From Approximative to Descriptive Models

Javier G. Marín-Blázquez[†] and Qiang Shen
School of Artificial Intelligence
Division of Informatics
The University of Edinburgh, UK
{javierg,qiangs}@dai.ed.ac.uk

Antonio F. Gómez-Skarmeta
Dept. Informática, I.A. y Electrónica
Facultad de Informática
Universidad de Murcia, Spain
skarmeta@dif.um.es

Abstract—**This paper presents an effective and efficient technique for translating rules that use approximative sets to rules that use descriptive sets and linguistic hedges of predefined meaning. The translated descriptive rules will be functionally equivalent to the original approximative ones, or the closest equivalence possible, while reflecting their underlying semantics. Thus, descriptive models can take advantage of any existing approach to approximative modelling which is generally efficient and accurate, whilst employing rules that are comprehensible to human users.**

I. Introduction

Fuzzy modelling has been proposed as an effective method to model systems that are too complicated or ill-defined for traditional methods. Among the interesting features of fuzzy models is the potential of fuzzy production rules in attaching semantic labels to the fuzzy sets [17], thereby creating a human comprehensible model of the system.

The majority of the fuzzy rule bases and/or the fuzzy sets used are, however, created and tuned to best fit the data available. They are not encoded to keep the meaning of the semantic labels but follow the so-called approximative approach. Such an approach is, under minor restrictions, functionally equivalent to neural networks [7] and, like them, offers little explanatory power over its inferences. Opposing this stands the descriptive approach, in which semantics are as important as accuracy. In descriptive modelling, prescribed fuzzy sets are either not allowed to be modified or, at most, are allowed to have very slight modifications. The descriptive sets used induce a fuzzy grid in the product space of the domain variables. As few modifications are permitted, the grid, and the hyperboxes delimited by it, is almost fixed. Often these hyperboxes may contain examples of different output states and, as the hyperboxes are fixed, there is no way to separate them. The approximative approaches solve this problem by changing the definitions of the sets, and therefore the hyperboxes themselves. Yet, in so doing, they generally ruin the underlying semantics. The current literature either pays little attention to this semantic ruining effect from using an approximative model (some even regarding such models as descriptive), or maintains a descriptive model and accepts high modelling errors. There is however a way to change the shape of the sets without disrupting the semantics. The use of hedges allows such fine modification and hence more freedom with the hyperbox definition.

One of the disadvantages of descriptive models is that the rules are usually generated by an exhaustive search in the input product space. This can only be done for a small number of variables/labels due to the potential combinatorial explosion. Even with a manageable size of variables/labels, the automatic generation of descriptive rules is generally very slow, whilst many existing approximative methods are able to find very accurate rules rapidly. Thus, the question becomes if there exists a way to generate descriptive rules using a variation of these approximative methods? In [15] a translation procedure is suggested to obtain descriptive explanations of approximative models. Here, the proposal is to generate a descriptive model with a two-step method. The first step is to use an approximative method to create accurate modelling rules and the second to convert the resulting approximative rules to descriptive ones. In [15], however, each rule is translated from one approximative hyperbox to one closest descriptive hyperbox. Here, the proposal is to perform a one-to-many rule translation using a heuristic method and to perform a fine tuning using a systematic search through evolutionary computing. The search will be guided by *functional equivalence* rather than by similarity between the antecedent approximative and the predefined descriptive fuzzy sets. In addition, to ensure a highly accurate translation, novel linguistic hedges are defined and used in this paper.

The rest of the paper is organised as follows. Section II proposes a set of useful linguistic hedges, which differ from those conventionally employed in the literature. Section III presents the translation process, covering two methods, one based on the use of heuristics and the other on a genetic algorithm. Section IV reports on typical experimental results, demonstrating the potential of the present proposal. The paper is concluded in section V, with further work pointed out.

II. Hedges

A linguistic hedge modifies the shape of a fuzzy set's definition, causing changes in the membership function [4]. That is, a hedge transforms one fuzzy set into another. The semantics of the transformed set can be extracted from the semantics 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. Conventional definitions of the hedges such as those of applying powers to the membership values of the original set [4], [8], do not result in significant changes on trapezoid fuzzy sets, which are most commonly used for computational

[†]Partly supported by the Fundación Séneca.

