly used to compare GFSs include missing values, that can be codified with fuzzy information, and linguistic data, that can also be assimilated to fuzzy sets, but we lack datasets of imprecisely measured data that allow us to compare the robustness of GFSs to that of stochastic methods in terms of the degree of imprecision in the data. This absence prevent us from optimizing GFSs towards the main objective of fuzzy techniques, as stated by Zadeh [21]: “Exploit the tolerance for imprecision [...] to achieve tractability.
robustness, and low solution cost”.

ACKNOWLEDGMENT

This work was supported in part by the Spanish Ministry of Education and Science, under grants TIC2002-04036-C05-05 and MTM2004-01269.

REFERENCES