Mathware & Soft Computing 7 (2000), 185-197

A Methodology for Constructing Fuzzy Rule-Based Classification Systems

J.M. Fernández Garrido and I. Requena Ramos
Depto. de Ciencias de la Computación e I.A. E.T.S.I. Informática.
Universidad de Granada. 18071 – Granada (Spain)
garrido@condor.ugr.es, requena@decsai.ugr.es

Abstract

In this paper, a methodology to obtain a set of fuzzy rules for classification systems is presented. The system is represented in a layered fuzzy network, in which the links from input to hidden nodes represents the antecedents of the rules, and the consequents are represented by links from hidden to output nodes. Specific genetic algorithms are used in two phases to extract the rules. In the first phase an initial version of the rules is extracted, and in second one, the labels are refined. The procedure is illustrated by applying it to two real-world classification problems.

Keywords: Classification Systems, Systems based on Fuzzy Rules, Genetic Algorithms.

1 Introduction

Two kinds of classification system may be distinguished: those designed to work autonomously and those whose main objective is to be used as tool to assist the user in a decision making process. In the first case, the main ---and possibly the only--- issue of importance is the performance of the system, namely, the hit rate in the classification task. In the second, in addition to a good success percentage, other properties such as comprehensibility, interpretability, robustness, etc. are required for the system to be accepted.

In this paper, a method of construction of easily interpretable classification systems is presented. It is based on the application of two fundamental tools: a fuzzy rule-based system (FRBS) (Man75, Pal92, Zad75) which provide easily interpretable models, and genetic algorithms (Gold89, Her95, Cor97a) as methods for the search of solutions. A descriptive approach is pursued in the construction process, so the labels for each linguistic variable are the same across all the rules.

Feldman [Feld93] proposed a FRBS construction model method which he applied to control problems. In his method the number of rules of the FRBS and the definition of the linguistic labels are preset . The FTBS is represented in a fuzzy network (see figure 1), and a classic binary genetic algorithm is used to obtain the Rule Set.

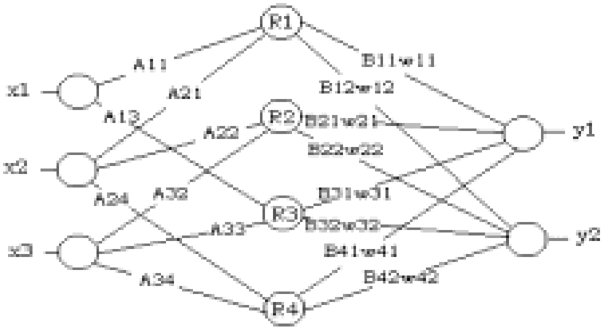


Figure 1: Structures of the fuzzy network

In the Figure 1, unit Ri represents the whole antecedent of a fuzzy rule of the type

IF x_1 is A_1 and x_2 is A_2 and .... and x_n is A_n THEN y_1 is B_1 and y_2 is B_2 and .... and y_m is B_m,

where A_{ij} and B_{jk} are linguistic labels and w_{ij} are real-valued weights associated to each term of the consequent.

In this work, taking as a starting point this method by Feldman, a new method to construct FRBS applied to classification problems is developed. To achieve it, an output y_k is assigned to each class. When sample inputs are applied to the classification system, it will conclude that the sample belongs to the class corresponding to the output with the highest value. In case no rule is fired, the sample will be assigned to class corresponding to output y_1, applying a default reasoning scheme.

To search the rule space a real-coded genetic algorithm will be employed. Special genetic operators will be deployed which can modify the initial number of rules (set of rules that define the system), so that the effective number is determined by the search process. The application of a mutation operator yielding a decreasing number of changes as the algorithm advances is also studied.

Two kinds of rules are considered:

a) Classic fuzzy rules, like:

IF x_1 is A_1 and x_2 is A_2 and .... and x_n is A_n *THEN* y_1 is B_1 and y_2 is B_2 and .... and y_m is B_m,

b) Rules with certainty degrees in the consequents:

IF x_1 is A_1 and x_2 is A_2 and .... and x_n is A_n *THEN* y_1 is $class_1$ degree r_1 and.... ..and y_m is $class_m$ with degree r_m , In both cases, A_{ij} and B_{jk} are linguistic labels, and a weight w_{ij} is assigned to each term of the consequent, which gives the relative importance of the term. The difference between a) and b) is that, in a), the output is a fuzzy set (label) and so must be defuzzified in order to obtain the class (y_i), but in b), the output is directly a real number.

