Genetic Tuning of Fuzzy Rule-Based Systems Integrating Linguistic Hedges *

Jorge Casillas^{a^{\dagger}}, Oscar Cordón^a, María J. Del Jesus^b, Francisco Herrera^a

^aDepartment of Computer Science and A.I., University of Granada, E-18071 Granada, Spain E-mails: {casillas,ocordon,herrera}@decsai.ugr.es

^bDepartment of Computer Science, University of Jaén, E-23071 Jaén, Spain E-mail: mjjesus@ujaen.es

Abstract

Tuning fuzzy rule-based systems for linguistic modeling is an interesting and widely developed task. It involves adjusting the membership functions composing the knowledge base. To do that, as changing the parameters defining each membership function as using linguistic hedges to slightly modify them may be considered.

This contribution introduces a genetic tuning process for jointly making these two tuning approaches. The experimental results show that our method obtain accurate linguistic models in both approximation and generalization aspects.

1 Introduction

The interpretability required from *linguis*tic fuzzy rule-based systems (FRBSs) (also known as Mamdani-type FRBS) makes them less accurate than other approaches such as *Takagi-Sugeno* [13] or *Non-Grid-Oriented* ones (also known as approximate FRBSs) [1]. Nevertheless, improvements in the linguistic model learning process and model structure can be performed allowing it to have more freedom degrees different from the usually considered [3].

Indeed, in the last few years, many approaches have been presented to automati-

cally learn the rule base (RB), constituted by the collection of linguistic rules themselves joined by means of the connective *also*, from numerical information (input-output data pairs representing the system behavior). However, there is not much information about the way to derive the data base (DB), which contains the term sets and the membership functions defining their semantics. A good DB learning would significantly improve the system accuracy. This is why some proposals try to adjust the membership functions involved in the variable fuzzy partitions, which is known as *tuning process*.

Two main approaches have been followed to perform the tuning process. On the one hand, the basic tuning [2, 5, 7, 8, 12] changes the membership functions by fitting the parameters defining their shapes. Within this approach, some proposals even consider the use of different kinds of fuzzy sets [11]. On the other hand, the extended tuning [4, 6] changes the shape of the membership functions by using linguistic hedges [15]. However, no proposals combining both approaches have been considered till now. This contribution aims at introducing a method based on genetic algorithms (GAs) that jointly performs a basic and extended tuning.

The paper is organized as follows: Section 2 is devoted to present the mentioned two different tuning approaches, Section 3 introduces the use of linguistic hedges to change the shape of the membership functions, Section 4 proposes a genetic tuning that combines both approaches, Section 5 shows experimental re-

^{*}This research has been supported by CICYT under project PB98-1319 $\,$

[†]Corresponding author, phone: +34.958.242376, fax: +34.958.243317

sults, and Section 6 outlines some conclusions and future works.

2 Basic and Extended Tuning

Firstly, it seems to be necessary to clarify what we mean with *basic* and *extended* tuning:

• Basic tuning [2, 5, 7, 8, 12]: One of the most common ways of tuning the membership functions is to change the basic parameters defining such functions. For example, if the following triangular-shape membership function is considered:

$$\mu(x) = \begin{cases} \frac{x-a}{b-a}, & \text{if } a \le x < b\\ \frac{c-x}{c-b}, & \text{if } b \le x \le c\\ 0, & \text{otherwise} \end{cases}$$

changing the basic parameters—a, b, and c—will vary the shape of the fuzzy set associated to the membership function (see Figure 1(a)), thus influencing the FRBS performance.

