Cooperative Coevolution for Linguistic Modeling with Weighted Double-Consequent Fuzzy Rules *

Rafael Alcalá[1]†, Jorge Casillas[2], Oscar Cordón[2], Francisco Herrera[2]
[1]Dept. Computer Science. University of Jaén, E-23071 – Jaén, Spain
[2]Dept. Computer Science and A.I. University of Granada, E-18071 – Granada, Spain

Abstract

This paper presents the use of weighted double-consequent rules for linguistic modeling with the aim of improving the accuracy of the so obtained models. However, the use of these kinds of rules significantly increases the solution search space since weights and double-consequents have to be considered in addition to the traditional approach. To solve this problem, a simple genetic algorithm and a cooperative coevolutionary technique have been proposed and tested with experimental results.

Keywords: linguistic modeling, double-consequent rules, weighted rules, cooperative coevolution

1 Introduction

One of the problems associated with linguistic modeling (LM) is its lack of accuracy in some cases. It is due to the inflexibility of the concept of linguistic variable, which imposes hard restrictions to the rule structure [1]. However, improvements in the LM making more flexible the rule structure can significantly increase the accuracy of the obtained model with a slight description loss [2].

Two specific possibilities to relax the model structure are the following:

- *Use of double-consequent rules*, which involves allowing the model to present rules where each combination of antecedents may have two consequents associated when it improves the model accuracy. This approach has been recently proposed in [5, 12].

- *Consider weighted rules* by giving a certainty factor to each rule involved in the model, i.e., using an additional parameter for each rule that indicates its importance degree in the inference process [4, 13, 20], instead of considering all rules equally important as in the usual case.

It is clear that both possibilities will improve the capability of the model to perform the interpolative reasoning and, thus, its performance. As can be seen, these two approaches are not incompatible and a combination of them would be very interesting. In this way, even more flexible rules would be obtained, thus involving a potential improvement of the accuracy.

This contribution propose to address both approaches designing *weighted double-consequent rules* for LM. This task could be performed by a simple genetic algorithm (GA) [11] that encodes the proposed kind of rule.

However, in LM, the use of weights and double-consequents significantly increases the solution search space since new parameters are considered in addition to the traditional approach.

Recently, an advanced evolutionary technique, the cooperative coevolution [14, 16, 15], has been proposed to solve problems with a large search space by independently evolving two or more species which together comprise solution structures.

We will use this novel technique to generate linguistic models with the said extended structure by using a preliminary fuzzy rule set with a large number of simple and double-consequent rules and coevolving two species, the subset of rules best cooperating and the weights associated to these rules.

The paper is organized as follows. In Section 2, the said specific possibilities to relax the model structure are presented. In Section 3, the weighted double-consequent rule structure is proposed, as well as two concrete evolutionary learning methods. Experimental results are shown in Section 4, whilst some concluding remarks are pointed out in Section 5. Coevolutionary algorithms are briefly presented at Appendix A.

*Supported by the Spanish CICYT, project PB98-1319
†Corresponding author, e-mail: alcala@ujaen.es

2 Preliminaries

2.1 Double-Consequent Fuzzy Linguistic Rules

More flexible rules may be obtained allowing the linguistic model to present rules where each combination of antecedents may have two consequents (linguistic terms of the output variable) associated [5, 12]. The rule structure obtained is:

IF X_1 is A_1 and ... and X_n is A_n
THEN Y is $\{B_1, B_2\}$,

with X_i (Y) being the linguistic input (output) variables, A_i being the linguistic label used in the i-th input variable, and B_1 and B_2 the two linguistic terms associated to the output variable.

The use of two consequents has no influence on the linguistic model inference system. The only restriction imposed is that the defuzzification method must consider the matching degree of the rules fired, for example, the *center of gravity weighted by the matching degree* defuzzification strategy [6] may be used.

We should note this structure does not constitute an inconsistency from the LM point of view but only a shift of the main labels making the final output of the rule lie in an intermediate zone between both consequents.

The consideration of this structure to generate advanced linguistic models is initially proposed in [12]. Another approach, according to the Accurate Linguistic Modeling (ALM) methodology, is introduced in [5]. This methodology consist of two steps:

1. Firstly, two rules, the primary and secondary in importance, are obtained in each combination considering a specific generation process. In this contribution, the generation process proposed by Wang and Mendel [18] is considered.

