Learning compact fuzzy rule-based classification systems with genetic programming

Francisco José Berlanga, María José del Jesus
Department of Computer Science
University of Jaén
Jaén E-23071, Spain
berlanga,mjjesus@ujaen.es

Francisco Herrera
Department of Computer Science and Artificial Intelligence
University of Granada
Granada E-18071, Spain
herrera@decsai.ugr.es

Abstract

The inductive learning of a fuzzy rule-based classification system (FRBCS) with high interpretability is made difficult by the presence of a large number of features that increases the dimensionality of the problem being solved. The difficult comes from the exponential growth of the fuzzy rule search space with the increase in the number of features considered.

In this paper we propose a genetic-programming-based method, where the disjunctive normal form (DNF) fuzzy rules compete in order to obtain an FRBCS with high interpretability and accuracy. The good results obtained with several classification problems support our proposal.

Keywords: Classification, high-dimensionality problems, genetic programming, fuzzy rules, high interpretability.

1 Introduction

One of the most important areas for the application of the fuzzy set theory are Fuzzy Rule-Based Systems (FRBSs). FRBSs have been successfully applied to various fields such as control, modelling and classification. While traditionally the main goal in the design of FRBSs has been the maximization of the precision, their interpretability has also been taken into account in some recent studies [3].

Regarding the interpretability of FRBSs, the difficulty comes from the exponential growth of the fuzzy rule search space with the increase in the number of features considered. This growth makes the learning process more difficult and, in most cases, it leads to an FBRCS with a rule base with a high cardinality, that decreases the interpretability of the system.

This problem can be tackled in different ways, a) compacting and reducing the rule set as a postprocessing approach ([15] [18]), and b) carrying out a feature selection process, that determines the most relevant variables before or during the inductive learning process of the FRBCS. Several feature selection processes have been proposed involved in the learning of FRBCSs ([2][4][12][20]).

In this paper, we tackle the learning of FRBCSs with high interpretability by means of a genetic-programming (GP) based approach. The definition of a context-free grammar that allows the learning of disjunctive normal form (DNF) fuzzy rules, together with the use of a competition mechanism between rules which deletes redundant rules during the learning process, can allow the obtaining of compact FRBCSs (with few rules and conditions per rule) with a high-generalization capability.

In order to do that, the paper is organized as follows. A brief revision of the evolutionary learning of FRBCSs is shown in section 2. Components of our proposal are explained in Section 3. Section 4 shows the results of the experimentation and the analysis carried out. Finally, the conclusions obtained are presented in Section 5.

2 Evolutionary learning of fuzzy rule-based classification systems

In recent years, the investigation into fuzzy systems has evolved into a more general framework, that considers the integration of fuzzy logic with other techniques such as evolutionary algorithms. An example is the evolutionary fuzzy rule-based systems in which the design process is treated as an optimisation or a search problem that the evolutionary algorithm must solve.

Genetic algorithms (GAs) are currently considered as one of the most well known and employed search techniques, so numerous authors have used them for the learning of FRBCSs. A extensive state of the art study can be found in [8].

The genetic programming (GP) [17] is another kind of evolutionary algorithm that uses variable-length trees to represent the different individuals in the population, instead of fixed-sized vectors (with binary, integer or real codification) as the GAs do.

Different proposals can be found for using the GP paradigm to evolve fuzzy rule sets. An initial paper in this topic is Fuzzy GP, developed by Geyer-Schulz [10], which combines a simple GA that operates on a context-free language with a context-free fuzzy rule language. Sanchez et al. propose an FRBCS learning process in [21] and [9] by combining GP operators with simulated annealing and GA respectively to establish the membership functions. Tsakonas et al. propose in [22] a GP-based algorithm for the learning of crisp and fuzzy rule-based classification systems. Other authors have also used the GP for the learning of FRBS for modelling (see [1] and [14], for instance).

3 Proposal of fuzzy rule-based classification systems learning using genetic programming

In this section, the principal components of the proposed method are completely explained.

In our algorithm we use a specific type of fuzzy rule whose structure, the grammar definition to generate it and the way to define the linguistic terms used are described in the next subsections.

The proposal includes a mechanism to maintain the diversity of the population and three genetic operators explained in subsection 3.5 and 3.6. Once the evolutionary process is finalized, a rule base simplification and certainty degree calculation processes are executed to improve the FRBCS.