0-7803-5877-5/00/$10.00 © 2000 IEEE

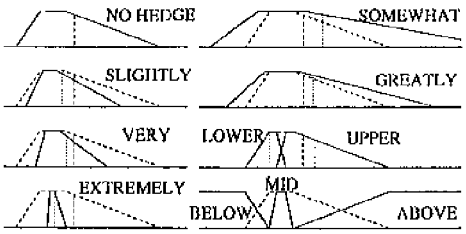

Fig. 1. Hedges applied to an irregular trapezoid and how the way they change the center of gravity

simplicity purposes. In particular, the full membership part of a trapezoid membership function does not get changed at all. In this work a different set of hedges is considered, which may be applied to dilute or intensify the original fuzzy sets by shrinking or expanding any parts of the trapezoids. In addition, three new hedges called UPPER, LOWER and MID that do not appear in the literature are also proposed.

A trapezoidal membership function, characterised by 4 parameters (a,b,c,d), consists of three consecutive segments. In the first segment (over a to b) the membership value increases linearly from zero to full membership. The second (over b to c) covers members of a full membership value and is referred to as the full membership segment (FMS). In the third segment (over c to d) the membership value decreases linearly from full to zero value. The application of dilation/intensification hedges should increase/decrease the size of these segments and, therefore, be implemented with the proportion of modification kept relatively to the center of the full membership segment ($CFMS$).

A. Dilation

Dilation hedges reduce the size of segments and parts of the membership values of an original set. For a given trapezoidal fuzzy set A with a membership function $\mu_A(x)$, the set modified by a dilation hedge DIL should comply with $\forall x \in X, \mu_{DIL \cdot A}(x) \leq \mu(x)$. The parameters of the modified set are therefore defined to be calculated by:

$$\begin{gathered} CFMS = \frac{b+c}{2} \\ a' = b' - ((b-a) * factor) \\ b' = CFMS - ((CFMS - b) * factor) \\ c' = CFMS + ((c - CFMS) * factor) \\ d' = c' + ((d-c) * factor) \end{gathered}$$

where $factor$ controls the shrinking degree of a dilation hedge. In order to reduce the set effectively $factor$ must satisfy $0 < factor < 1$. In particular, the commonly used hedge terms $SLIGHTLY$, $VERY$ and $EXTREMELY$ may be defined by setting $factor$ to $\frac{2}{3}$, $\frac{1}{2}$ and $\frac{1}{8}$ respectively.

B. Intensification

Intensification hedges increase the size and parts of the membership values of a set. In this case, an intensification hedge INT should comply with the following: $\forall x \in X, \mu_{INT \cdot A}(x) \geq \mu_A(x)$. The parameters of the modified set are calculated as Dilation hedges, although this time the $factor$ will be greater than one. In particular, $SOMEWHAT$ and $GREATLY$ can be implemented by setting the $factor$ to 2 and $\frac{3}{2}$.

C. Restriction

Restriction hedges [4], $ABOVE$ and $BELOW$, are applied to variables where fuzzy values are ordered. The set modified by applying the $BELOW$ hedge represents the set which is "less than" the original and that by the $ABOVE$ denotes the set which is "greater than" the original set. Their membership functions are defined as follows:

$$\mu_{ABOVE \cdot A}(x) = \begin{cases} x < a & 1 \\ a \leq x < b & 1 - \mu_A(x) \\ x \geq b & 0 \end{cases}$$

$$\mu_{BELOW \cdot A}(x) = \begin{cases} x < c & 0 \\ c \leq x < d & 1 - \mu_A(x) \\ x \geq d & 1 \end{cases}$$

D. Detailisation

This new type of hedge splits the original set into three, but keeps the order of the split sets with respect to the order of the elements belonging to the FMS of the original. The resulting three sets $LOWER \cdot A$, $MID \cdot A$ and $UPPER \cdot A$, arranged in an increasing order, are defined by:

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

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

$$UPPER \cdot A \begin{cases} a' = b + \frac{b-c}{3} & b' = b + \frac{2*(b-c)}{3} \\ c' = c & d' = d \end{cases}$$

Finally, for any original shouldered membership function, hedges are applied by such that it is trapezoidal with their extreme parameters (a and b for the left shouldered and c and d for the right shouldered sets) set to be equal to a certain maximum/minimum value.