2. Then, after decomposing each double-consequent rule into two independent simple ones, the selection process proposed in [10] is employed to select a subset of rules presenting good cooperation.

   It is based on a binary-coded GA where each gene indicates if the corresponding rule is considered or not to belong to the final rule base. Appropriate selection and recombination operators are used.

2.2 Weighted Fuzzy Linguistic Rules

Another possibility to extend the classical model structure making it more flexible is to use weighted rules [4, 13, 20]. Its structure is:

IF X_1 is A_1 and ... and X_n is A_n
THEN Y is B *with* $[w]$,

with w being the real-valued rule weight, and *with* the operator that attaches a weight to a rule.

With this structure, the fuzzy reasoning must be extended. A possibility is to infer with the FITA (First Infer, Then Aggregate) scheme [6] and compute the defuzzified output as the following *weighted sum*:

$$y_0 = \frac{\sum_i m_i \cdot w_i \cdot P_i}{\sum_i m_i \cdot w_i},$$

with y_0 being the crisp value obtained from the defuzzification process, m_i the matching degree of the rule i, w_i the weight associated to the rule i, and P_i the characteristic value of the output fuzzy set corresponding to the rule i. In this contribution, the center of gravity will be considered as characteristic value [6].

These weights are usually used to handle inconsistencies with advanced inference methods [20] or neural networks [4]. Moreover, some proposals make use of them to improve the model accuracy with an automatic learning of weights using different techniques such as gradient descent processes [13].

3 Evolutive Learning of Weighted Double-Consequent Rules

This section proposes the use of a more flexible model structure that combine the two said approaches, thus having weighted double-consequent rules with the following structure:

IF X_1 is A_1 and ... and X_n is A_n
THEN Y is $\{B_1, B_2\}$ *with* $[w_1, w_2]$,

with w_1 and w_2 being the weights associated to the rules composed using the consequents B_1 and B_2, respectively. Therefore, a weighted double-consequent rule can be seen as two weighted single-consequent rules with the same antecedent and different consequents.

To generate linguistic models with this new structure, we may follow an operation mode similar to the said ALM methodology [5], but including the weight learning. Therefore, after performing the first step of the ALM methodology, where an initial set of numerous double-consequents rules is generated, the two following tasks must be performed:

- Selection of a subset of rules presenting good cooperation.
- Learning of the weights associated to these rules.

These interdependent tasks significantly increases the search space making the selection of the used search technique crucial.

In the following two subsections, we introduce two evolutionary approach to our learning problem: a simple GA (evolving the said tasks together) and a cooperative coevolutionary algorithm (simultaneously evolving both tasks).

3.1 A Simple Genetic Algorithm

A simple GA performing the rule selection together with the learning of weights was developed as first approximation to the problem. Generational and **steady-state** [19] approaches was considered, obtaining best results in the latter case.

A double **coding scheme** ($CS_1 + CS_2$) for both *rule selection* and *weight learning* is used:

- For the CS_1 part, the coding scheme generates binary-coded strings of length m (number of single-consequent rules in the rule set previously derived). Depending on a rule is selected or not, the alleles "1" or "0" will be respectively used in the corresponding gene.
- For the CS_2 part, the coding scheme generates real-coded strings of length m. The value of each gene indicates the weight used in the corresponding rule. These values may take any value in the interval $[0, 1]$.

The **initial pool** is generated with an individual having all genes with the value "1" for both coding schemes, and the remaining individuals with values at random.

The **crossover** operator will depend on the chromosome part where it is applied: in the CS_1 part, the standard two-point crossover is used; in the CS_2 part, the max-min-arithmetical crossover [9] is considered.

As regards the **mutation** operator, it changes to the opposite value of the gene in the CS_1 part, and to a value at random within the interval $[0, 1]$ in the CS_2 part.

The results obtained in a preliminary experimentation were not satisfactory, presenting even a worse behavior than a similar approach considering only weighted rules. This fact conflicts with the intuitive idea that a more flexible structure should provide more accurate models. Thus, the problem seems to be in the search process, that is not powerful enough to find better solutions.

Indeed, as David Goldberg stated, the integration of single methods into hybrid intelligent systems goes beyond simple combinations. For him, the future of Computational Intelligence "lies in the careful integration of the best constituted technologies" and subtle integration of the abstraction power of fuzzy systems and the innovating power of genetic systems requires a design sophistication that goes further than putting everything together [8].