• Extended tuning [4, 6]: Sometimes, more flexible alternative expressions for the membership functions are considered to vary the compatibility degree to the fuzzy sets. For example, a new membership function can be obtained raising the membership value to the power of α , a positive parameter that defines a nonlinear scaling function, i.e.,

$$\mu'(x) = (\mu(x))^{\alpha}, \quad 0 < \alpha,$$

In this case, the tuning process involves adjusting the additional α parameter to improve the FRBS performance (Figure 1(b) shows the effect of this tuning approach). The extended tuning has associated some changes in the model structure to admit the additional parameters. These new expressions will be analyzed in more detail in the next section. In this kind of tuning, the process is usually restricted to a set of values (e.g. $\alpha \in \{0.5, 1, 2\}$) [4, 6] where, moreover of improving the interpretability (as we will see in the next section), the search space is reduced.



Figure 1: Two kinds of tuning for the membership function shapes: (a) basic tuning by changing the basic membership function parameters, (b) extended tuning by changing additional membership function parameters

These tuning processes can be basically applied in two different ways: different parameters for each of the membership functions composing the DB ($DB \ level$) or a different parameter for each membership function involved in each linguistic rule ($RB \ level$).

3 Alternative Model Structure with Linguistic Hedges

To perform extended tuning at RB level, it is necessary to relax the model structure by including certain operators that slightly change the meaning of the linguistic labels involved in the system.

A way to do so without losing the description to a high degree is to use *linguistic hedges* (as Zadeh highlighted in [15]). A linguistic hedge (also known as linguistic modifier) is a function that lets us alter the membership functions for the fuzzy sets associated to the linguistic labels, obtaining a definition with a higher or lower precision depending on the case. Two of the most well known modifiers are the *concentration* linguistic hedge "very" and the *dilation* linguistic hedge "more-orless". Their expressions are

$$\mu_T^{very}(x) = (\mu_T(x))^2 \,,$$

$$\mu_T^{more-or-less}(x) = \sqrt{\mu_T(x)}$$

and their effects on a triangular membership function are showed in Figure 2.



Figure 2: Effects of the linguistic hedges "very" and "more-or-less"

An example of the use of linguistic hedges at RB level would be, for instance, the following:

IF X_1 is very high and X_2 is good THEN Y is more-or-less small.

When extended tuning at RB level is applied using linguistic hedges, the process should be considered as a *tuning of rules*—instead tuning of membership functions—because actually the rule structure is changed (considering the rule structure as the mapping between linguistic variables and linguistic terms).

4 A Basic and Extended Genetic Tuning Process for FRBSs

Basic [2, 5, 7, 8, 12] and extended [4, 6] tuning have been considered under separate cover in the specialized literature. However, no processes combining both approaches have been proposed till now. In this section, a tuning based on GAs will be introduced to jointly fitting the membership functions by changing their basic parameters and fitting the rules using linguistic hedges. The tuning involves starting from a previous RB derived by any linguistic learning method.

The main components of the proposed genetic tuning process are the following:

• A generational GA with the Baker's stochastic universal sampling procedure

together with an elitist selection scheme is considered.

• The objective (fitness function) will be to minimize the well-known *mean square error* (MSE):

MSE =
$$\frac{1}{2 \cdot N} \sum_{l=1}^{N} (Y^l - y^l)^2$$
,

with N being the data set size, Y^l being the output obtained from the FRBS when the *l*-th example is considered, and y^l being the known desired output.

- A double coding scheme $(CS_a + CS_b)$ for both membership function $(CS_a \text{ part})$ and rule $(CS_b \text{ part})$ tuning is used.
 - For the CS_a part, a 3-tuple of real values for each triangular membership function is used, thus being the DB encoded into a real-coded chromosome built by joining the membership functions involved in each variable fuzzy partition. A variation interval to every gene in the CS_a part is used to preserve meaningful fuzzy sets. Figure 3 shows an example of the interval variations considered for each membership function parameter according to the cross points between fuzzy sets.
 - For the CS_b part, the coding scheme generates integer-coded strings of length $N_r \cdot (n + 1)$ (with N_r being the number of rules and n being the number of input variables). Each gene can take any value in the set $\{0, 1, 2\}$ with the following correspondence to the linguistic hedge used: $c_{ij} = 0$ for the "very" linguistic hedge, $c_{ij} = 1$ when no linguistic hedge is used, and $c_{ij} = 2$ for the "more-or-less" linguistic hedge (with c_{ij} being the gene corresponding to the *j*-th variable of the *i*-th rule).