III. Mapping approximative into descriptive rules

The aim of this work is to find an effective and efficient way to translate rules that use approximative sets to rules that use descriptive sets and hedges. The translated rules will be equivalent to the original or the closest equivalence possible with the advantage of having (or regaining) semantic interpretation.

Many different methods exist for automatically generating approximative rules. Broadly speaking, they can be categorised into three groups: Clustering methods [2], partition methods [3] and search methods [13], [5], [6], [14]. For the the present work it does not matter 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.

To perform the translation two different methods are proposed here. One is based on a heuristic search. The second method uses evolutionary computation techniques. As evolutionary search usually works better if a good start point is identified, the first method will be employed as the generator of the initial population for the genetic method which will then make a finer-grain search.

A. Heuristic approach

This approach is based on hyperbox intersection. Descriptive rules (hyperboxes defined on descriptive sets shown as white boxes in figure 2) are created if they intersect with an approximative rule (the hyperbox defined by the antecedent approximative sets like the shadowed box in figure 2). This produces a crude translation (and so it is in fact used as the generator of the initial population for the evolutionary method to be presented later).

A.1 Basic Method

The idea is to build a layered graph to represent degrees of intersection (using a similarity measure). Each layer of the graph consists of a certain number of nodes, each of which represents the rate of intersection between one of the approximative sets of the antecedent and one of the descriptive sets of the same variable involved. The rate of intersection between two sets S_1 and S_2 is hereafter called the *Intersection Value* (IV) of the two, which is defined by:

$$IV_{S_1,S_2} = \frac{A(S_1 \cap S_2)}{A(S_1)} * \frac{A(S_1 \cap S_2)}{A(S_2)}$$

with $A(Set)$ being the area of a set. Clearly, 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.

After finishing the building of the graph all possible paths are checked. A T-Norm is used to evaluate each path yielding an overall coverage degree of the resulting descriptive rule over the original approximative one. The *Coverage Degree* (CD) of a rule R_1 over another rule R_2 is defined by:

$$CD_{R_1R_2} = T\text{-}Norm(IV_{S_{i1},S_{i2}})$$

where $i = 1, \ldots, n$, with n being the number of antecedent variables. A standard T-norm like Min can be used for the implementation of this evaluation.

All valid paths will then become rules and the set of rules obtained will jointly be a translation of the original approximative rule. To reduce the search an IV-threshold may be used to prune the graph by removing nodes of a low IV value. This will help to reduce the number of possible paths and hence speed up the entire graph creation and evaluation process.

For example, the approximative rule given in figure 2 (IF x_1 IS A_1 AND x_2 is A_2 THEN OUTPUT A) leads to the generated graph as shown in figure 3 (without using any IV-threshold) with $IV_{High,A_1} = 0.43$, etc. Using the T-Norm, for instance, the following four CDs are obtained with respect to the four resulting descriptive rules: Min(0.43,0.5)=0.43 (High-Medium), Min(0.43,0.69)=0.43 (High-Small), (0.62,0.5)=0.5 (Medium-Medium) and Min(0.62,0.69)=0.62 (Medium-Small).

Thus, the resultant set of descriptive rules which collectively form the translation of the given approximative rule are:

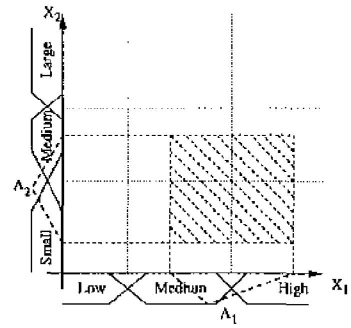

Fig. 2. Approximative rule and descriptive sets

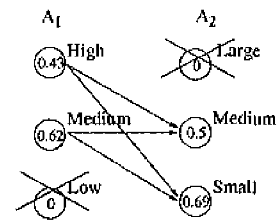

Fig. 3. Graph generation

R_1: IF x_1 IS High AND x_2 IS Medium
THEN OUTPUT A, CD=0.43
R_2: IF x_1 IS High AND x_1 IS Small
THEN OUTPUT A, CD=0.43
R_3: IF x_1 IS Medium AND x_1 IS Medium
THEN OUTPUT A, CD=0.5
R_4: IF x_1 IS Medium AND x_1 IS Small
THEN OUTPUT A, CD=0.62