Therefore, a more advanced optimization technique is needed. The following section presents a new algorithm within the cooperative coevolutionary paradigm that properly achieves our objective.

3.2 The Cooperative Coevolutionary Algorithm

As we have seen, the problem that concern us can be easily decomposed in two subtasks, the rule selection and the learning of weights. Therefore, it can be solved coevolving two species cooperating to form the complete solution by selecting a set of weighted rules. In the following subsections, the main characteristics of the proposed cooperative coevolutionary algorithm are presented. A brief introduction to the evolutionary paradigm is presented in Appendix A.

3.2.1 Interaction Scheme Between Species

Let F_{ij} be the model obtained by combining the chromosomes i and j of the species 1 (selected rules) and 2 (weights). The objective will be to minimize the well known *mean square error* (MSE):

$$\mathrm{MSE}_{ij} = \frac{1}{2 \cdot N} \sum_{l=1}^{N} (F_{ij}(x^l) - y^l)^2 \ ,$$

with N being the number of training data, $F_{ij}(x^l)$ being the model inferred output when the input x^l is presented, and y^l the known desired output.

Thus, individuals in the species 1 and 2, are respectively evaluated with the fitness functions f_1 and f_2, defined as follows:

$$f_1(i) = \min_{j \in R_2 \cup P_2} \mathrm{MSE}_{ij}$$
$$f_2(j) = \min_{i \in R_1 \cup P_1} \mathrm{MSE}_{ij} \ ,$$

with i and j being individuals of species 1 and 2 respectively, R_1 and R_2 being the set of the fittest individuals in the previous generation of the species 1 and 2 respectively, and P_1 and P_2 being individual sets selected at random from the previous generation of the species 1 and 2 respectively.

Whilst the sets $R_{1|2}$ allow the best individuals to influence in the process guiding the search towards good solutions, the sets $P_{1|2}$ introduce diversity in the search. The combined use of both kinds of sets make the algorithm have a trade-off between exploitation ($R_{1|2}$) and exploration ($P_{1|2}$). The cardinalities of the sets $R_{1|2}$ and $P_{1|2}$ are previously defined by the designer.

A **generational GA** [11] scheme is followed in both species. Baker's stochastic universal sampling procedure together with an elitist mechanism that ensures to select the best individual of the previous generation are used.

The specific operators employed in each species are described in the following sections.

3.2.2 Species 1: Selecting Fuzzy Rules

For the species 1, we will use the GA-based method to select rules proposed in [5]. The used coding scheme was introduced in Section 3.1 as the CS_1 part. Thus, a chromosome c_1^p will be a binary vector representing the subset of rules finally obtained.

The **initial pool** is generated at random except for one individual, which represents the complete previously obtained rule set:

$$\forall k \in \{1, \ldots, m\},\ c_1^1[k] = 1,$$

For this species, the standard two-point **crossover** operator is used. As regards the **mutation** operator, it changes to the opposite value of the gene.

3.2.3 Species 2: Learning weights

In this case, the coding scheme also was introduced in Section 3.1 as the CS_2 part. Now, a chromosome c_2^p will be a real-valued vector representing the weights associated to the rules considered.

The **initial pool** for this species is generated with a chromosome having all the genes with the value "1", and the remaining individuals with values at random within the variation interval $[0,1]$:

$$\forall k \in \{1, \ldots, m\},\ c_2^1[k] = 1.0,$$

The max-min-arithmetical **crossover** operator [9] is considered. As regards the **mutation** operator, it simply involves changing the value of the selected gene by other value obtained at random within the interval $[0,1]$.

4 Experimental Results in the Electrical Maintenance Cost Estimating Problem

This experimental study will be devoted to analyze the behavior of the proposed method to learn weighted double-consequent rules by using cooperative coevolutionary algorithms (WALM-CC). With this aim, we have chosen the problem of estimating the maintenance costs of the medium voltage electrical network in a town [7].

We will analyze the accuracy of the linguistic models generated from the proposed process compared to the four following methods: the well-known ad hoc data-driven method proposed by Wang and Mendel (WM) [18], the said ALM method [5], a simple GA that learns the weights of the rule set derived by the WM method (WRL), and the method proposed in Section 3.1 (WALM).

With respect to the model reasoning method used, we have selected the *minimum t-norm* playing the role of the implication and conjunctive operators, and the *center of gravity weighted by the matching* strategy acting as the defuzzification operator [6].

