From Approximative to Descriptive Fuzzy Classifiers

Javier G. Marín-Blázquez, *Student Member, IEEE,* and Qiang Shen

***Abstract*—This paper presents an effective and efficient approach for translating fuzzy classification rules that use approximative sets to rules that use descriptive sets and linguistic hedges of predefined meaning. It works by first generating rules that use approximative sets from training data, and then translating the resulting approximative rules into descriptive ones. Hedges that are useful for supporting such translations are provided. The translated rules are functionally equivalent to the original approximative ones, or a close equivalent given search time restrictions, while reflecting their underlying preconceived meaning. Thus, fuzzy descriptive classifiers can be obtained by taking advantage of any existing approach to approximative modeling, which is generally efficient and accurate, while employing rules that are comprehensible to human users. Experimental results are provided and comparisons to alternative approaches given.**

***Index Terms*—Approximate modeling, descriptive modeling, functionally equivalent translation, fuzzy classification rules, linguistic hedges.**

I. Introduction

IN APPLICATIONS such as systems monitoring and medical diagnosis, domain attributes often emerge from an elusive vagueness, a readjustment to context or an effect of human imprecision. The use of the soft boundaries of fuzzy sets, namely the graded memberships, allows subjective knowledge to be incorporated in describing these attributes and their relationships. Fuzzy techniques have proven to be very successful for creating, for example, robust controllers and user-friendly classifiers [1]–[3] to address such problems. Even when precise knowledge is available, fuzziness may be a concomitant of complexity involved in the reasoning process. Among the interesting features of fuzzy approaches is the potential of fuzzy production rules in attaching meaningful labels to the fuzzy sets [4], thereby allowing a human comprehensible representation of the system under consideration.

Fuzzy rule bases are typically assumed to be given by domain experts. However, the acquisition of knowledge on rule structures often forms the bottleneck to advance the success of fuzzy systems in practice [5], though the linguistic labels or fuzzy sets that are used within the rules may be subjectively defined. For many applications, there exists a considerable volume of historical data obtained by observing the behavior of the system concerned. Therefore, it is desirable to be able to automatically generate rules from given data. Many techniques exist for this, most of which follow the so-called approximative approach,[1] which works by creating and tuning the fuzzy rule bases to best fit the data. The rules generated are not encoded to keep the meaning of the linguistic labels of the fuzzy sets used. Such an approach is, under minor restrictions, functionally equivalent to neural networks [6], and the resulting systems offer little explanatory power over their inferences.

Opposing approximative modeling stands the descriptive approach, in which model transparency is as important as accuracy. Prescribed fuzzy sets are either not allowed to be modified or, at most, are permitted to have very slight modifications. The descriptive sets used, i.e., the fuzzy sets defined by humans with a preconceived linguistic label attached, induce a fuzzy grid in the product space of the domain variables. As little modification is permitted, the grid and the hyperboxes delimited by these sets are almost fixed. Often these hyperboxes may contain examples of different output states and, as they are fixed, there is no way to separate these outputs directly.

It is possible to implicitly modify fuzzy rule bases without disrupting the definition of the underlying fuzzy sets, by the use of linguistic hedges [4] which allow more freedom in manipulating the hyperboxes. Not all pure descriptive methods[2] support the addition of hedges though (e.g., the well established work as reported in [2]). When no hedges can be applied, an increase in the number of fuzzy sets, the addition of confidence factors, or prioritizing some data as critical, may increase the performance. However, these methods typically also give rise to a loss in the interpretability.

Approximative approaches avoid the problem of fixed hyperboxes by changing the definitions of the fuzzy sets and hence the hyperboxes themselves. This ruins the underlying prespecified meaning attached to the fuzzy labels which have a natural appeal to the commonsense understanding of the words used. This is particularly so when a free or weakly constrained modification of fuzzy sets is carried out, even for approaches such as the one reported in [7]. The current literature pays little attention to this side-effect of using an approximative model and focuses on the accuracy of derived models (some even regarding such models as descriptive or interpretable), or simply maintains a descriptive model and accepts high modeling errors.

Recent attempts have been made to regain some of the fuzzy systems' transparency [8]–[10], by reducing otherwise possibly many antecedent approximative sets into a manageable number that possess interesting properties [11]. However, the linguistic

Manuscript received October 9, 2001; revised November 7, 2001. The work of J. G. Marín-Blázquez was supported in part by the Fundación Marín-Blázquez, Spain.

The authors are with the Division of Informatics, The University of Edinburgh, Edinburgh EH8 9YL, U.K. (e-mail: javierg@dai.ed.ac.uk; qiangs@dai.ed.ac.uk).

Publisher Item Identifier 10.1109/TFUZZ.2002.800687.

[1]The word *approximative* is used here instead of approximate to mirror the word *descriptive* in descriptive modeling which is itself an approximate approach.

[2]Pure descriptive methods are those that do not allow any redefinition of the fuzzy sets used.

labeling is done, when possible, *a posteriori*; the labels are attached to fuzzy sets generated by an approximative method but not to those given by humans. As the fuzzy sets used are self clustered from training data and then given an artificial name, they may not have an intuitive interpretation. Human users of the resulting fuzzy systems have to make do with the "friendly" words produced by the computer rather than the other way around. Significant work has been proposed to obtain descriptive explanations of approximative models [12] where each approximative rule fired is translated from one approximative hyperbox to one closest descriptive hyperbox. This represents an important departure from pure approximative modeling approaches. However, it adds on additional runtime cost and the explanation generated may not be sufficiently accurate due to the one-to-one approximate translation.

This paper presents an alternative approach, based on the initial investigations as reported in [13] and [14], for producing descriptive fuzzy systems with a two-step mechanism. The first is to use an approximative method to create accurate rules and the second to convert the resulting approximative rules to descriptive ones. The conversion is, in general, one-to-many, implemented by a heuristic method that derives potentially useful translations and then by performing a fine tuning of these translations via evolutionary computation. Both steps are computationally efficient. The resultant descriptive system is ready to be directly applied for inference; no approximative rules are needed in runtime. Note that the work described here is focused on classification tasks.

The overall conversion process proposed is guided by *functional equivalence* rather than by similarity between approximative fuzzy sets and predefined descriptive ones in the antecedent part of the rules. To ensure a highly accurate translation, novel linguistic hedges are defined. The ultimate objective is to obtain a whole descriptive ruleset and to use it to perform the inference direct, thereby providing not only human comprehensible models but also straightforward explanation of the reasoning based upon the resulting models.

The rest of the paper is organized as follows. Section II describes the background and gives further detailed reasons for the present work. Section III proposes a set of useful linguistic hedges, which differ from those conventionally employed in the literature. Section IV presents the translation process, which maps approximative rules onto descriptive ones. It covers two methods, one based on the use of heuristics and the other on a genetic algorithm [15]. The latter is designed to use the result of the former as its initial population generator for efficiency purposes. Section V reports on typical experimental results, demonstrating the potential of the present research. The paper is concluded in Section VI, with further work pointed out.

II. Background and Justifications

The task of descriptive modeling is to find a finite set of descriptive rules capable of reproducing the input–output behavior of the system being considered. For classification problems, without losing generality, the system to be modeled is assumed to be a multiple-input–single-output (MISO) one. That is, a system of M inputs and one output that can be described by a set of K rules such as

$$R_i\text{: IF } x_1 \text{ is } D_i^1 \text{ AND} \ldots \text{AND } x_M \text{ is } D_i^M \text{ THEN } y \text{ is Class}_h \quad (1)$$

where R_i is the ith rule $(1 \le i \le K)$, x_j is the jth input variable $(1 \le j \le M)$, y is the output variable to be assigned to one of the possible output classes, and D_i^j are descriptive fuzzy sets for these variables. D_i^j can be either a single descriptive fuzzy set or a combination of one or two hedges and a descriptive fuzzy set. Note that more than two hedges per variable are allowed in theory. However, a joint use of more than two hedges often destroys the readability of the resulting descriptive rules and hence is not desirable to be employed in practice. Descriptive fuzzy sets are human defined and fixed throughout both the modeling and the inference processes.

This follows the general principle of supervised learning [16]. The only information about the behavior of the system under consideration is assumed to be a (usually large) set of input–output example pairs, where for each instantiation of the input variables an associated class is indicated

$$\begin{aligned}\Omega &= \{(x_{t1}, x_{t2}, \ldots, x_{tM}, y_t)\} \\ &= \{(x_t^M, y_t), \qquad t = 1, \ldots, N\}.\end{aligned} \quad (2)$$

The ruleset to be induced is required to approximate the function φ: $X^M \to \text{Class}Y$ (that theoretically underlies the system behavior) in the most consistently possible way with the given examples of input–output pairs. It is assumed that the collection of the data examples represents the system behavior in the product space $(X^M \times Y)$, where $X^p = (X_1 \times X_2 \times \cdots \times X_M)$, $X_1, X_2, \ldots, X_p$ are the domains of discourse of the inputs and Y is the domain of the output classes.