A.2 Extended Method

An obvious improvement of the method above is to include hedges. In so doing, the number of nodes will, however, increase dramatically. This is because the label of a node can now be any combination of a descriptive set and a hedge used to modify the set. Even if nodes with an IV value below the IV-threshold are eliminated this may still result in an important increase of the nodes in the graph. To reduce such increases, for each layer a similarity measure between any pair of nodes is calculated. A node that is very similar to some others can then be eliminated. For this elimination two different heuristic methods are proposed here.

The first (EHeu 1) ensures that all nodes will not be similar among themselves above a certain threshold (termed S-Threshold). It works via identifying the best IV value node that has a similarity value above the S-Threshold to other nodes. All these other nodes are removed as they have a smaller IV value. After this, the process iterates with the node of the next highest IV value which possesses a similarity value still above the S-Threshold to the remaining nodes.

The second method (EHeu 2) ensures a particular number of nodes per layer. It works by, for each layer, sorting the similarity values of every pair of nodes and removing, in order of high to low similarity, the node of a pair with a smaller IV value, until reaching an allowed number of nodes remaining.

B. GA-based approach

This approach relies on the concept of *functional equivalence*. A method following this novel approach will search for a set of descriptive rules that collectively behave like the original approximative rule to be translated. That is, for the data that is covered by the approximative rule, the found descriptive rules will fire with at least the same firing strength. And, for data that would not cause the original approximative rules to fire, the resultant descriptive rules will either not fire or fire if their consequents comply with the desired output. The search mechanism is herein implemented by a GA.

B.1 Training Sets and Objective Functions

One strategy for translation is that each approximative rule is translated independently. (Assuming that the result of rule firings always takes the consequent that is obtained by executing the rule which has the highest rule firing strength.) Therefore, a training subset of data for each original rule needs to be generated from the global training set. Such rule training subsets are obtained via the following preprocessing process.

Given a set $\mathcal{X}$ of training examples, for each example $x_i \in \mathcal{X}$ the firing strength $AFS_R(x_i)$ of the approximative rule R is determined. If $AFS_R(x_i) > 0$ it would be desirable that, if the consequent of this rule is the same as the desired consequent, the resulting translated descriptive rules will fire for such an example with a strength $DFS_R(x_i)$ equal or greater than $AFS_R(x_i)$. This kind of example will be referred to as type one.

If $AFS_R(x_i) > 0$ but the consequent does not match the desired, then it would be desirable that the firing strengths of the resulting descriptive rules $DFS_R(x_i)$ will 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 is expected to be covered by other 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.

In summary, $\forall x_i \in \mathcal{X}$, the preprocessing aims to allow the genetic algorithm to enforce:

$AFS_R(x_i) > 0$		
same cons.	$DFS_R(x_i) \geq AFS_R(x_i)$	Type 1
diff. cons.	$DFS_R(x_i) \leq AFS_R(x_i)$	Type 2

$AFS_R(x_i) = 0$		
same cons.	ignore x_i	
diff. cons.	$DFS_R(x_i) = 0$	Type 3

In general, a given approximative rule may not be covered by just one descriptive rule. It may happen to obtain several descriptive rules which, collectively, form the functionally equivalent to the original. For this reason, a new objective is needed to minimise the number of descriptive rules used to represent the given approximative rule.

Given the goal of genetic search being to obtain a model that behaves as the original approximative one, if the original model is inaccurate the translated will generally also be. To overcome in part this potential drawback, an extra objective of reducing the model error is also included.