Figure 3: Variation intervals for each membership function parameter to preserve meaningful fuzzy sets

- The **initial population** is generated in the following way:
 - 1. A chromosome that represents the previously obtained DB and RB, i.e., all genes taking the original values in the CS_a part and the allele 1 in the CS_b part.
 - 2. A half of the population is generated with the CS_a part at random (within the variation interval for each gene) whilst the genes in CS_b part will take the value 1.
 - 3. The remaining chromosomes are generated with the original values for the DB in the CS_a part and values at random (within the set $\{0, 1, 2\}$) in the CS_b part.
- The **crossover** operator will depend on the chromosome part where it is applied:
 - In the CS_a part, the max-minarithmetical crossover [7] is considered. Thanks to its formulation, this operator avoids violation of the restrictions imposed by the variation intervals.
 - In the CS_b part, the standard twopoint crossover is used.

The two best chromosomes among the eight (four different CS_a parts combined with two different CS_b parts) offspring obtained will be selected to replace their parents.

- The **mutation** operator will also depend on the chromosome part where it is applied:
 - In the CS_a part, the Michalewicz's non-uniform mutation operator [9] within the variation interval for each gene is considered.
 - In the CS_b part, the mutation operator changes the gene to the allele 1 when a gene with alleles 0 or 2 must be mutated, and randomly to 0 or 2 when a gene with allele 1 must be mutated.

5 The Rice Taste Evaluation Problem

5.1 Introduction

Qualification of rice taste is usually put into effect by means of a subjective evaluation called the *sensory test*. In this test, a group of experts evaluates the rice according to a set of characteristics associated to it. These factors are: flavor, appearance, taste, stickiness, and *toughness*. Because of the large quantity of relevant variables, the problem of rice taste analysis becomes very complex, thus leading to solve it by means of modeling techniques capable of obtaining a model representing the non-linear relationships existing in it. To do so, we are going to use the data set presented in [10]. This set is composed of 105 data vectors collecting subjective evaluations of the six variables in question (the five mentioned and the overall evaluation of the kind of rice), made up by experts on this number of kinds of rice grown in Japan.

With the aim of not biasing the learning, we have randomly obtained ten different partitions of the mentioned set, composed by 75 pieces of data in the training set and 30 in the test one, to generate ten models in each experiment.

5.2 Experimental Results

A model (with 15 linguistic rules) previously obtained by the well-known Wang and Mendel's learning method (WM-method) [14] when solving the rice evaluation problem will be used to be adjusted by our tuning process (BE-tun-method). Its performance will be compared with two different methods where basic (B-tun-method) [5] and extended (Etun-method) [4] tuning are respectively performed at DB and RB levels.

An initial DB constituted by a primary fuzzy partition for each variable will be considered in the WM-method. Every partition is formed by two labels with triangular-shaped equally distributed fuzzy sets giving meaning to them. The values of parameters used in the GA processes are the following: a population size of 61 individuals, 100 generations, 0.6 as crossover probability, 0.2 as mutation probability, 0.35 for the weight factor in the max-min-arithmetical crossover, and 5 for the weight factor in the non-uniform mutation.

Table 1 collects the results obtained by the learning and tuning methods, where MSE_{tra} and MSE_{tst} stand for the values of the MSE function (multiplied by 1,000) obtained over the training and test data sets, respectively. The best results are shown in boldface.

Concerning the computational cost, the tuning methods lasted 93 (B-tun), 37 (E-tun), and 131 (BE-tun) seconds to perform the adjustment in a Pentium II 350 MHz. The reason of these differences mainly lie in the crossover operator employed. Anyway, the algorithm executions took an acceptable time in the linguistic modeling field.