It is, in general, computationally prohibitive to perform exhaustive search in the space of all possible combinations of descriptive sets (with or without the use of linguistic hedges), in order to obtain descriptive rules. This is due to the "curse of dimensionality" [17], [18]. The higher the number of variables available, and/or the higher the cardinality of the fuzzy partition for each variable, the higher (exponentially) the number of possible rules. This is generally true for approximative models as well. In fact, the size of all possible rules associated to the general product ruleset R_{Set} of a given system, allowing up to two hedges per fuzzy set for example, may be represented by

$$\begin{aligned}|R_{\text{Set}}| &= H_{L_1 1} \times H_{L_1 2} \times L_1 \times \cdots \times H_{L_M 1} \times H_{L_M 2} \\ &\quad \times L_M \times S \\ &= h^{2\times M} \times s \times \prod_{i=1}^{p} k_i\end{aligned} \quad (3)$$

where L_i is the fixed set of labels for the ith variable in the input space, of a cardinality $|L_i| = k_i$; $H_{L_i j} \in H$, $j = 1, 2$, with H being the set of applicable hedges of a cardinality $|H| = h$; and S is the fixed set of class labels of the output space, of a cardinality $|S| = s$. The modeling task is to select the smallest possible subset of R_{Set} that characterizes it to a degree as high as possible. This value increases dramatically as input dimensionality increases. This makes it impossible to perform exhaustive search for any moderately sized problem. Even robust but nonexhaustive search techniques such as genetic algorithms

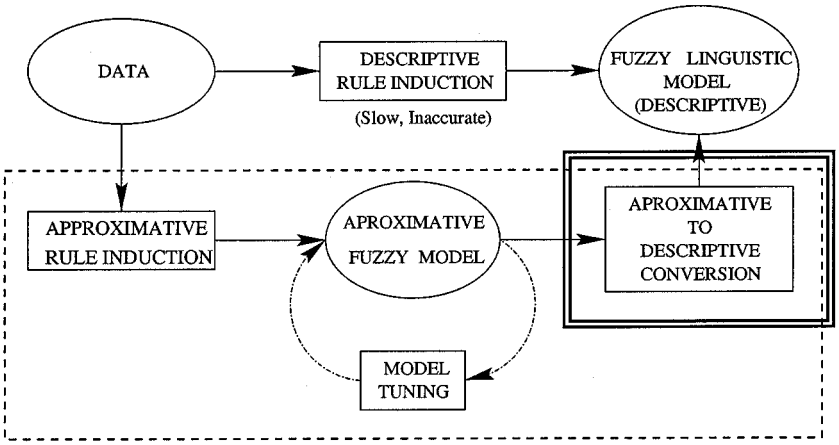

Fig. 1. Descriptive rule generation.

(GAs), may not perform well when $|R|$ is large [19]. This is mainly because many of the rules generated may not cover any data. The subset of interesting rules that at least cover some data may indeed be very small compared with the total; much effort of the search is often wasted in order to find that small subset. However, the search of interesting rules can be considerably reduced if it starts with a rough solution and then a GA is used to optimize this solution.

The question becomes how to find a rough solution efficiently. Fortunately, there already exist approximative rule generation techniques that are, focusing on given data, able to find very accurate rules, without using pure and brute force search. Being data driven [16], these techniques avoid the empty parts of the input space. There are no restrictions over the fuzzy sets they use, that is, these sets do not have to satisfy any prescribed linguistic interpretation. The resultant approximative rules "point" to places in the search space where desirable descriptive rules potentially exist. These approximative rules can then be transformed into descriptive ones with a heuristic method, which is used as the generator of the first population for a GA that will optimize the translation.

Fig. 1 shows the basic ideas of the present work. Instead of using a direct descriptive rule induction technique, which is generally rather slow and inaccurate, it is proposed to use a fast and accurate approximative rule induction algorithm first. (The approximative fuzzy model induced can be tuned to improve its modeling accuracy). The approximative model is then converted into a fuzzy linguistic model that utilizes predefined descriptive sets. It is this conversion process that forms the major work reported herein.

Through the use of approximative rules, a vast volume of the search is already done. The GA's effort is directed to perform fine adjustments. The emerging solution, i.e., a descriptive ruleset, can be improved due to the neighborhood search operators included in the GA or inserted between GA generations. In general, the better the initial translation the less effort the GA will have to apply, as the initial solution is already close to the final ruleset, usually far closer than a random one.

III. HEDGES

The application of a linguistic hedge modifies the shape of the membership function of a fuzzy set [20], transforming one fuzzy set into another. The meaning of the transformed set can easily be interpreted from the meaning of the original set and that embedded in the hedge applied.

The definition of hedges has more to do with common sense knowledge in a domain than with mathematical theory. Although a simple linguistic hedge may be used to express different modifications in different applications the general type of the underlying operation remains the same, varying only in terms of detailed parameter settings. For example, concentration/dilation hedges are often implemented by applying a power function to the original set membership values [20], [21]. That is, given the original membership function $\mu_S(x)$ of a fuzzy set S and hedge H, the membership function of $H \cdot S$ is $\mu_{H \cdot S}(x) = \mu_S^e(x)$, where the exponent e is greater than 1 for concentration and less than 1 for dilation. Different values can be assigned to the exponent e; for hedge EXTREMELY, for instance, $e = 2$ is used in [20], while $e = 8$ is employed in [21].

Conventional definitions of hedges do not result in significant changes on trapezoid fuzzy sets, which are most commonly used for computational simplicity purposes. In particular, the full membership part of a trapezoid membership function does not get changed at all. In this work, a different implementation of the hedges is considered, which may be applied to concentrate or dilute an original fuzzy set by shrinking or expanding any parts of the trapezoid. In addition, three new hedges named UPPER, LOWER, and MID that do not appear in the literature are also proposed.

A trapezoidal membership function, characterized by four parameters (a, b, c, d), consists of three consecutive segments as illustrated in Fig. 2. The application of concentration/dilation hedges should decrease/increase the size of these segments and, therefore, be implemented with the modification being in proportion to the center of the full membership segment (m). The following formalizes these ideas and defines the hedges used in this work.

A. Concentration

Concentration hedges reduce the size of segments or each part of the membership values of an original set. For a given trapezoidal fuzzy set S with a membership function $\mu_S(x)$, the set modified by a concentration hedge CON should comply with

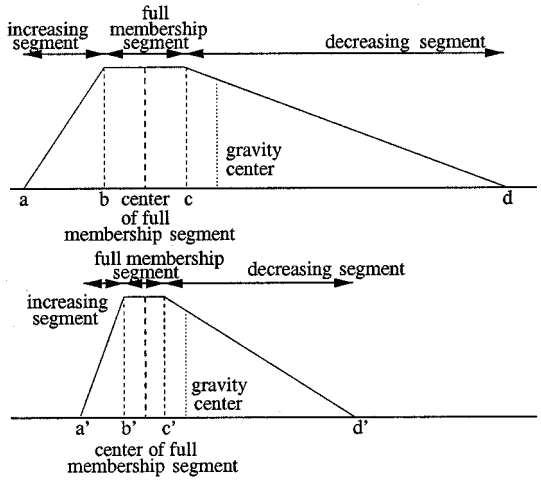

Fig. 2. Parts of a trapezoidal set and application of the hedge VERY.

$\forall\, x \in X,\ \mu_{CON\cdot S}(x) \le \mu(x)$. The parameters of the modified set are, therefore, defined by

$$\begin{aligned} m &= \frac{b+c}{2} \\ b' &= m - ((m-b)*\beta) \qquad c' = m + ((c-m)*\beta) \\ a' &= b' - ((b-a)*\beta) \qquad d' = c' + ((d-c)*\beta) \end{aligned} \tag{4}$$

where β controls the degree of shrinking. In order to reduce the set effectively β must satisfy $0 < \beta < 1$. In particular, the commonly used hedge terms MORE, VERY (see Fig. 2) and EXTREMELY can be defined as follows:

- MORE reduces the segments to 2/3 of the original size ($\beta = 2/3$);
- VERY reduces the segments by half ($\beta = 1/2$);
- EXTREMELY reduces the segments to 1/8 of the original size ($\beta = 1/8$).

B. Dilation

Dilation hedges increase the size of segments or each part of the membership values of a fuzzy set. As opposite to a concentration hedge, a dilation hedge DIL should comply with the following intuition: $\forall\, x \in X,\ \mu_{DIL\cdot S}(x) \ge \mu_S(x)$. The parameters of the modified set are calculated as concentration hedges, but this time the factors will be greater than one.

- GREATLY increases the segments by 2 times the original size ($\beta = 2$).
- LESS increases the segments by 3/2 of the original size ($\beta = 3/2$).

Note that the pair MORE and LESS, and the pair VERY and GREATLY are complementary; they cancel each other's effect as they express exactly opposite concentration–dilation concepts. No hedge was found in the literature that matches the opposite of EXTREMELY (perhaps REMOTELY could be a candidate for this), but including such a dilation hedge is as simple as setting $\beta = 8$.

C. Restriction

Restriction hedges [20], ABOVE and BELOW, are applicable to variables where fuzzy values are ordered. The set modified by applying the ABOVE hedge denotes the set which is "greater than" the original set and that by the BELOW hedge represents the set which is "less than" the original. Therefore, the resulting sets are shouldered ones, the left shouldered for BELOW and the right shouldered for ABOVE. Their membership functions are defined as follows:

$$\mu_{\text{ABOVE}\cdot S}(x) = \begin{cases} x < c, & 0 \\ c \le x < d, & 1 - \mu_S(x) \\ x \ge d, & 1 \end{cases}$$

$$\mu_{\text{BELOW}\cdot S}(x) = \begin{cases} x < a, & 1 \\ a \le x < b, & 1 - \mu_S(x) \\ x \ge b, & 0. \end{cases} \tag{5}$$

Note that the application of restriction hedges to some fuzzy sets may make no sense. This includes cases where the hedge ABOVE is to be applied to a right shouldered set [or any set whose $\mu(x) = 1$ when $x \to \infty$] and similar ones where BELOW is used to modify a left shouldered set. Such nonsense hedge-set combinations are disallowed, as they always return zero memberships and, hence, cause the rules which would otherwise involve them not to fire anyway.