4.1 Problem Description

Estimating the maintenance costs of the medium voltage electrical network in a town [7] is a complex but interesting problem. Since an actual measure is very difficult to obtain, the consideration of models becomes useful. These estimations allow electrical companies to justify their expenses. Moreover, the model must be able to explain how a specific value is computed for a certain town. Our objective will be to relate the *maintenance costs of medium voltage line* with the following four variables: *sum of the lengths of all streets in the town*, *total area of the town*, *area that is occupied by buildings*, and *energy supply to the town*.

We will deal with estimations of minimum maintenance costs based on a model of the optimal electrical network for a town in a sample of 1,059 towns.

To develop the different experiments in this contribution, the sample has been randomly divided in two subsets, the training and test ones, with an 80%-20% of the original size respectively. Thus, the training set contains 847 elements, whilst the test one is composed by 212 elements. *Five linguistic terms* for each variable are considered.

4.2 Experimental Results and Analysis

The following values have been considered for the parameters of each method:

- *Rule selection step of the ALM*: 61 individuals, 500 generations, 0.6 as crossover probability, and 0.1 as mutation probability.
- *WRL*: 61 individuals, 1,000 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.
- *WALM*: 61 individuals, 1,000 generations, 0.25 as mutation probability, and 0.35 for the weight factor of the crossover operator.
- *WALM-CC*: 62 individuals (31 for each species), 1,000 generations, 0.6 and 0.2 for the crossover and mutation probabilities in both species respectively, 0.35 and 5 for the weight factors of the crossover and mutation operators in the species 2 respectively, the three fittest individuals ($|R_{1|2}| = 3$) and two random individuals ($|P_{1|2}|$=2) of each species are considered for the coupled fitness.

The results obtained by the five methods analyzed are shown in Table 1, where $\#R$ stands for the number of rules, and MSE_{tra} and MSE_{tst} respectively for the error obtained over the training and test data. The best results are shown in boldface.

In view of the obtained results, we can see that the ALM and WRL methods — which are respectively based on double-consequent rules and weighted rules — present significant improvements over the WM method.

As regards the WALM method, the results obtained were worse than the ones obtained with WRL, when theoretically the former should obtain better results. It confirms the need of a more advanced optimization technique. Indeed, analyzing the model obtained by the WALM-CC method, we can conclude that it presents the best performance in both approximation (MSE_{tra}) and generalization (MSE_{tst}). It is due to its ability to tackle decomposable complex problems.

Table 1: Results obtained in the electrical problem

Method	**#R**	**MSE**$_{tra}$	**MSE**$_{tst}$
WM	66	71,294	80,934
ALM	**37**	64,889	70,850
WRL	66	33,639	33,319
WALM	58	38,919	40,476
WALM-CC	59	**24,961**	**28,225**

5 Concluding Remarks and Further Works

In this paper, a new model structure using weighted double-consequent rules has been proposed with the aim of improving the performance of the so obtained models. Its main interest lies in the fact of making more flexible the model structure in a different way of the usually considered ones (e.g., learning fuzzy membership functions).

The proposed model involves, however, tackling an enlarged search space, thus being the selection of a proper technique crucial. To address this fact, a cooperative coevolutionary method have been proposed. Its good accuracy results, compared with other related approaches, has been contrasted when solving a real-world problem.

As further works, we propose to adding other mechanisms that make more flexible the linguistic models, for example with the learning method of membership functions proposed in [3]. Thus, even more accurate linguistic models could be obtained.

A Coevolutionary Algorithms

GAs [11] are general-purpose global search algorithms that use principles inspired by natural population genetics. In a GA, each individual in the population represents a candidate solution to the problem and has an associated *fitness* to determine which individuals are used to form new ones in the process of competition. The new individuals are created using genetic operators such as crossover and mutation.

Within this field, a new paradigm has been recently proposed, coevolutionary algorithms [14]. They involve two or more species (populations) that permanently interact among them by a coupled fitness. Thereby, in spite of each species has its own coding scheme and reproduction operators, when an individual must be evaluated, its goodness will be calculated considering some individuals of the other species. This coevolution makes easier to find solutions to complex problems.