3.1 Representation of the solution

One of the most important aspects in the design of any rule evolutionary learning process is the genetic representation of the solution. The different evolutionary methods follow two approaches in order to encode rules within a population of individuals [8]:

- The "Chromosome = Set of rules", also called the *Pittsburgh* approach, in which each individual represents a rule set.
- The "Chromosome = Rule" approach, in which each individual codifies a single rule, and the whole rule set is provided by combining several individuals in the population.

In turn, within the "Chromosome = Rule" approach, there are there generic proposals:

- The *Michigan* approach, in which each individual codifies a single rule. This kind of system is usually called a learning classifier system. It is rule-based, message-passing system that employs reinforcement learning and the GA to learn rules that guide its performance in a given environment [16].
- The *IRL (Iterative Rule Learning)* approach, in which each chromosome represents a rule, but the solution is the best individual obtained and the global solution is formed by the best individuals obtained when the algorithm is run multiple times. MOGUL [7] and SLAVE [13] are proposals that follow this approach.
- The "*cooperative-competitive*" approach, in which the complete population or a subset of it codifies the rule base. REGAL [11] and LOGENPRO [24] are examples with this kind of representation.

Our method follows the cooperative-competitive approach and includes a mechanism to maintain the diversity of the population, that will be described in section 3.5.

3.2 Grammar definition

Once the representation of the solution has been clarified, the next step consists of the definition of a grammar according to the problem to be solved, that allows the learning of one fuzzy rule per individual. This grammar, must also allow the learning of disjunctive normal form (DNF) fuzzy rules, which have the following form:

If X_1 is $\hat{A}_1$ and ... and X_n is $\hat{A}_n$ then Y is C

where each input variable X_i takes as a value a set of linguistic terms $\hat{A}_i = \{A_{i1}$ or … or $A_{iLi}\}$ joined by a disjunctive operator, whilst the output variable (Y) has one of the class values.

This rule also includes a certainty degree (CD $\in [0,1]$), which represents the confidence of the classification in the class represented by the consequent. This certainty degree is calculated according to the ratio of positive examples covered by the rule, and it is not learned during the evolutionary process but is calculated at the end of this, following the process explained in section 3.8.

In Table 1, an example of the grammar for a classification problem with two features (X_1, X_2), three linguistic labels per feature (Low, Medium, High) and three classes (C1, C2, C3) is shown. This grammar also allows the learning of DNF rules and the absence of some input features.

Table 1: Grammar example

Start ➔ [If] antec [then] conseq.
antec ➔ descriptor1 [and] descriptor2.
descriptor1 ➔ [any].
descriptor1 ➔ [X_1 is] label.
descriptor2 ➔ [any].
descriptor2 ➔ [X_2 is] label.
label ➔ {member (?a, [L, M, H, L $\vee$ M, L $\vee$ H, M $\vee$ H, L $\vee$ M $\vee$ H])}, [?a].
conseq ➔ [Class is] descriptorClass.
descriptorClass ➔ {member (?a, [C1, C2, C3])}, [?a].

3.3 Data base definition

The definition of the fuzzy sets that specify the meaning of each term used in the grammar, is done before the evolutionary process begins by using expert knowledge, or in its absence, by using triangular fuzzy sets divided in a uniform way.

3.4 Fitness function

In our method, a fitness function based on the estimation of the two next measures has been used:

1. *Confidence*: It measures the accuracy of an individual, that is, the confidence of the consequent to be true under the antecedent

$$\text{confidence} = \frac{tp}{(tp+fp)} x \frac{tn}{(fn+tn)}$$

where *tp* is the number of true positives, *fp* is the number of false positives, *tn* is the number of true negatives and *fn* is the number of false negatives.

2. *Support*: It measures the coverage of the knowledge represented in the individual.

$$\text{support} = \frac{tp}{(tp+fn)} x \frac{tn}{(fp+tn)}$$

Both measures are combined to form the fitness function in the following way.

$$raw\_fitness = \begin{cases} \text{support}, & \text{if support < min\_support} \\ \text{support} \times \text{confidence}, & \text{otherwise} \end{cases}$$

If the support of the rule is below a user-defined minimum threshold, the confidence value should not be considered to avoid the waste of effort to evolve those individuals with a high confidence but low support.

3.5 Maintaining the diversity of the population

Several approaches have been designed to maintain the diversity using GAs (crowding, fitness sharing, etc). These approaches are based on two main principles:

1. The parents should be among the most similar individuals to the offspring.
2. The estimate of some similarity measure between individuals.

However, these ideas present problems when used with GP, since the parents and the offspring could be totally different due the variable-length nature of the individuals. Furthermore, it is much more complex to calculate how one individual is similar to another individual in GP. To solve these problems, it is necessary to use an approach which does not take into account the individual structure. In our method, the approach used is the Token Competition [24].