D. Detailization

This proposed new type of hedge splits the original set into three, but keeps the order of these split sets the same as the order of the elements belonging to the full membership segment of the original. Therefore, these hedges will only make sense on variables whose values are ordered, as with the restriction hedges. The resulting three sets LOWER$\cdot S$, MID$\cdot S$ and UPPER$\cdot S$, arranged in an increasing order, are defined by

$$\text{LOWER}\cdot S \begin{cases} a' = a & b' = b \\ c' = b + \dfrac{b-c}{3} & d' = b + \dfrac{2*(b-c)}{3} \end{cases}$$

$$\text{MID}\cdot S \begin{cases} a' = b & b' = b + \dfrac{b-c}{3} \\ c' = b + \dfrac{2*(b-c)}{3} & d' = c \end{cases}$$

$$\text{UPPER}\cdot S \begin{cases} a' = b + \dfrac{b-c}{3} & b' = b + \dfrac{2*(b-c)}{3} \\ c' = c & d' = d. \end{cases} \tag{6}$$

As an example, if the original fuzzy set S is (0, 3, 12, 14) then LOWER$\cdot S$ is (0, 3, 6, 9), MID$\cdot S$ is (3, 6, 9, 12) and UPPER$\cdot S$ is (6, 9, 12, 14).

E. The NOT Operator

Although NOT is not a hedge but a logical operator, in terms of its application effects it may be viewed as a hedge for presentational convenience. This is because the application of this operator to a fuzzy set also changes the shape of the membership function of that set (as $\mu_{\text{NOT}\cdot S} = 1 - \mu_S$). For this reason, it will be treated similarly as any other hedge hereafter.

Finally, to have an overview of the new hedges introduced beforehand, Fig. 3 shows the results of applying them to a given irregular trapezoidal fuzzy set.

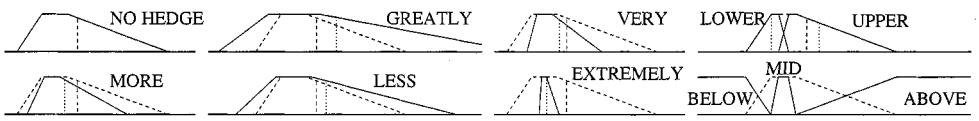

Fig. 3. Hedges applied to an irregular trapezoid and how they change the center of gravity.

IV. Mapping Approximative Onto Descriptive Rules

The main aim of this paper is to find an efficient and effective way to translate rules that use approximative sets into rules that use descriptive sets and hedges. The translated rules will be equivalent to the original, or a close equivalent within the limitations of the GA search, with the advantage of having or regaining human-comprehensible interpretation. For this, it does not matter as to which technique is used to generate the original approximative rules. What is required is a set of approximative rules and the definition of the descriptive fuzzy sets and linguistic hedges. In the experimental studies to be presented later, a hybrid method is used to produce the initial approximative rules. For the use of the proposed heuristic method (for generating the initial descriptive rules) there must also exist a similarity measure between the fuzzy sets used in the approximative rules, which can be of any type, and the descriptive trapezoidal fuzzy sets. Trapezoids are adopted as final descriptive sets for computational efficiency purposes.

To perform the mapping a concatenation of two methods is proposed here. One is based on a heuristic search. (As the space of potential descriptive rules can be very large, techniques of branch and bound are applied and so the power of this heuristic method may be rather restrictive). The other uses a GA to work on the full search space. As evolutionary search usually works better when a good start point has been identified [22], the first method will be employed as the generator of the initial population for the GA, which will then make a finer-grain search. The heuristic translation may not yield spectacular results but it is far better than a random start as will be shown later.

A. Heuristic Approach

This approach is based on hyperbox intersection with given approximative rules. Descriptive rules, i.e., hyperboxes defined on descriptive sets, are created if they intersect with an approximative rule (or the hyperbox defined by the antecedent approximative sets). This produces a preliminary translation, which is the one often used for explanation purposes in the existing literature [12]. The basic component of this proposed method uses no hedges and serves to introduce the underlying ideas of the heuristic translation. This has been extended to include the use of hedges, but only the basics are explained here with the extensions outlined for presentational simplicity.

This heuristic method works by building a layered graph to represent degrees of intersection between approximative and descriptive sets, using an intersection-based similarity measure. Each layer of the graph consists of a certain number of nodes; each of which represents the amount of intersection between one of the approximative sets of an antecedent variable and one of the descriptive sets of the same variable. Thus, each path of the resulting graph may be interpreted as a possible descriptive rule which coarsely approximates a given approximative one.

```
for i = 1 to p
  for j = 1 to |L_i|
    if I_{S_i L_ij} > I-threshold
       add(generate_node(i,j), graph)
    endif
  endfor
endfor
for i = 2 to p
  for j = 1 to |L_i|
    for k = 1 to |L_{i-1}|
      connect(node(i,j),node(i-1,k))
    endfor
  endfor
endfor
reset_find_path(graph,1,p)
while (more_paths(graph,1,p))
  add(generate_rule(get_next_path(graph,1,p),rulebase))
endwhile
```

Fig. 4. Graph generation algorithm.

The amount of similarity between two sets S_1 and S_2 is hereafter called the *similarity value* (I) of the two. In this paper, the similarity used is defined by

$$I_{S_1S_2} = \frac{A(S_1 \cap S_2)}{\text{Max}(A(S_2),\, A(S_1))} \tag{7}$$

with $A(Set)$ denoting the area of the set Set.

The similarity this value may vary from zero for no intersection to one for equality. When intersection is null the corresponding node is removed from the graph. Incidentally, although the above particular definition is utilized in this work, empirical results have shown that other similarity metrics proposed in the literature [23]–[25] may be adopted to take its place without major disruption in the mapping results. However, this definition has proven to be computationally simple and performance-wise robust.

Supported by such a similarity metric, given an approximative rule Q

$$\text{IF} x_1 \text{ is } S_1 \text{AND } x_2 \text{ is } S_2 \text{ AND } \ldots \text{ AND } x_p \text{ is } S_p \text{ THEN Class}$$

and a collection of descriptive sets $\{L_{ij} | j = 1, 2, \ldots, k_i\}$ per variable x_i the preliminary method to build the graph is summarized in Fig. 4.

To illustrate this basic approach consider the following example. Assume that the input space is two-dimensional. For each of the two input variables, x_1 and x_2, three descriptive fuzzy sets are defined such that x_1 may take a value on either $L_{11} =$ Low, $L_{12} =$ Medium, or $L_{13} =$ High, and x_2 on either $L_{21} =$ Small, $L_{22} =$ Medium, or $L_{23} =$ Large. Suppose that the approximative rule to be translated is

$$\text{IF} x_1 \text{ IS} S_1 \text{ AND } x_2 \text{IS } S_2 \text{THEN } A$$

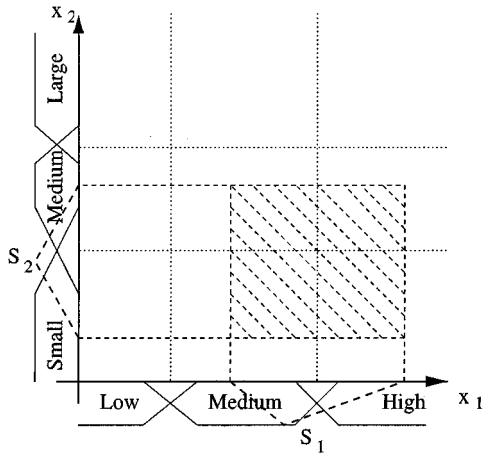

Fig. 5. Approximative rule and descriptive sets.

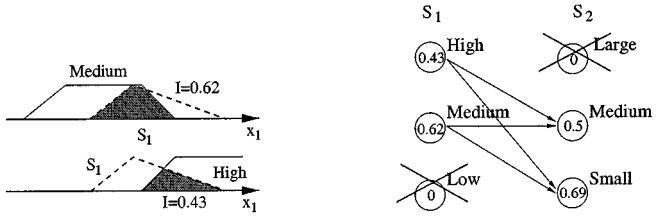

Fig. 6. Intersections of descriptive and approximative sets (left) and graph generation (right).

where S_1 and S_2 are two approximative fuzzy sets defined on the domains of x_1 and x_2 respectively and A is a possible class value. The descriptive sets, the grid generated by these sets, and the hyperbox covered by the approximate rule are given in Fig. 5.

The first layer of nodes is created by taking on the approximative set of the first antecedent of the original rule, in this case S_1, and then constructing a node for each descriptive set L_{1i}, $i = 1, 2, 3$ (that is, Low, Medium, and High) of x_1 if the similarity measure between L_{1i} and S_1 is larger than I-threshold (zero by default). Suppose that the measure between S_1 and L_{12} and that between S_1 and L_{13} are $I_{S_1L_{12}} = 0.62$ and $I_{S_1L_{13}} = 0.43$, respectively. Also, suppose that S_1 does not intersect L_{11}. Thus, two nodes are created, as illustrated under S_1 in Fig. 6.

This process is repeated for each variable that appears in the antecedent of the original approximative rule, resulting in different layers of nodes with each layer corresponding to one variable. Then, all the nodes in one layer are connected to the nodes in the next with the arrow of each link pointing from a previous node to a newly created one, as also shown in Fig. 6.