Different kinds of interactions may be considered among the species according to the dependencies existing among the solution subcomponents. Generally, we can mention two different kinds of interaction:

- *Competitive coevolutionary algorithms* [17]: Those where each species competes with the remainder. In this case, increasing the fitness of an individual in a species implies decreasing the fitness of the ones other species, i.e., the success of somebody else entails the personal failure.
- *Cooperative* or *symbiotic coevolutionary algorithms* [16]: Those where all the species cooperate to build the problem solution. In this case, the fitness of an individual depends on its ability to cooperate with individuals from other species.

Therefore, the use of cooperative coevolutionary algorithms is recommendable when the following issues arise [15]:

1. the search space is huge,
2. the problem may be decomposable in subcomponents,
3. different coding schemes are used, and
4. there is strong interdependencies among the subcomponents.

References

[1] A. Bastian, How to handle the flexibility of linguistic variables with applications, Int. J. Uncertainty, Fuzziness and Knowlegde-Based Syst. 2:4 (1994) 463–484.

[2] J. Casillas, O. Cordón, 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, Dept. Computer Science and A.I., University of Granada, Granada, Spain, 2000.

[3] J. Casillas, O. Cordón, F. Herrera, A cooperative coevolutionary algorithm for jointly learning fuzzy rule bases and membership functions, summitted to the International Conference in Fuzzy Logic and Technology (EUSFLAT), Leicester, England, 2001.

[4] J.-S. Cho, D.-J. Park, Novel fuzzy logic control based on weighting of partially inconsistent rules using neurtal network, J. Intelligent and Fuzzy Syst. 8 (2000) 99–110.

[5] O. Cordón, F. Herrera, A proposal for improving the accuracy of linguistic modeling, IEEE Trans. Fuzzy Syst. 8:3 (2000) 335–344.

[6] O. Cordón, F. Herrera, A. Peregrín, Applicability of the fuzzy operators in the design of fuzzy logic controllers, Fuzzy Sets Syst. 86:1 (1997) 15–41.

[7] O. Cordón, F. Herrera, L. Sánchez, Solving electrical distribution problems using hybrid evolutionary data analysis techniques, Applied Intelligence 10 (1999) 5–24.

[8] D. Goldberg, A meditation on the computational intelligence and its furture, Technical Report IlliGAL, Report #2000019, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, Foreword Proc. 2000 Int. Symposium Computational Intelligence.

[9] F. Herrera, M. Lozano, J.L. Verdegay, Tuning fuzzy controllers by genetic algorithms, Int. J. Approximate Reasoning 12 (1995) 299–315.

[10] F. Herrera, M. Lozano, J.L. Verdegay, A learning process for fuzzy control rules using genetic algorithms, Fuzzy Sets Syst. 100 (1998) 143–158.

[11] Z. Michalewicz, Genetic algorithms + data structures = evolution programs, Springer-Verlag, Heidelberg, Germany, 1996.

[12] K. Nozaki, H. Ishibuchi, H. Tanaka, A simple but powerful heuristic method for generating fuzzy rules from numerical data, Fuzzy Sets Syst. 86 (1997) 251-270.

[13] N.R. Pal, K. Pal, Handling of inconsistent rules with an extended model of fuzzy reasoning, J. Intelligent and Fuzzy Syst. 7 (1999) 55–73.

[14] J. Paredis, Coevolutionary computation, Artificial Life 2 (1995) 355–375.

[15] C.A. Peña-Reyes, M. Sipper, Fuzzy CoCo: a cooperative coevolutionary approach to fuzzy modeling, IEEE Trans. Fuzzy Syst. (to appear). Draft available at `http://lslwww.epfl.ch/~penha`.

[16] M.A. Potter, K.A. De Jong, Cooperative coevolution: an architecture for evolving coadapted subcomponents, Evolutionary Computation 8:1 (2000) 1–29.

[17] C.D. Rosin, R.K. Belew, New methods for competitive coevolution, Evolutionary Computation 5:1 (1997) 1–29.

[18] L.X. Wang, J.M. Mendel, Generating fuzzy rules by learning from examples, IEEE Trans. Syst. Man. Cybern. 22 (1992) 1414–1427.

[19] D. Whitley, J. Kauth, GENITOR: A different genetic algorithm, Proc. Rocky Mountain Conf. on A.I., Denver (1988) 118–130.

[20] W. Yu, Z. Bien, Design of fuzzy logic controller with inconsistent rule base, J. Intelligent and Fuzzy Syst. 2 (1994) 147–159.