This way, the genetic algorithm will be guided by several objectives as listed below:

Objectives to minimise
$\sum_{x_i \in T_1} max(0, AFS(x_i) - DFS(x_i))$
$\sum_{x_i \in T_2} max(0, DFS(x_i) - AFS(x_i))$
$\sum_{x_i \in T_3} DFS(x_i)$
Number of Rules
Model error

There exist in the genetic algorithms literature several approaches to deal with multiple objectives. In particular, the one used in this work is called "Sum of Weighted Global Ratios" [1]. This method independently normalises each objective first with respect to the best and worst value ever found for it. Then, each objective is weighted and added together into a final fitness value.

For specific application domains, there may be further more objectives requiring consideration (see section IV). Such domain-particular objectives can be readily incorporated with the above mentioned general ones to define the fitness function.

Another strategy of translation valid for problems with a limited range of output values (such as classification problems) is to translate globally all approximative rules with the same output value. The training subsets are calculated in a similar way but using the set of all such equal output approximative rules to calculate the approximative firing strength (AFS).

B.2 Genetic Representation and Genetic Engine

The genetic chromosome representation is based on the work in [10]. The specific codification and use of this DNA-like representation is explained in [12]. It allows a variable number of rules and a variable number of conditions in each rule. The same variable may even appear several times in the same antecedent. The length of the chromosomes is also variable.

The GA adopted is a steady-state one. The parents are chosen by linear rank and the children replace the worst members of the population if they are better. The search stops when the population does not improve for a number of generations. To avoid premature convergence, a minimum number of evaluations is enforced and the mutation rate is kept rather high.

The inference engine works by firing the rule with highest strength. When a translation process is terminated, the rules that never fired are eliminated. The GA uses all data available because the ultimate goal is not to generate a model that generalises but to translate an already existing model.

IV. Results

To demonstrate the proposed approach at work, typical classification problems are used here [16],

TABLE I

EXPERIMENTAL RESULTS

	Thyroid				Glass(Easier)				Glass(Harder)			
	Num. Rules		Error		Num. Rules		Error		Num. Rules		Error	
ANFIS	8		3.7%		25		18.6%		25		18.6%	
TRAP	8		11.6%		24		28%		24		28%	
EHeu 1	9		14.8%		25		35%		14		46.2%	
EHeu 2	11		15.8%		24		45.5%		24		42.5%	
Lozow	19		12.56%		29		49.5%		23		50%	
Lozow	20		12.09%		36		56.0%		28		48.1%	
Genetic Individual												
	MEAN		BEST		MEAN		BEST		MEAN		BEST	
	NR	Err	NR	Err	NR	Err	NR	Err	NR	Err	NR	Err
EH1 Low	10.1	8.2%	9	4.1%	24	36.7%	25	32.7%	20	41.3%	18	32.2%
EH1 Hgh	8.5	10.1%	8	5.1%	20.8	36.6%	19	32.2%	18.7	44.7%	19	38.3%
EH2 Low	10.7	8.9%	11	5.1%	22.1	34.5%	23	27.1%	18.7	41.1%	18	33.6%
EH2 Hgh	8.1	8.8%	9	5.1%	20.8	37.6%	22	29.4%	18.1	43.9%	21	35.5%
Genetic Global												
EH1 Low	7.5	9.2%	6	7.9%	19.3	39%	17	32.2%	17.5	38.7%	14	31.3%
EH1 Hgh	4.5	9.1%	7	5.1%	13.8	38.5%	15	32.2%	10.5	43.8%	11	35.9%
EH2 Low	7.8	9.0%	11	5.1%	17.6	38.7%	18	28.9%	15	40.6%	15	30.8%
EH2 Hgh	4.7	10.2%	5	6.5%	12.1	41.5%	14	32.2%	9.9	45.2%	14	36.4%

including the New Thyroid problem (5 input variables and 1 output variable with 2 classes) and the Glass problem (9 input variables and 1 output variable with 7 classes). Such classification problems require the maximisation of the number of correctly classified samples whilst minimising the number of incorrectly or not covered cases. This introduces the following new objectives, in addition to those general ones given in section III-B.1.