It is assumed that each example in the training set can provide a resource called a token, for which all chromosomes in the population will compete to capture. If an individual (rule) can match the example, it sets a flag to indicate that the token is seized. Other weaker individuals then cannot get the token.

The priority of receiving tokens is determined by the strength of the individuals. The individuals with a high fitness score can exploit the niche by seizing as many tokens as they can. The other ones entering the same niche will have their strength decreased because they cannot compete with the stronger ones. The fitness score of each individual is modified based on the tokens it can seize. The modified fitness is defined as:

$$\text{Modified\_fitness} = \text{raw\_fitness} \times \frac{\text{count}}{\text{ideal}}$$

where *raw_fitness* is the fitness score obtained from the evaluation function, *count* is the number of tokens that the individual actually seized and *ideal* is the total number of tokens that it can seize, which is equal to the number of examples that the individual matches.

As a result of token competition, there exist individuals that cannot seize any token. These individuals are redundant as all of their examples are already covered by other stronger individual and, hence, they can be replaced by new individuals.

3.6 Genetic operators

Offspring are generated by one of the next three genetic operators:

1. *Crossover*: Produces one child from two parents. A part in the first parent is randomly selected and replaced by another randomly part in the second one, but under the constraint that the offspring produced must be valid according to the grammar.
2. *Mutation*: A part of the rule is selected and replaced by a randomly generated new part. Since the offspring have to be valid according to the grammar, a selected part can only mutate to another part with a compatible structure.
3. *Dropping Condition*: Due the probabilistic nature of GP, redundant constraints may be generated in the rule. Thus, it is necessary to generalize the rules, to represent the actual knowledge in a more concise form. Dropping condition selects randomly one descriptor in the antecedent part and then turns it into "any". The attribute in the descriptor is no longer considered in the rule, hence, the rule can be generalized.

3.6 Evolutionary process

An initial population of rules is randomly generated, according to the grammar production rules.

In each iteration, parents are selected to generate offspring by the ranking selection scheme. The number of new individuals evolved is equal to the initial population size.

Individuals in the population and offspring obtained by the application of the genetic operators, are joined to form a new population. The size of the resulting population is double the original. The individuals of this population are ordered by their fitness score and the token competition is carried out. As a result of token competition, some individuals will have their fitness modified to zero, hence they would be replaced or eliminated from the population. Finally, the individuals in the population are ordered again by their fitness score, and the population size is set to its original one.

Because individuals can be eliminated from the population, the size of the final population can be smaller than the initial one. This shows that the proposed method is available to get reduced and compact fuzzy rule sets, which contain only the necessary rules to cover the whole training set. Our method is clearly orientated to get FRBCS with high interpretability without a significant performance loss.

3.7 Rule base simplification

Once the evolutionary process has finished, a post-processing step is carried out for eliminating redundant rules. During the rule base learning process it may happen that the algorithm learns two rules, where one is included in the other. For example in the following two rules

R1: If X_1 is Low then Class is C1

R2: If X_1 is Low $\vee$ Medium then Class is C1

the second rule includes the first one, hence, it does not make sense to keep both of them in the rule set. In this case, the logic solution deletes the first rule because the examples that it covers are also covered by the second rule.

This process aims to increase the interpretability of the previously learned FRBCS, by deleting redundant rules.

3.8 Certainty degree calculation

Finally, our method ends with the certainty degree calculation for each individual (rule) in the population.

The certainty degree is obtained as the quotient S_j / S, where S_j is the sum of the matching degrees for the training examples belonging to class represented by the consequent which are covered by the antecedent of the rule, and S the sum of the matching degrees for all the training examples which are covered by the antecedent of the rule, independently of the class they belong to.

4 Experimental study

In order to analyse the behaviour of the proposed method, an experimental study has been carried out using the Wisconsin, Lung Cancer and Thyroid databases (Table 2) from the *UCI repository of machine learning Databases* (http://www.ics.uci.edu/~mlearn/MLRepository.html).

Table 2: UCI Databases

Name	N of Features	N of Examples
Wisconsin	9	699
Lung Cancer	56	32
Thyroid	21	7200

For each different database, we have used 10-fold cross-validation procedure to estimate the FRBCS error.