Table 1: Results obtained when solving therice evaluation problem

Method	\mathbf{MSE}_{tra}	\mathbf{MSE}_{tst}
WM	13.284	13.284
WM + B-tun	1.106	2.138
WM + E-tun	1.758	1.731
WM + BE-tun	0.831	1.654

5.3 Analysis of Results

In view of the obtained results, significantly good accuracy degrees—both in approximation (MSE_{tra}) and generalization (MSE_{tst}) are obtained by our tuning proposal.

As may be observed, the genetic tuning that changes the membership functions by fitting their basic parameters (the B-tunmethod) has a clear tendency to overfit the problem. On the contrary, the method that tunes the rules using linguistic hedges (the Etun-method) obtains a good balance of accuracy, but it is not as high as desired.

However, thanks to the combined action of the two tuning approaches done by our method, better results both of approximation and generalization degrees are obtained. On the one hand, the good approximation capability provided by changing the basic parameters is considered while, on the other hand, the use of linguistic hedges makes the obtained model have a good generalization degree as well.

Figure 4 shows the DB and RB generated by the WM+BE-tun method for a specific data set partition of the problem. In that figure, we may observe how the fact of adding linguistic hedges to some linguistic variables of some rules and, at the same time, slightly tuning the membership functions composing the DB, causes the accuracy results to significantly improve. Moreover, thanks to the consideration of linguistic hedges—which give a linguistic interpretation to the changes performed—and to preserve meaningful fuzzy sets, linguistic models with a good interpretability are obtained.

6 Concluding Remarks and Further Works

Two different approaches for tuning the membership functions involved in an FRBSs are usually followed: either changing the parameters that define the fuzzy set shapes or using linguistic hedges in the fuzzy rules. However,



(b) Tuned RB

Rule	Flavor	Appearance	Taste	Stickiness	Toughness	Evaluation
R_1	m- or - l bad	very bad	very bad	<i>m</i> - <i>or</i> - <i>l</i> not-sticky	m- or - l tender	<i>m-or-l</i> low
R_2	very bad	bad	bad	$\operatorname{not-sticky}$	m- or - l tough	low
R_3	m- or - l bad	good	very bad	very not-sticky	<i>very</i> tender	m- or - l low
R_4	very bad	m - or - $l \mod$	m - or - $l \mod$	not-sticky	m- or - l tender	very low
R_5	very bad	m - or - $l \mod$	m - or - $l \mod$	$_{ m sticky}$	m- or - l tender	very low
R_6	m - or - $l \mod$	m- or - l bad	bad	<i>m</i> - <i>or</i> - <i>l</i> not-sticky	m- or - l tender	very low
R_7	good	m- or - l bad	m- or - l bad	very not-sticky	very tough	m- or - l low
R_8	good	m- or - l bad	very good	very not-sticky	m- or - l tender	m- or - l low
R_9	good	good	bad	<i>m</i> - <i>or</i> - <i>l</i> not-sticky	m- or - l tough	low
R_{10}	m - or - $l \mod$	$very \mod$	bad	very not-sticky	m- or - l tender	very high
R_{11}	good	$very \mod$	very bad	sticky	<i>very</i> tender	m- or - l high
R_{12}	m - or - $l \mod$	m - or - $l \mod$	very good	<i>m-or-l</i> not-sticky	tender	very high
R_{13}	good	m - or - $l \mod$	m - or - $l \mod$	<i>m-or-l</i> not-sticky	m- or - l tough	$_{ m high}$
R_{14}	m - or - $l \mod$	$very \mod$	very good	m- or - l sticky	m- or - l tender	m- or - l high
R_{15}	m - or - $l \mod$	$very \mod$	$very \mod$	very sticky	tough	m- or - l high

Figure 4: DB and RB generated by the WM+BE-tun method for a specific data set partition. $MSE_{tra/tst}$ before tuning = 0.014704/0.016700, $MSE_{tra/tst}$ after tuning = 0.000873/0.001314

no tuning methods performing both tasks together have been properly considered.