Additional classification objectives
Maximum number of correctly classified
Minimum number of incorrectly classified
Minimum number of not covered

A neuro-fuzzy approximative rule induction algorithm ANFIS [8] using bell-shaped approximative sets were trained for both problems. The ANFIS bell sets were then changed to trapezoidal sets (losing accuracy in this step of course) and the resulting approximative rule set (TRAP) is used as the original set of rules to illustrate the proposed translation methods. The first step of the translation is performed using the extended heuristic method with the two methods outlined in III-A.2. As indicated before, the output of the heuristic method is used as the initial population for the genetic algorithm. The two translation strategies (individual or global) with two different weights (low and high) on the number-of-rules objective were executed. To ensure the readability and understandability of the resulting descriptive rules, the maximum number of hedges allowed to be applied to a given set in the genetic step is limited to 2, as the use of more than two hedges to one fuzzy set makes it very difficult to understand the condition of the resulting rule. For comparison, Lozowski's descriptive induction algorithm [11] (exhaustive search) with different parameter settings were also tested.

Fuzzyfication was proportionally generated taken only into account the size of the universe of discourse, although it is expected in general to be given by experts. For the Glass problem two fuzzification schemes were used to show the fact that fuzzyfication with more labels gives a higher degree of freedom to the translation: an easier schema with 3 labels per variable and a harder one with 2 labels per variable. For the Thyroid problem the scheme with 3 labels per variable is used. Twenty runs for each GA were performed. An overall summary of the results obtained are listed in table I.

For the thyroid problem the TRAP rule set loses a lot of accuracy, becoming a quite poor model. A direct use of the heuristic method, as it is merely based on the intersection between approximative and descriptive sets, always worsens the original. However, the genetic algorithm, based on *functional equivalence* and error reduction, improves the original rule set (TRAP) from which they come and has a performance close to that of ANFIS, with best values almost equal.

The Glass problem presents a rather difficult data set. ANFIS itself got an error of 18.6% and TRAP increases it to 28%. The heuristic method loses big chunks of the original rules (due to the extreme shapes of the original approximative antecedent sets) although still always outperforming Lozowski's algorithm. The GA considerably improves the heuristic results, getting again very close to the performance of the translated approximative rules (TRAP).

Global translation strategy usually underperforms when compared to the individual strategy in terms of classification accuracy for easier problems, but it shows similar performance for harder ones. However, the use of global translation decreases significantly the number of resultant descriptive rules. Low pressure (imposed over the number of rules), as may be expected, produces more accurate results at the cost of an increase in the number of rules. High pressure keeps the number of generated descriptive rules close to the number of the original descriptive rules.

It seems that there is no big difference between the extended heuristic methods which use similarity-based graph pruning. Nevertheless, Method 2 allows a better control over the size of the initial translated rule set (obtained before removing unfired rules).

Lozowski's algorithm is almost always outperformed in terms of both the number of resultant rules and the rules' classification error. The only results with a smaller classification error (see Thyroid against H1 and H2) needs twice the number of rules.

V. Conclusions

A major disadvantage of the existing methods for developing descriptive models is that the generation of the rules is usually made via an exhaustive search throughout the product input space. Also, the rules produced by such methods usually have a low accuracy. However, approximative rule generation methods can be very fast and accurate, through they tend to having the difficulty in conveying the underlying semantics of the data. 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 facilitated by the use of linguistic hedges to modify the prescribed, meaningful descriptive fuzzy sets, such that the modified will closely resemble the original approximative ones (that were created by any standard approximative modelling technique). The modification process is implemented via a *functional equivalence* guided systematic search. Further to this, a fast heuristic algorithm is presented to implement a preliminary translation that will be used as the initial population for an evolutionary algorithm which performs the fine grain modifications and hence the required translation.

The results demonstrate that the translations do not decrease substantially the accuracy of the original approximative models and the descriptive rules obtained are interpretable by humans. The proposed approach outperforms exhaustive search methods for descriptive rule generation, while in the meantime reduce dramatically the search space.

A number of possible improvements can be done to the translation process implemented herein. In particular, a version of ANFIS that directly uses trapezoidals could help to produce better models than TRAP. Also, TRAP may be used to generate the initial population only (due to easy computation of intersections between trapezoidals), whilst the genetic training values of AFS could be obtained from the ANFIS model directly or from N-dimensional ellipsoid approximative sets. Thus, the genetic algorithm would translate the bell-shaped ANFIS or N-dimensional ellipsoids that have shown better approximative results [8], [9].