Once the graph is generated ,paths from any node in the first layer to a node in the last are constructed. Each path becomes an emerging rule, with the antecedent variables taking the labels of the nodes of that path. Thus, the resultant set of descriptive rules which collectively form a preliminary translation of the given approximative rule are

R_1: IF x_1 IS High AND x_2 IS Medium THEN A;
R_2: IF x_1 IS High AND x_1 IS Small THEN A;
R_3: IF x_1 IS Medium AND x_1 IS Medium THEN A;
R_4: IF x_1 IS Medium AND x_1 IS Small THEN A.

This heuristic method does not ensure a good coverage of an approximative rule unless the threshold used is very low. However, a low threshold potentially implies many nodes and hence many descriptive rules. This implies that the method can be quite sensitive to such parameters settings. Nevertheless, this method is proposed to act as the starting point for the evolutionary search and its accuracy is not of utmost importance. Also, it can be itself improved.

An obvious improvement is to include hedges. In so doing, the number of nodes will, however, increase drastically. This is because the label of a node may now be any combination of a descriptive set and a number of hedges used to modify the set. Even if nodes with a similarity value below the threshold are eliminated, and if the up ceiling of the number of hedges applicable to a set is limited to two, this may still result in a significant increase of the number of nodes in the graph.

To perform an extra reduction of the graph and hence the number of emerging descriptive rules, various heuristics may be applied to eliminate unwanted nodes. In particular, if some nodes within a layer are similar to each other only one of them would then be needed. Also, external control of the desirable distinctions amongst possible values per input variable, that is the number of nodes permitted per layer, can be used to select those which are most dissimilar between one another. Both methods are implemented in this paper; they ensure that the nodes left are different among themselves. Of course, these methods are assisted with the requirement that whatever nodes to be chosen they must attain a high similarity value.

B. GA-Based Approach

While the heuristic approach relies on the use of similarity, the evolutionary computation-based approach proposed here depends on the concept of *functional equivalence.* It works by searching for a set of descriptive rules that collectively behave like the original approximative rule from which they are translated. That is, for data that is covered by an approximative rule, the found descriptive rules will fire with at least the same firing strength as the original. Furthermore, for data that would not cause the original approximative rule to fire, the resultant descriptive rules will either not fire or fire if their consequents comply with the desired output. As indicated before, the search mechanism is herein implemented by a GA.

1) Training Sets and Objective Functions: Each approximative rule may be translated independently. Multiple descriptive rules are considered per approximative rule as, in general, an approximative rule R may not be covered by just one resulting descriptive rule. To implement the translation in this manner, a training subset for each approximative rule R needs to be generated from the original training set (from which the approximative rules were obtained). Such rule training subsets are derived via a data selection and enrichment process as introduced in the following.

Suppose that there are K_R emerging descriptive rules that, collectively, form the *functional equivalent* to a given approximative rule R. Given a set $\mathcal{X}$ of the original training examples, for each $x_i \in \mathcal{X}$ the firing strength $AFS_R(x_i)$ of the approximative rule under translation is calculated. If $AFS_R(x_i) > 0$, it would be desirable that, if the consequent of this rule is the same

as the desired consequent, any of the resulting translated descriptive rules R_j, $j = 1, \ldots, K_R$ will fire for such an example with a strength $DFS_{R_j}(x_i)$ equal to or greater than $AFS_R(x_i)$. This kind of example will be hereafter referred to as an example of *type one*. If, however, $AFS_R(x_i) > 0$ and the consequent does not match the desired, then the firing strength of each resulting descriptive rule $DFS_{R_j}(x_i)$ should be less than, or at worst equal to, $AFS_R(x_i)$. This kind of example will be referred to as *type two*. Furthermore, if $AFS_R(x_i) = 0$ then, if the example x_i is of the same desired consequent as that of the original rule, it is not selected to form the training subset (as this example provides no influence in executing this learned rule and is expected to be covered by other approximative rules). If, however, the consequent is different, the firing strength of the resulting descriptive rules should be zero. This last kind of example will be referred to as *type three*.

Clearly, for any $x_i \in \mathcal{X}$ and a given original approximative rule R, x_i is selected to form the training subset for translating R if and only if it is an example of either of the three types. Each data point is enriched by the inclusion of its type and its $AFS_R(x_i)$. This data selection process allows the GA to enforce the following objectives in performing search for suitable descriptive rules, where T_t, $t \in 1, 2, 3$, denotes the subset of training data of type t:

- $\forall\, x_i \in T_1 DFS_{R_j} \geq AFS_R(x_i)$
- $\forall\, x_i \in T_2 DFS_{R_j} \leq AFS_R(x_i)$
- $\forall\, x_i \in T_3 DFS_{R_j} = 0$.

As the classification inference is performed by choosing the output value of the rule that has the highest firing strength, enforcing the aforementioned conditions yields a descriptive model that is at least as accurate as the original approximative model. This is because the inequality restrictions allow an increase in the firing strength of the descriptive rules to be learned over correct training data (type one) and a reduction in the firing strength over incorrect training examples (type two). However, in general, not all training examples will satisfy these restrictions, and it is the job for the GA to reduce the discrepancies between the descriptive and approximative firing strengths as much as possible.

For efficiency, the GA should be guided to search for a set of emerging descriptive rules of a minimum cardinality. This means that an objective is needed to minimize the number of descriptive rules used to act as the given approximative rule. Also, any difference between the DFS of an emerging rule (within the resulting descriptive rule set) and the AFS of the original approximative rule for each data type should be restricted to be minimum. Hence, another objective is introduced to minimize the variance of individual rule error. This way, the error that may be produced by the translated rules will be as much evenly distributed among all rules as possible, thereby avoiding individual rules with a particularly high error.

In summary, in searching for a set of descriptive rules that would jointly function as a given original approximative rule R, the GA search will be guided by objectives as listed in Table I, where K_R is the current number of emerging descriptive rules. In this table E_{R_j} denotes the individual error of a descriptive rule, and δ_R represents the mean of the individual errors of all

TABLE I
OBJECTIVES

Functional Equivalence Objectives (Minimize)	
Expression	*Description*
$\sum_{x_i \in T_1, j=1,\ldots,K_R} max(0, AFS_R(x_i) - DFS_{R_j}(x_i))$	Error of type 1 training data
$\sum_{x_i \in T_2, j=1,\ldots,K_R} max(0, DFS_{R_j}(x_i) - AFS_R(x_i))$	Error of type 2 training data
$\sum_{x_i \in T_3, j=1,\ldots,K_R} DFS_{R_j}(x_i)$	Error of type 3 training data
$\frac{1}{K_r} \sum_{j=1,\ldots,K_R} (\delta_R - E_{R_j})^2$	Overall error variance
K_R	Number of rules
Additional Classification Objectives	
Maximize Number of Correctly Classified	
Minimize Number of Incorrectly Classified	
Minimize Number of Not Covered	

the emerging descriptive rules with regard to the original approximative rule R. They are defined as follows:

$$E_{R_j} = \sum_{x_i \in T_1} \max(0,\, AFS_R(x_i) - DFS_{R_j}(x_i)) + \sum_{x_i \in T_2} \max(0,\, DFS_{R_j}(x_i) - AFS_R(x_i)) + \sum_{x_i \in T_3} DFS_{R_j}(x_i) \tag{8}$$

$$\delta_R = \sum_{j=1,\ldots,K_R} \frac{E_{R_j}}{K_R}. \tag{9}$$

There exist in the GA literature several approaches to deal with such problems that have multiple objectives, including the aggregation approach [26], non-Pareto approach [27], and Pareto-based approach [26]. This paper adopts the first of these, as it offers a conceptually simpler method by converting multiple objectives into a compounded single objective. In particular the aggregation function used is the sum of weighted global ratios (SWGR) [28]. The aggregation method first independently normalizes each objective with respect to the best and worst value ever found for it and, then, weighs and adds together each objective to form the single overall fitness value.

The individual translation strategy described above has the drawback that, when the individual translations are put together to form the final translated descriptive rule set, the independently translated rules may interfere with each other. Although a close fit of the descriptive rules to the approximative ones may help resolve this problem, this cannot be guaranteed. Therefore, it is interesting to consider possible alternative translation strategies.

Instead of translating individual approximative rules one by one, the first possible alternative approach, valid for problems with a limited range of output values (such as classification problems as mainly concerned herein), is to translate one group of all the approximative rules regarding a single output value at a time. In this so-called group approach, an AFS value is calculated as the firing strength of the entire subset of rules concerning the same output value, which is defined by the strongest

firing strength of all the original approximative rules that characterize the same class. Thus the translation can be done class by class, instead of rule by rule.

For completeness, another version of the GA search strategy is also included here, and termed the global strategy, where all rules for all classes are represented together in each chromosome. That is, a chromosome is itself a whole translation of the given approximative model.[3]

Pure *functional equivalence* guidance (i.e., the exclusive use of only the objectives of Table I) may miss some otherwise possible improvements of the overall performance of the learned ruleset, because of its trying to fit the approximative model rather than to fit the training data. Empirically, for classification problems, it is generally better to use the influence of *functional equivalence* objectives along with the above classification objectives and to decrease *functional equivalence* influence as the GA goes on. That is, the search will initially have a strong focus on the improvement of the heuristic translation to get close to the approximative model and later it will concentrate on the satisfaction of the classification-specific objectives. The reason that search is not guided by the classification objectives alone right from the start is to speed up the finding of optimal descriptive ruleset, by first approximating the emerging descriptive rules to a potentially good accuracy level (offered by the good approximative model) and then optimizing them locally.