In this paper, we have introduced a GA for jointly performing the said two tuning approaches. The application of our proposal to a real-world linguistic modeling problem has shown good results appropriately facing one of most usual drawbacks presented in tuning processes: the lack of accuracy when a good generalization degree is claimed.

As further works, with the aim of improving the accuracy at the expense of losing certain interpretability, we propose to employ a more flexible extended tuning by directly using real-coded additional parameters (e.g, any value in the range [0, 5]) instead linguistic hedges (which actually consider only three possible values, $\{0.5, 1, 2\}$). Moreover, consideration of the combined tuning for fuzzy rulebased classification systems should be made.

References

- R. Alcalá, J. Casillas, O. Cordón, and F. Herrera. Building fuzzy graphs: features and taxonomy of learning for non-grid-oriented fuzzy rule-based systems. Technical Report #DECSAI-99-01-17, Department of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain, 1999.
- [2] P. P. Bonissone, P. S. Khedkar, and Y. Chen. Genetic algorithms for automated tuning of fuzzy controllers: a transportation application. In *Proceedings of the 5th IEEE International Conference on Fuzzy Systems*, pages 674–680, New Orleans, USA, 1996.
- [3] J. Casillas, O. Cordón, and F. Herrera. Can linguistic modeling be as accurate as fuzzy modeling without losing its description to a high degree? Technical Report #DECSAI-00-01-20, Department of Computer Science and Artificial Intelli-

gence, University of Granada, Granada, Spain, 2000.

- [4] O. Cordón, M. J. del Jesus, and F. Herrera. Genetic learning of fuzzy rule-based classification systems cooperating with fuzzy reasoning methods. *International Journal of Intelligent Systems*, 13:1025– 1053, 1998.
- [5] O. Cordón and F. Herrera. A threestage evolutionary process for learning descriptive and approximate fuzzy logic controller knowledge bases from examples. *International Journal of Approximate Reasoning*, 17(4):369–407, 1997.
- [6] A. González and R. Pérez. A study about the inclusion of linguistic hedges in a fuzzy rule learning algorithm. *International Journal of Uncertainty*, *Fuzziness and Knowledge-Based Systems*, 7(3):257–266, 1999.
- [7] F. Herrera, M. Lozano, and J. L. Verdegay. Tuning fuzzy controllers by genetic algorithms. *International Journal of Approximate Reasoning*, 12:299–315, 1995.
- [8] C. L. Karr. Genetic algorithms for fuzzy controllers. AI Expert, 6(2):26–33, 1991.
- [9] Z. Michalewicz. Genetic algorithms + data structures = evolution programs. Springer-Verlag, Heidelberg, Germany, 3rd edition, 1996.
- [10] K. Nozaki, H. Ishibuchi, and H. Tanaka. A simple but powerful heuristic method for generating fuzzy rules from numerical data. *Fuzzy Sets and Systems*, 86(3):251– 270, 1997.
- [11] Y. Shi, R. C. Eberhart, and Y. Chen. Implementation of evolutionary fuzzy systems. *IEEE Transactions on Fuzzy Sys*tems, 7(2):109–119, 1999.
- [12] H. Surmann, A. Kanstein, and K. Goser. Self-organizing and genetic algorithms for an automatic design of fuzzy control and decision systems. In *Proceedings*

of the 1st European Congresss on Fuzzy and Intelligent Technologies, pages 1097– 1104, Aachen, Germany, 1993.

- [13] T. Takagi and M. Sugeno. Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 15:116–132, 1985.
- [14] L. X. Wang and J. M. Mendel. Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems*, *Man, and Cybernetics*, 22(6):1414–1427, 1992.
- [15] L. A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning. *Information Science*, 8,8,9:199–249,301–357,43–80, 1975.