The genetic algorithm was poorly optimised within the present work. Optimisations to the individual rules may be done to population members periodically during the genetic search. Furthermore, the whole implementation could be integrated with an on-line descriptive modeller that could include micro-tuning of the original fuzzy descriptive partition. Here, by micro-tuning it means a tuning where the modified sets are constraint to keep a very high similarity with the original ones such that the underlying semantics are not disrupted. The modeller might also allow adaptation to new unforeseen data thorough vicinity detection for close but uncovered points and generation and translation of new approximative rules for far away points.

A further possible component of improvement is to build into the GA evaluation function another objective which minimise the number of hedges present in the descriptive rules that are to be created since conditions with less or no hedges are easier to read. Work is currently being carried out along these lines at Edinburgh.

References

[1] P. J. Bentley. *Evolutionary Design by Computers*. Academic Press Ltd, London, 1999.

[2] James C. Bezdek and Sankar K. Pal, editors. *Fuzzy Models for Pattern Recognition: Methods that Search for Structures in Data*. IEEE, Pascataway, 1992.

[3] CL. Chen, SH. Hsu, CT. Hsieh, and WK. Lin. Generating crisp-type fuzzy models from operating data. In *Proceedings of the 7th IEEE International Conference on Fuzzy Systems*, pages 686–691, 1998.

[4] Earl Cox. *The Fuzzy Systems Handbook*. AP Professional, 1994.

[5] A. F. Gómez Skarmeta and F. Jimenez. Generating and tuning fuzzy rules using hybrid systems. In *Proceedings of the 6th IEEE International Conference on Fuzzy Systems*, volume 1, pages 247–252, 1997.

[6] Isao Hayashi, Hiroyoshi Nomura, Hisayo Yamasaki, and Noboru Wakami. Construction of fuzzy inference rules by neural network driven fuzzy reasoning with learning functions. *Int. J. Approximate Reasoning*, 6:241–266, February 1992.

[7] J.-S. R. Jang, Y. C. Lee, and C.-T. Sun. Functional equivalence between radial basis function networks and fuzzy inference systems. *IEEE Transactions on Neural Networks*, 4(1):156–159, January 1993.

[8] J.-S. R. Jang, C.-T. Sun, and E. Mizutani. *Neuro-Fuzzy and Soft Computing*. Matlab Curriculum. Prentice Hall, 1997.

[9] A. Kroll. dentification of functional models using multidimensional reference fuzzy sets. *Fuzzy Sets and Systems*, 80:149–158, 1996.

[10] D.D. Leich. *A New Genetic Algorithm for the Evolution of Fuzzy Systems*. PhD thesis, Dept. of Egineering Science, University of Oxford, 1995.

[11] A. Lozowski, T. J. Cholewo, and J. M. Zurada. Crisp rule extraction from perceptron network classifiers. In *Proceedings of International Conference on Neural Networks*, volume Plenary, Panel and Special Sessions, pages 94–99, Washington, D.C., 1996.

[12] J. Gómez Marín-Blázquez and Q. Shen. Towards the generation of descriptive fuzzy rules via approximative modelling. In *Proceedings of 6th UK Worshop On Fuzzy Systems*, pages 1–14. Brunel University, Sept 1999.

[13] H. Nomura, I. Hayashi, and W. Noboru. A Learning Method of Fuzzy Inference Rules by Descent Method. *Proceedings of the 1st IEEE International Conference on Fuzzy Systems*, pages 203–210, 1992.

[14] Patrick K. Simpson. Fuzzy min-max neural networks-part 1: Classification. *IEEE Transactions on Neural Networks*, 3(5):776–786, September 1992.

[15] M. Sugeno and T. Yasukawa. A fuzzy logic based approach to qualitative modeling. *IEEE Transactions on Fuzzy Systems*, 1(1):7–31, August 1993.

[16] Uci machine learning databases. Available on web: `http://ftp.ics.uci.edu/pub/machine-learning-databases/`.

[17] Lofti A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning i. *Information Sciences*, 8:199–249, 1975.