Finally, it is worth noting that no guarantees may be given to obtain the closest equivalent translation when a GA run terminates. In general, such a guarantee cannot be obtained without performing an exhaustive search. However, given limited computational resources, the translations are empirically very close to the original approximative model in function (as shown later).

2) Genetic Representation and Genetic Engine: In this paper, the genetic chromosome representation is based on the work as reported in [30]. Basic ideas of this representation and its associated inference mechanism are outlined below; the detailed codification is beyond the scope of this paper and can be found in [31].

The GA adopted here is a steady-state one. The selection of the two parents is done for one by linear ranking and for the other by random choice. Each child replaces a random member of the worst half of the population. The search stops when the best half of the population does not improve for a prescribed number of generations. Diversity is maintained thanks to the random replacement within the worst half of the population.

After a translation process terminates, those possible descriptive rules that did not fire with the training data available are eliminated. Note that there may exist cases where the eliminated rules cover certain training data, but such data is already covered by other rules with higher firing strength, so they did not ever fire. In experimental studies, to be reported next, not all available data is used for training in order to exploit part of the data to check for possible overfitting.

Three different mutation and four different crossover operators have been implemented in order to investigate what combinations may lead to a good translation.[4] The mutation rate and the rate at which a different crossover is used are both allowed to change dynamically. Empirically, the inclusion of this dynamic schema helps improve significantly the performance of the GA employed.

[3]This is known as a Pittsburgh-style GA [29].

[4]Details of the mutation and crossover operators are omitted here to save the space.

TABLE II
CLASSIFICATION PROBLEMS

Name	No. of Inputs	No. of Output Classes	No. of Samples
Breast Cancer	9	2	683
Diabetes	8	2	768
Iris	4	3	150
New Thyroid	5	3	215
Wine	13	3	178

V. EXPERIMENTAL RESULTS

This section presents computer simulation results of applying the proposed descriptive techniques to a number of benchmark problems. The experimental background is first described. A simple example in terms of resultant descriptive rules is given next, in comparison to the original approximative rules. Comprehensive results are then reported and analyzed, supported by comparisons with related work.

A. Experimental Background

To demonstrate the proposed approach at work, benchmark classification problems are used here [32], including the Breast Cancer, Diabetes, New Thyroid, Wine, and Iris datasets. Table II summarizes the setups of these datasets.

A neurofuzzy approximative rule induction algorithm ANFIS [21],[5] which uses bell-shaped approximative sets, and which has been optimized, was trained for these problems. The resulting approximative ruleset is employed as the original set of rules for translation.

As indicated before, the output of the heuristic method is used to act as the generator of the initial population for the GA that performs finer search for the final descriptive rules. To ensure the readability and understandability of the resulting descriptive ruleset, the maximum number of hedges (including the NOT operator) allowed to be applied to a given set is limited to two.

For comparison, the pure descriptive induction algorithm as given in [33], which is a form of exhaustive search with different parameter settings is also tested. In particular, this algorithm (referred to as Lozowski's algorithm hereafter) has a parameter that trades off between the model accuracy and the size of learned ruleset. It determines the minimum difference between the firing strengths of any given rules that have the same antecedent but different class values. In the present investigation, this parameter is set up with reference to the number of rules that the heuristic method has generated to ease comparison. Also, for comparison purposes, results obtained by running the standard C4.5 algorithm [34] are included.

To be simple and fair for comparison, for each problem considered the fuzzification was carried out proportionally with respect to the size of the universe of discourse of the individual

[5]Note that the ANFIS algorithm is considered as one of the best approximative modelers at the moment.

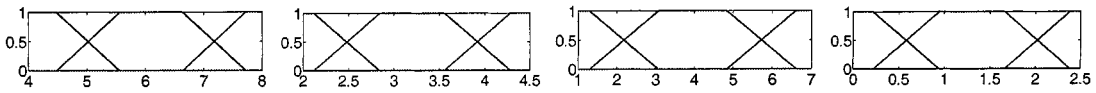

Fig. 7. Descriptive sets for sepal length and width, and petal length and width, respectively (Iris).

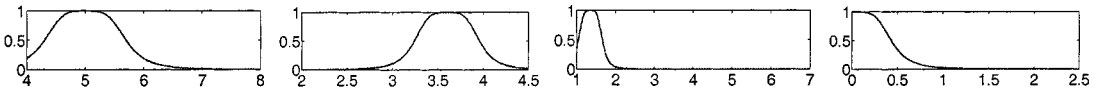

Fig. 8. Fuzzy sets for approximative rule A1 (Iris Setosa).

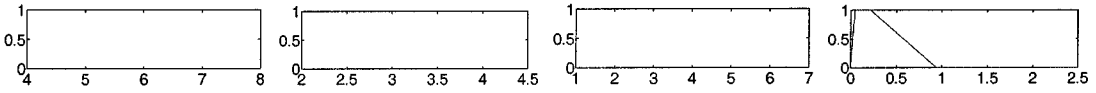

Fig. 9. Hedged sets for descriptive rule D1 (Iris Setosa).

variables. That is, for each variable, the distance between its maximum and minimum value within the data set is divided such that all of them approximately cover an equal range of the underlying real values, with soft boundaries of course. The fuzzy sets resulting from such a partition are regarded as the given descriptive sets. This is implemented for illustrative purposes, and is not necessary in practical applications of the present work. In fact, the whole idea is that the fuzzy partitioning and labeling will be done by user/experts. To demonstrate the effectiveness of this approach the fuzzification scheme used has only three labels per variable.

The GA uses a population of 30 rulesets and run for 10 000 evaluations (not generations). Note that, to allow comparison, for a given execution the evaluations are divided among the different subproblems or sub-GAs, if applicable. These sub-GAs are GAs running on parts of the translation. For example, for the individual translation strategy a sub-GA is a GA used to translate a particular rule. In group strategy, it is a GA running a particular class. The term sub-GA does not apply to the global translation strategy as there is only one problem, i.e., the translation of the whole approximative ruleset. Thus, an execution of the global strategy will have 10 000 evaluations, while a group execution for three classes will have 3333 evaluations per class translation and an execution of the individual translation strategy for ten rules will have 1000 evaluations for each rule translation.

As GA execution is computationally affordable, it is worthy to execute the genetic search for as many times as possible. This is in order to obtain the best among as large number of different translations as possible to act as the final translation. The figures to be presented below will show the mean error of translated rulesets depending on the number of runs allowed for the GA. Such error measures are obtained using a bootstrapping of 1000 samples over 100 real runs. Experimental results are given for the GA guided by *functional equivalence* and also for the GA guided by classification (error) rate alone. To avoid possible overfitting, each dataset has been separated into a training set containing 75% of all the given data and a test set comprising the remaining 25%.

B. Example of Transparency Gained by Translation

To reflect the fundamental differences between approximative and descriptive modeling, an example ANFIS ruleset for the Iris problem [35] and one of its descriptive translations are given here. The approximative rules are as follows.

Rule A1: if x_0 is $Bell_{0.70,1.99,5.00}$ and x_1 is $Bell_{0.36,2.06,3.60}$ and x_2 is $Bell_{0.31,2.11,1.35}$ and x_3 is $Bell_{0.40,2.04,0.03}$ then Class is $Setosa$

Rule A2: if x_0 is $Bell_{1.02,1.99,6.23}$ and x_1 is $Bell_{0.301,2.31,2.84}$ and x_2 is $Bell_{0.39,2.18,4.02}$ and x_3 is $Bell_{0.11,2.31,1.53}$ then Class is $Versicolor$

Rule A3: if x_0 is $Bell_{0.58,1.99,7.53}$ and x_1 is $Bell_{0.71,2.00,3.06}$ and x_2 is $Bell_{0.27,2.11,6.33}$ and x_3 is $Bell_{0.64,1.99,2.18}$ then Class is $Virginica$

Rule A4: if x_0 is $Bell_{0.14,1.91,6.25}$ and x_1 is $Bell_{0.53,2.15,2.46}$ and x_2 is $Bell_{0.24,2.11,5.31}$ and x_3 is $Bell_{0.37,1.97,2.30}$ then Class is $Virginica$

Obviously, these rules are not readable, though they may be generated very rapidly and they may well generalize the given training data.

Suppose that the labels attached to a variable's three possible descriptive sets are named long, medium, and short or thin, medium, and wide, depending on whether the variable refers to length or width respectively. Each descriptive set may be modified by zero up to two hedges (as defined in Section III). Given the approximative rules, the following translated rules may be generated.

```
Rule D1: IF Petal Width IS Upper Thin
THEN Iris-Setosa
Rule D2: IF Sepal Length IS Medium
AND Sepal Width IS Greatly Medium AND
Petal Length IS Very Medium AND Petal
Width IS Very Medium THEN Iris-Versi-
color
Rule D3: IF Petal Width IS Lower
Greatly Wide THEN Iris-Virginica
```

To examine the translation results, Fig. 7 shows the descriptive partition used in this example. Figs. 8 and 9 plot the mem-

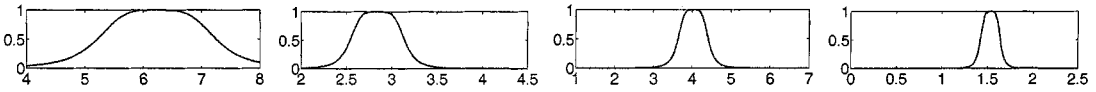

Fig. 10. Fuzzy sets for approximative rule A2 (Iris Versicolor).