Our method (from now on called FRBCS_GP) has been compared to other fuzzy rule learning techniques and with a decision tree method:

1. *Wang & Mendel*: In [23] it is proposed a fuzzy control rule learning method, which Chi et al. extend for classification problems [5], that generates a fuzzy rule for each example in the training set and does not carry out any feature selection process.
2. *Ravi et al.*: In [20] a process for deriving fuzzy rules for high-dimensionality classification problems is proposed. This approach uses a more reduced set of features extracted from the original ones by the principal component analysis.
3. *SLAVE:* In [13] a GA-based (with binary codification) method for the learning of DNF fuzzy rules is proposed. In [12], this method is extended by the inclusion of a feature selection process. This extension will be called 2SLAVE from now.
4. *Tsakonas et al.*: In [22] a GP-based FRBCS learning process is proposed. This proposal uses a Pittsburgh approach to represent the solutions.
5. *C4.5*: It is a classification algorithm proposed by Quinlan [19] as an extension of his previously proposed ID3 algorithm. It is based on information theory and it also includes a feature selection method. This algorithm uses divide-and-conquer method and the information gain measure for constructing a decision tree, which can be later transformed into a crisp rule set.

The parameters of our algorithm are the following: It stops after 1000 iterations, initial population size is 20, crossover probability is 0.5, mutation probability is 0.4, dropping condition probability is 0.1, and the minimum threshold for the support used in fitness function is 0.01. We have used 5 linguistic labels per variable in all the experiments.

The results are showed in Table 3, where #R indicates the average rule number, #Var the average antecedent variables per rule, #Cond the average antecedent conditions number per rule and the %Test the correct percentage with test examples. The subscripts in %Test are related to the fuzzy reasoning method (FRM) used, so 1 corresponds to the classical FRM (max-min) and 2 with the normalised sum [6] respectively (except in the C4.5 algorithm in where FRM is not used, so it has been decided to place the results in the first of the last two columns).

Table 3: Databases results

Wisconsin

Method	#R	#Var	#Cond	$\%Test_1$	$\%Test_2$
WM	296.5	9	9	66.33	66.19
Ravi	44.77	5	5	86.21	86.21
2SLAVE	3.87	5.47	15.37	88.93	89.77
Tsakonas	18.13	1.27	1.38	62.2	60.37
C4.5	25	4.46	5.08	94.43	-
FRBCS_PG	7.97	1	2.12	91.97	92.8

Lung Cancer

Method	#R	#Var	#Cond	$\%Test_1$	$\%Test_2$
WM	28.8	56	56	0	0
Ravi	26.97	7	7	4.44	4.44
2SLAVE	3.73	26.43	59.22	77.97	74.87
Tsakonas	16.3	1.82	1.87	54.44	48.89
C4.5	6.2	2.74	2.82	78.34	-
FRBCS_PG	3.23	1	2.6	82.47	80.5

Thyroid

Method	#R	#Var	#Cond	%Test$_1$	%Test$_2$
WM	1078	21	21	91.42	91.44
Ravi	35.87	6	6	77.19	77.19
2SLAVE	3.46	9.24	22.93	92.5	92.5
Tsakonas	17.47	1.53	1.63	80.38	57.18
C4.5	24.5	3.69	3.88	99.61	-
FRBCS_PG	5.36	1	2.21	92.13	92.13

Analysing the results, we can point out the following considerations:

- Our method learns rule sets with a lower number of variables and labels per rule, for all the considered problems. It also learns rule bases with a small number of rules. Therefore the resulting FRBCSs have a high interpretability level.
- Analysing the performance of our approach, it presents a good performance in test for all the problems, obtaining the best results for some of them.

5 Conclusions

In this work, we have proposed a genetic-programming-based method to obtain FRBCSs with a high interpretability. Since the GP individuals are represented by variable-length trees, they can naturally allow for the absence of any input feature, getting rules with fewer antecedent conditions. On the other hand, the use of token competition mechanism to increase the diversity into the population, makes the rules compete among themselves giving out a smaller number of rules with a high-generalization capability.

The effectiveness of the method has been demonstrated over several classification problems and the results are promising. Therefore, we consider this approach to be an interesting alternative for the learning of interpretable FRBCSs for high-dimensionality problems.

As future work we will incorporate a proper multiobjective approach within the leaning process.

Acknowledgments

This work has been supported by the projects TIC-2002-04036-C05-01 and TIC-2002-04036-C05-04, and the networks TIN2004-20061-E and TIN2004-21343-E, by the Spanish Ministry of Science and Technology and FEDER funds.

References

[1] A. Bastian. "Identifying fuzzy models utilizing genetic programming". *Fuzzy Sets and Systems* 113 (3) 1, pp. 333-350, 2000.