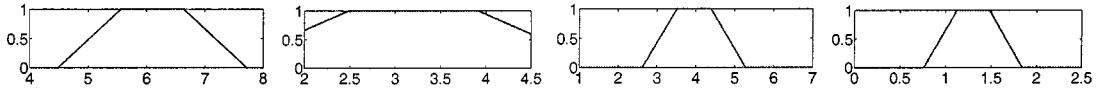

Fig. 11. Hedged sets for descriptive rule D2 (Iris Versicolor).

bership functions involved in the antecedent of the approximative rule A1 and its translation D1 (only the Peal Width has a value, Upper Thin). Also Figs. 10 and 11 plot rule A2 and its translation D2.

These descriptive rules may appear rather different from the original approximative ones, yet they have the same functionality. The translated rules are represented in linguistic words with predefined meanings. In using such a rule base, both the interpretation of the inferences performed and the explanation of the fuzzy system itself becomes straightforward. Very interestingly, for this example, the number of resultant descriptive rules is actually less than that of their original, while these two rulesets entail the same classification accuracy. Also, two out of the three descriptive rules are more concisely represented than any of the four original rules.

In general, it is very difficult to expect a double translation (which starts from a descriptive model, produces a dataset from this model, generates an approximative model of that dataset and then translates it back to a descriptive model) to reproduce the same original descriptive ruleset or even to resemble the original closely. This is more obvious in modeling complex systems where many possible combinations of rules can yield similar rule-firing results. No principled way exists to guarantee that a data driven rule induction mechanism would reproduce exactly the same rules that were first used to create the data. What can be expected most is to be able to generate a set of rules that match the data as closely as possible, hoping that such a set of rules do not differ too much from the underlying one. Although multiple descriptive rulesets may be obtained from one given approximative model, only one optimized is eventually chosen to act as the translation. Thus, the explanation will be unique for a problem at hand once the translation process is completed.

C. Results on Model Accuracy

First of all, it is interesting to investigate the effects of using different translation strategies. Fig. 12 collects the results of such experiments on the Diabetes problem, as an example, while results on other problems are very similar. It can be seen that the group strategy gave better results than the global and individual ones during the training phase. In testing, all the three strategies performed very similarly overall. However, difference exists in terms of the number of rules needed to achieve the similar test results. Individual translation gave more, sometimes many more, rules as the GA tried to produce good translations locally. The pressure on the search exerted by the number of rules objective on individual strategy was not so high as the pressure by

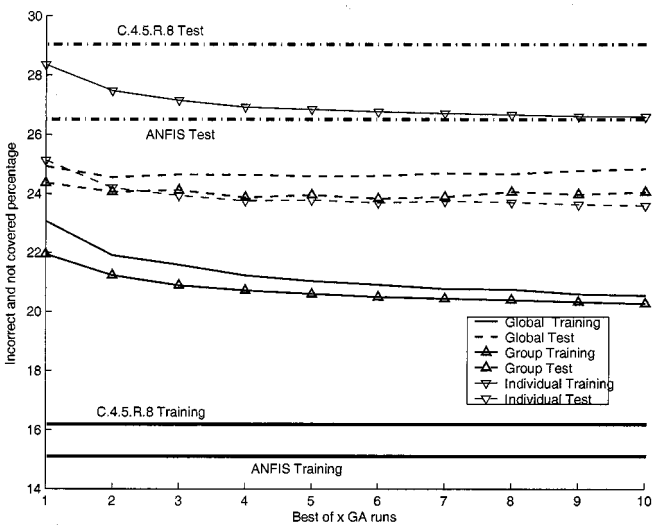

Fig. 12. Classification error versus translation strategy for the Diabetes problem.

the same objective in carrying out group or global translations, where more data points were involved. When using group or global strategies, it may be the case that several approximative rules can be covered by just one descriptive rule, with potential savings in the number of rules generated.

As group and global strategies allow for more data to be covered, there is more pressure to reduce the number of rules using either of these strategies than the individual strategy. However, this difference in pressure can be compensated to a certain extent as more general strategies have more evaluations to run. The strong point of the individual strategy rests in the fact that it leads to very short and compact rules as can be seen in Fig. 13.

The pressure on the number of rules objective can be used to reduce the size of the resultant ruleset. This will make it easier to understand the general behavior of the system under consideration. Nevertheless, in general, a reduced number of rules is likely to result in a reduced classification rate (see Figs. 13 and 14 or Figs. 12 and 15).

As different translation strategies lead to very similar testing performances of the resulting descriptive rulesets, only those results obtained by the use of group strategy are hereafter presented. Tables III and IV give the results of runs with respect to the following (where Trn, Tst, and Rul, respectively, stand for the classification error percentage over training data, the classification error percentage over testing data, and the number of rules generated):

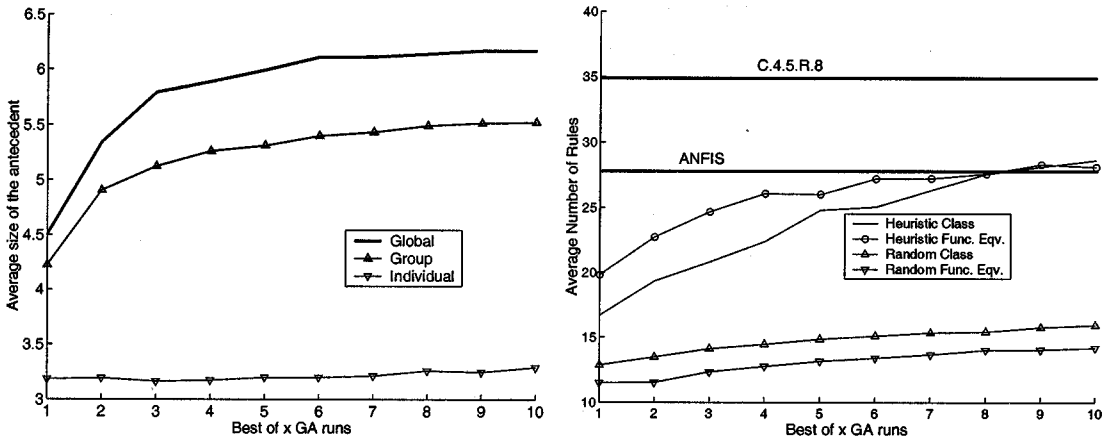

Fig. 13. Mean size of antecedents versus translation strategy (left) and number of rules using the group strategy (right) for the Diabetes problem.

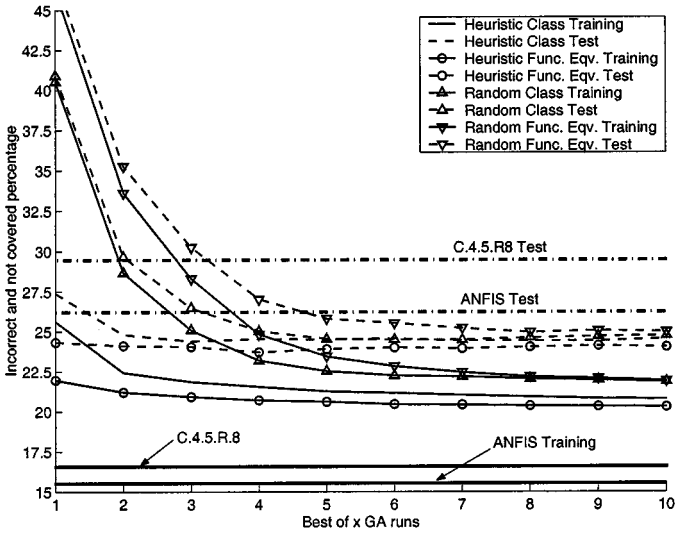

Fig. 14. Classification error for the Diabetes problem using the group strategy.

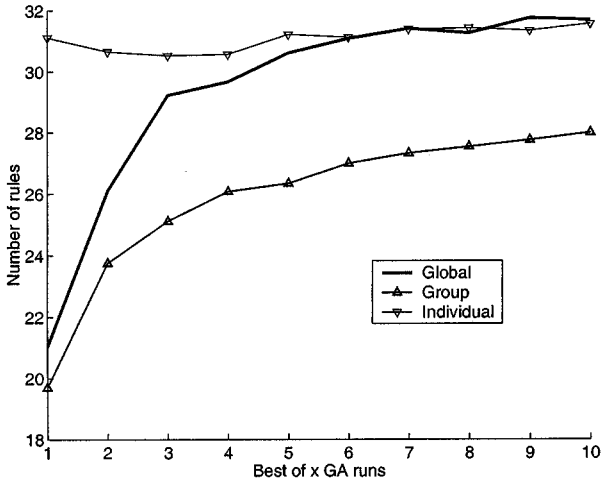

Fig. 15. Number of rules versus translation strategy for the Diabetes problem.

1) starting from a heuristic translation and guiding the GA by classification objectives only;
2) starting from a heuristic translation and guiding the GA by *functional equivalence* and classification objectives;
3) starting from randomly generated rules and guiding the GA by classification objectives only;
4) starting from randomly generated rules and guiding the GA by *functional equivalence* and classification objectives.

Within these tables, the most interesting points to compare are those given in the columns concerning items 2) and 3) above. The former shows the result of what is suggested in this work, and the later shows that of pure data-driven search (no heuristic translation nor approximative model provided). Note that results on the Diabetes problem were collected with two different weights set for the *number of rules* objective, in order to illustrate the impact of this objective upon the translation.

To support the comparison, results of applying a descriptive ruleset which is obtained by the initial heuristic translation alone, and those of using C4.5, ANFIS and Lozowski's algorithm are also provided as given in Table V. The work presented in this paper performs well and does so consistently in testing. The accuracy of translated descriptive models is close to that achievable by the optimized ANFIS (comparing Table IV and the middle column of Table V), and generally outperforms the other descriptive modeling techniques tested (comparing Table IV and columns three and four of Table V).

Fig. 14 shows a graphical comparison, for the Diabetes problem, of the translation results with respect to the number of GA runs. It reveals the difference between different ways of guiding the GA, either starting from a set of descriptive rules obtained by the heuristic translation of the approximative rules or from a set of randomly generated descriptive rules. Graphs plotting the classification errors for the different classification problems appear to be almost the same in their general tendency. They only differ in the actual values of the classification error and in how many GA runs are needed to reach a state where running more GAs will not improve the result. These graphs are therefore omitted here.

This general trend shows that a GA-based translation starting from random rules produces systematically worse results than that starting from the heuristic translation. In addition, the results achievable with a heuristic translation start are far more stable than those obtained with a random start. This supports the need for generating such first crude translation. For simpler

TABLE III
MEAN RESULTS OF TRANSLATIONS FOR ONE GA RUN ONLY

Problem	Heuristic Class			Heur Fun. Eqv.			Random Class			Rand Fun. Eqv.		
	Trn	Tst	Rul	Trn	Tst	Rul	Trn	Tst	Rul	Trn	Tst	Rul
Breast Cancer	2.9	7.3	9.5	1.1	5.6	8.5	14.9	18.3	9.8	10.0	14.0	9.1
Diabetes (Norm)	25.0	26.8	16.9	21.9	24.5	19.8	41.0	41.5	12.8	46.2	46.8	11.5
Diabetes (High)	28.7	29.7	8.3	25.3	27.2	9.1	48.5	48.6	9.4	46.4	46.4	3.4
Iris	1.3	4.6	5.4	1.1	4.4	5.3	5.2	9.9	5.2	3.7	6.2	5.1
New Thyroid	12.3	13.0	7.2	7.0	7.9	6.2	54.6	55.3	7.2	45.9	46.9	6.1
Wine	3.8	10.3	10.3	1.5	5.7	10.8	45.4	47.5	10.2	36.4	39.7	10.6

TABLE IV
MEAN RESULTS OF THE BEST TRANSLATION OUT OF 5 GA RUNS

Problem	Heuristic Class			Heur Fun. Eqv.			Random Class			Rand Fun. Eqv.		
	Trn	Tst	Rul	Trn	Tst	Rul	Trn	Tst	Rul	Trn	Tst	Rul
Breast Cancer	0.8	5.7	9.3	0.7	5.6	8.3	0.8	5.7	9.1	0.7	5.7	9.0
Diabetes (Norm)	21.2	24.4	24.0	20.5	23.8	26.7	24.6	27.2	14.0	23.5	25.4	15.7
Diabetes (High)	22.7	26.0	9.8	21.4	24.2	10	25.2	26.6	8.2	25.1	27.8	9.1
Iris	0.49	3.98	5.4	0.43	3.43	4.7	0.9	5.0	5.4	0.8	4.6	5.5
New Thyroid	6.4	7.1	7.5	5.8	6.6	6.1	11.3	13.3	7.3	9.3	11.4	6.2
Wine	0.3	6.5	10.8	0.3	3.2	9.8	6.8	13.1	9.7	2.7	9.21	10.4

TABLE V
RESULTS OF INITIAL TRANSLATION AND OTHER MODELING METHODS

Problem	C4.5			ANFIS			Lozowski			Heuristic		
	Trn	Tst	Rul	Trn	Tst	Rul	Trn	Tst	Rul	Trn	Tst	Rul
Breast Cancer	1.2	7.6	29	0.4	4.1	18	10.3	15.2	74	25.7	23.9	94
Diabetes	16.1	29.2	35	15.7	26.6	28	38.2	39.6	65	33.5	32.8	43
Iris	1.8	2.6	5	0.8	2.6	4	4.5	10.5	11	8.9	11.6	9
New Thyroid	1.9	7.4	13	3.1	1.8	4	82.6	83.3	4	25.5	25.9	4
Wine	0.8	2.2	17	0	2.2	6	39.8	44.4	21	30.1	22.2	23

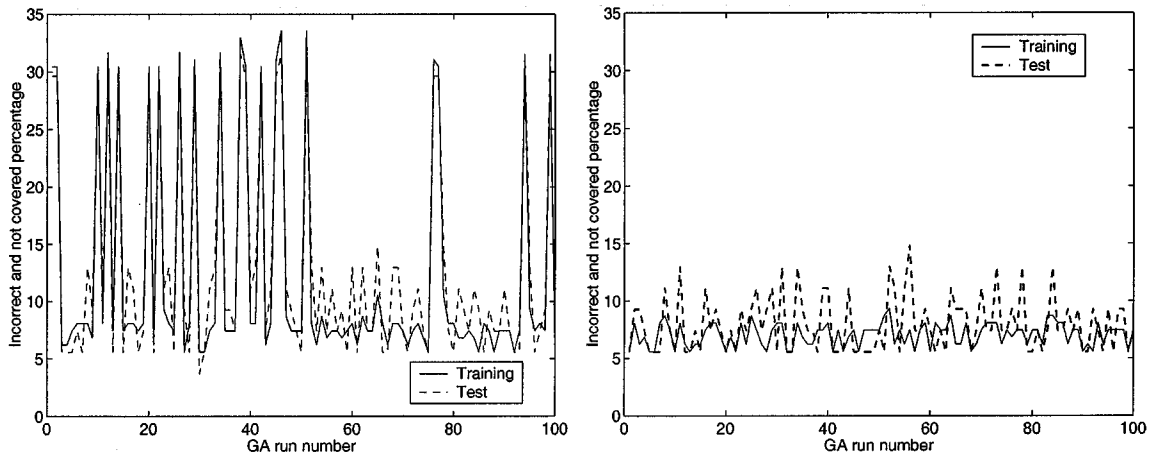

Fig. 16. 100 runs for class guided (left) and functional equivalence guided GA (right) on thyroid.

problems (e.g., Breast Cancer), both starting points may eventually lead to similar classification accuracy, yet several runs are needed to ensure this similarity. Nevertheless, as shown in Table III, if only a single run is executed (say, due to a computational resource limit) the results obtained using a randomly generated initial population would be much worse than those obtained using the initial population produced by the heuristic method. It clearly pays off to generate the heuristic translation first and to use such a descriptive rule set to act as the generator for the initial population of the GA.

Guidance for the GA by classification error only is also, in general, rather unstable, when compared with the use of *functional equivalence* guidance. This comparison is shown in Fig. 16, where results of 100 runs on classifying the Thyroid problem, with a heuristic start and using the group strategy, are depicted. Following the guidance by classification error alone produces rather poor runs with high error peaks, which actually happens in all tested problems. This unstability of the results can be partially overcome with a higher number of GA runs, though those extra runs are not always affordable computation-

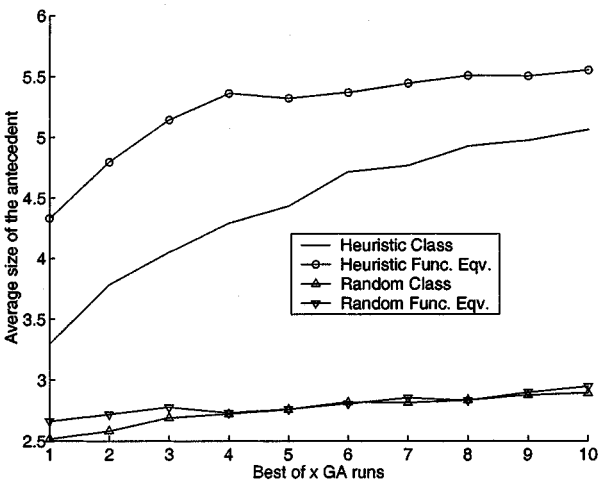

Fig. 17. Mean size of the antecedents for the Diabetes problem using the group strategy.

ally. Even if extra GA runs were affordable it would still be better to use the criterion of *functional equivalence* over all the runs.

The aformentioned results of: 1) starting from a heuristic translation and 2) making use of *functional equivalence* reveal an important point. That is, generating an approximative model first (to get required *functional equivalence* objectives) and then implementing a heuristic translation (to act as the initial population generator for the GA) improves significantly the final descriptive ruleset.

A positive side effect of the way that the rules are codified in the GA is that the size of the antecedent, i.e., the number of conditions in the antecedent, is variable. This implies that there is an implicit attribute reduction going along with the GA-based optimization as shown in Figs. 13 and 17. This reduction is more evident in high dimensional problems. If desired, the reduction of antecedent conditions may even be explicitly introduced as another optimization objective.

Finally, it is worth noting that the performance of Lozowski's algorithm, which is an exhaustive search based method, never gets close to that of ANFIS or C4.5. In all the problems tested this algorithm gives poorer results than the approximative modeling methods used and yet requires more rules even for reaching such less desirable results. Compared just to the heuristic translation Lozowski's algorithm only defeats it for the Breast Cancer and Iris problems, but its corresponding translated models contain a considerably higher number of rules. However, to be fair with this comparison it must be remembered that Lozowski's algorithm makes use of no hedges.

VI. Conclusion

A major disadvantage of several existing methods for building descriptive fuzzy systems is that the generation of fuzzy rules is usually made via an exhaustive search throughout the input product space. In addition, the rules produced by pure descriptive methods usually have a low accuracy. However, approximative rule generation methods can be very fast and accurate, through they tend to having difficulties in interpreting the underlying meaning of the data being modeled and in facilitating understanding of the inferences drawn from the resultant models.