[2] J. Casillas, O. Cordón, M.J. Del Jesús, and F. Herrera, "Genetic feature selection in a fuzzy rule-based classification system learning process for high-dimensional problems". *Information Sciences* 136 (1-4), pp. 135-157, 2001.

[3] J. Casillas, O. Cordón, F.Herrera, L.Magdalena (Eds.). *Interpretability Issues in Fuzzy Modeling*. Springer-Verlag, 2003. Series Studies in Fuzziness and Soft Computing, Vol. 128.

[4] D. Chakraborty, and N.R. Pal, "A neuro-fuzzy scheme for simultaneous feature selection and fuzzy rule-based classification". IEEE Transactions on Neural Networks 15 (1), pp. 110-123, 2004.

[5] Z. Chi, J. Wu, and H. Yan, "Handwritten numeral recognition using self-organizing maps and fuzzy rules". *Pattern Recognition* 28 (1), pp. 59-66, 1995.

[6] O. Cordón, M.J. del Jesús, and F. Herrera. "A Proposal on Reasoning Methods in Fuzzy Rule-Based Classification Systems". *International Journal of Approximate Reasoning* 20, pp. 21-45, 1999.

[7] O. Cordón, M.J. del Jesús, F. Herrera and M. Lozano. "MOGUL: A Methodology to Obtain Genetic fuzzy rule-based systems Under the iterative rule Learning approach". *International Journal of Intelligent Systems* 14 (11), pp. 1123-1153, 1999.

[8] O. Cordón, F. Herrera, F. Hoffmann, L. Magdalena. Genetic Fuzzy Systems. *Evolutionary tuning and learning of fuzzy knowledge bases*. World Scientific, 2001.

[9] S. García, F. González, and L. Sánchez. "Evolving fuzzy based classifiers with GA-P: A grammatical approach". Genetic Programming: *Second European Workshop, EuroGP'99*. Lecture Notes in Computer Science 1598, pp 203-210, 1999.

[10] A. Geyer-Schulz. *Fuzzy rule-based expert systems and genetic machine learning*. Heidelberg: Physica-Verlag, 1995.

[11] A. Giordana, F. Neri. "Search-intensive concept induction". *Evolutionary Computation* 3 (4), pp. 375-416, 1995.

[12] A. González, and R. Pérez, "Selection of relevant features in a fuzzy genetic learning algorithm". *IEEE Transactions on Systems, Man and Cybernetics – Part B* 31 (3), pp. 417-425, 2001.

[13] A. González, and R. Pérez, "SLAVE: A genetic learning system based on an iterative approach". *IEEE Transactions on Fuzzy Systems* 27, pp. 176-191, 1999.

[14] F. Hoffmann, and O. Nelles. "Genetic programming for model selection of TSK-fuzzy systems". *Information Sciences* 136 (1-4), pp. 7-28, 2001.

[15] H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, "Selecting fuzzy if-then rules for classification problems using genetic algorithms". IEEE Trans. Fuzzy Systems 3 (3), pp. 260-270, 1995.

[16] T. Kovacs, *Strength or Accuracy: Credit Assignment in Learning Classifier Systems*. Springer-Verlag, 2004.

[17] J.R. Koza, *Genetic programming – on the programming of computers by means of natural selection*. Cambridge MA, USA: The MIT Press. 1992.

[18] A. Krone, P. Krause, and T. Slawinski, "A new rule reduction method for finding interpretable and small rule bases in high dimensional search spaces". *Proceedings of the Ninth IEEE International Conference on Fuzzy Systems* vol. 2, pp. 694-699, May 2000.

[19] J.R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[20] V. Ravi, P.J. Reddy, and H.J. Zimmermann, "Pattern classification with principal component analysis and fuzzy rule bases". *European Journal of Operational Research* 126 (3), pp. 526-533, 2000.

[21] L. Sánchez, I. Couso, and J.A. Corrales. "Combining GP operators with SA search to evolve fuzzy rule based classifiers" *Information Sciences* 136 (1-4), pp. 175-191, 2001.

[22] A. Tsakonas, G. Dounias, J. Jantzen, H. Axer, B. Bjerregaard and D.G. von Keyserlingk. "Evolving rule-based systems in two medical domains using genetic programming". *Artificial Intelligence in Medicine* 32 (3), pp. 195-216, 2004.

[23] L.X. Wang, and J.M. Mendel, "Generating fuzzy rules by learning from examples". *IEEE Transactions on Systems, Man, and Cybernetics* 22 (6), pp. 1414-1427, 1992.

[24] M.L. Wong, and K.S. Leung, *Data Mining using grammar based genetic programming and applications*. Kluwer Academics Publishers, 2000.