In order to generate accurate rules that possess the desirable property of being readily comprehensible to human users and to create such rules in an efficient way, a translation technique from approximative to descriptive rules has been proposed here. The translation is enabled by the use of linguistic hedges to change implicitly the prescribed, meaningful descriptive fuzzy sets, such that the modified will closely resemble the original approximative ones in function. The approximative rules may be themselves created by any standard approximative modeling technique. The modification process is indeed implemented via a *functional equivalence* guided search. Novel hedges that are useful for supporting such translations are defined in this work. A heuristic algorithm has been presented, which implements a preliminary translation that is employed to act as the generator for the initial population used by a GA to perform the fine grain modifications.

The results obtained so far have demonstrated that the proposed approach does not significantly decrease the accuracy attainable by the original approximative models, and that the descriptive rules obtained are interpretable by humans. It outperforms exhaustive search methods for descriptive rule generation in terms of search efficiency, as searches for descriptive rulesets are herein heuristically guided. In terms of classification accuracy, it also outperforms pure descriptive methods which do not use hedges, while it is impractical to run exhaustive search with hedges, as that would complicate even more the usually already huge search space. These results were achieved without important attempts to optimize the GA employed.

Acknowledgment

The authors would like to thank A. F. Gómez-Skarmeta of the University of Murcia, Murcia, Spain, and P. Ross of Napier University, Edinburgh, U.K., for their helpful discussions and assistance in the research reported, while taking the full responsibility of the views expressed here. They would also like to thank the anonymous referees for their constructive comments which were very useful in revising this paper.

References

[1] S. Abe, "Dynamic cluster generation for a fuzzy classifier with ellipsoidal regions," *IEEE Trans. Syst., Man, Cybernetics B*, vol. 28, pp. 869–876, Dec. 1998.

[2] L. X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, pp. 1414–1427, Nov. 1992.

[3] H. Roubos and M. Setnes, "Compact and transparent fuzzy models and classifiers through iterative complexity reduction," *IEEE Trans. Fuzzy Syst.*, vol. 9, pp. 516–524, Aug. 2001.

[4] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning i," *Inform. Sci.*, vol. 8, pp. 199–249, 1975.

[5] T. Rauma, "Knowledge acquisition with fuzzy modeling," in *Proc. 5th IEEE Int. Conf. Fuzzy Systems*, vol. 3, 1996, pp. 1631–1636.

[6] J.-S. R. Jang, Y. C. Lee, and C.-T Sun, "Functional equivalence between radial basis function networks and fuzzy inference systems," *IEEE Trans. Neural Networks*, vol. 4, pp. 156–159, Jan. 1993.

[7] D. Nauck and R. Kruse, "NEFCLASS-X—A soft computing tool to build readable fuzzy classifiers," Tech. Rep., 3, BT, July 1998.

[8] M. Setnes and H. Roubos, "Ga-fuzzy modeling and classification: Complexity and performance," *IEEE Trans. Fuzzy Syst.*, vol. 8, pp. 509–522, Oct. 2000.

[9] M. Setnes, R. Babuska, and H. B. Verbruggen, "Transparent fuzzy modeling," *Int. J. Human–Comput. Stud.*, vol. 49, no. 2, pp. 159–179, 1998.
[10] ——, "Rule-based modeling: Precision and transparency," *IEEE Trans. Syst., Man, Cybern. C*, vol. 28, pp. 165–169, Feb. 1998.
[11] J. Valente de Oliveira, "Semantic contrains for membership function optimization," *IEEE Trans. Syst., Man, Cybern. A*, vol. 29, pp. 128–138, Jan. 1999.
[12] M. Sugeno and T. Yasukawa, "A fuzzy logic based approach to qualitative modeling," *IEEE Trans. Fuzzy Syst.*, vol. 1, pp. 7–31, Aug. 1993.
[13] J. Gómez Marín-Blázquez, Q. Shen, and A. F. Gómez Skármeta, "From approximative to descriptive models," in *Proc. 9th IEEE Int. Conf. Fuzzy Systems*, May 2000, pp. 829–834.
[14] J. Gómez Marín-Blázquez and Q. Shen, "Linguistic hedges on trapezoidal fuzzy sets: A revisit," in *Proc. 10th IEEE Int. Conf. Fuzzy Systems*, vol. 1, Dec. 2001, pp. 412–415.
[15] D. E. Goldberg, *Genetic Algorithms in Search, Optimization 6 Machine Learning*. Reading, MA: Addison-Wesley, 1989.
[16] T. M. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.
[17] H. Ishibuchi, T. Nakashima, and T. Morisawa, "Simple fuzzy rule-based classification systems perform well on commonly used real-world data sets," in *Proc. NAFIPS'97*, 1997, pp. 251–256.
[18] S. Dick, A. Kandel, and W. E. Combs, "Comments on 'Combinatorial rule explosion eliminated by a fuzzy rule configuration'," *IEEE Trans. Fuzzy Syst.*, vol. 7, pp. 475–478, Aug. 1999.
[19] H. Ishibuchi, T. Nakashima, and T. Murata, "Performance evaluation on fuzzy classifier systems for multidimensional pattern classification problems," *IEEE Trans. Syst., Man, Cybern. B*, vol. 29, pp. 601–618, Oct. 1999.
[20] E. Cox, *The Fuzzy Systems Handbook*. Cambridge, MA: AP Professional, 1994.
[21] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*. Upper Saddle River, NJ: Prentice-Hall, 1997.
[22] P. D. Surry and N. J. Radcliffe, "Inoculation to initialize evolutionary search," in *Proc. 3rd AISB Workshop Evolutionary Computation*, T. C. Fogarty, Ed., Brighton, U.K., 1996, pp. 269–285.
[23] D. S. Yeung and C. C. Tsang, "A comparative study on similarity-based fuzzy reasoning methods," *IEEE Trans. Syst., Man, Cybern. B*, vol. 27, pp. 216–227, Apr. 1997.
[24] R. Zwick, E. Carlstein, and D. V. Budescu, "Measures of similarity among fuzzy concepts: A comparative analysis," *Int. J. Approx. Reas.*, vol. 1, pp. 221–242, 1987.
[25] L. A. Zadeh, "Similarity relations and fuzzy orderings," in *Fuzzy Sets and Applications: Selected Papers by L. A. Zadeh*, R. R. Yager, S. Ovchinnikov, R. M. Tong, and H. T. Nguyen, Eds. New York: Wiley, 1987, pp. 81–104.
[26] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
[27] J. D. Schaffer and J. J. Grefenstette, "Multi-objective learning via genetic algorithms," in *Proc. 9th Int. Joint Conf. Artificial Intelligence*, A. Joshi, Ed., Los Angeles, CA, Aug. 1985, pp. 593–595.
[28] P. J. Bentley, *Evolutionary Design by Computers*. London, U.K.: Academic, 1999.
[29] S. F. Smith, "A learning system based on genetic algorithms," Ph.D. dissertation, Univ. Pittsburgh, Pittsburgh, PA, 1980.
[30] D. D. Leich, "A new genetic algorithm for the evolution of fuzzy systems," Ph.D. dissertation, Dept. Engineering Sci., Univ. Oxford, Oxford, U.K., 1995.
[31] J. Gómez Marín-Blázquez and Q. Shen, "Toward the generation of descriptive fuzzy rules via approximative modeling," in *Proc. 6th U.K. Workshop Fuzzy Systems*, 1999, pp. 1–14.
[32] Uci machine learning databases. [Online]. Available: http://ftp.ics.uci.edu/pub/machine-learning-databases/.
[33] A. Lozowski, T. J. Cholewo, and J. M. Zurada, "Crisp rule extraction from perceptron network classifiers," in *Proc. Int. Conf. Neural Networks*, Plenary, Panel, and Special Sessions, Washington, DC, 1996, pp. 94–99.
[34] J. R. Quinlan, *C4.5: Programs for Machine Learning.*. San Mateo, CA: Morgan Kaufmann, 1993.
[35] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eug.*, pt. II, vol. 7, pp. 179–188, 1936.

Javier G. Marín-Blázquez (S'99) was born in Murcia, Spain, in 1971. He received the B.Sc. (Honors) and the M.Sc. degrees in computer science from the University of Murcia, Murcia, Spain, and the M.Sc. degree in artificial intelligence from the University of Edinburgh, Edinburgh, U.K., in 1992, 1994, and 1998, respectively. He is currently working toward the Ph.D. degree in artificial intelligence at the University of Edinburgh.

He is a Junior Lecturer in the Division of Informatics, the University of Edinburgh, working in the Approximate and Qualitative Reasoning Group. From 1994 to 1997, he was a Research Assistant at the University of Murcia. His research interests include fuzzy and linguistic modeling, pattern recognition, and evolutionary algorithms. He has published approximately 15 peer-refereed articles on topics within artificial intelligence and related areas.

Qiang Shen received the B.Sc. and M.Sc. degrees in communications and electronic engineering from the National University of Defence Technology, China, and the Ph.D. degree in knowledge-based systems from Heriot-Watt University, Edinburgh, U.K.

He is a Senior Lecturer in the Division of Informatics, University of Edinburgh, Edinburgh, U.K., where he leads the Approximate and Qualitative Reasoning Group. His research interests include fuzzy and imprecise modeling, model-based inference, pattern recognition, and knowledge refinement and reuse. He has published 120 peer-refereed papers on topics within artificial intelligence and related areas in academic journals and conference proceedings.