The construction of the model is carried out in two phases:

Phase 1. Once the linguistic label definitions have been established (usually by an expert), a genetic algorithm will search for a base of rules that obtains a good percentage of successes. This phase can also be used as feature selection process, because the methodology consider the possibility of eliminating variables which do not participate in the rules.

Phase 2. Starting from the bases of rules obtained in the stage above, and using another genetic algorithm, the definition of the linguistic rules will be refined but keeping the global semantic across the rules in the base.

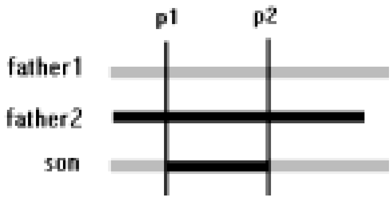
Now, we proceed to describe the method in detail.

2. Phase 1. Construction of the rule base

The definition of the linguistic labels is fixed a priori, being C_i the number of labels for the variable i. The search of the base of rules will be carried out with a genetic algorithm, in which each chromosome codes a base of rules, and the evaluation function is the success percentage in the classification of a training set.

Each rule will be coded like a t-uple R_1: (A_{11}, ..,A_{1r}; B_{11}, w_{11}, ..., B_{1p}, w_{1p}), in which the antecedent A_{ij} will take values in the set {0,.., C_i} (the value 0 indicates that the variable doesn't participate in the rule), the consequent B_{jk} in the set {1,.., C_j}, and the weights w_{jk} in the interval [0,1]. In the case of rules with degrees of certainty in the consequent, the degrees of certainty r_i will take values in the interval [0,1]. The initial population will be generated randomly.

The mutation operator will act on a rule from the chromosome to which is applied, modifying the values of some variables in the rule according to a uniformly defined random variable: If the variable is an input or output, the label is changed randomly by another label . The degrees of certainty in the consequent, and the weights, will be modified respectively, according to $w_{ij}(t+1) = w_{ij}(t) + u\ w_{ij}(t)$; $r_{ij}(t+1) = r_{ij}(t) + u\ r_{ij}(t)$ (where $u \sim U([-0.1, 0.1])$):



If the value is out of the interval [0,1], a rebound effect is applied:

If $w_{ij}(t+1) > 1$, $w_{ij}(t+1) = 1-(w_{ij}(t+1)-1)$; if $w_{ij}(t+1) < 0$, $w_{ij}(t+1) = -w_{ij}(t+1)$

The number of values to change in each mutation, can be fixed a priori or can be variable, decreasing as the number of iterations increases.

The crossing operator will be classic, with two crossing points, $p_1 < p_2$ with exchange of complete rules, which are selected randomly in the set {0, 1, ... N}, where N is the fixed rule number [for the case of chromosomes with a variable rule number, N= min (length(parent1), length(parent2)), and length(parent1) is the rule number in parent1,]. The length of the new chromosome is the one of the first parent.

To modify the length of the chromosomes, the operators “add_rule”, that chooses a chromosome randomly and adds it a rule of the best chromosome (randomly), and “delete_rule”, that eliminates a rule of the chromosome chosen randomly according to an uniform distribution, are introduced.

The genetic algorithm will be an elitist model, always conserving the best chromosome, using as selection strategy a model of steady state according to which the chromosomes generated by crossing, add_rule and delete_rule replace to the worst in the population and the mutation transforms the chromosome on which it is applied. The size of the population is fixed. As stop criterion, a fixed number of generations is used.

EXPERIMENTAL RESULTS

To test the behaviour of this algorithm, it has been applied the problems of IRIS and PIMA. In each problem, five randomly partitions have been generated (a training set with 2/3 of the samples, and the rest for test). Ten indepen-dent executions of the algorithm have been run on each partition, carrying out a total of 50 executions for experiment.

The experiments carried out, for each model (linguistic label and degrees of certainty) are:

Rules Number	Change Number in mutation	
	Fixed	Decreasing
Fixed	Experiment1	Experiment2
Variable	Experiment3	Experiment4

The parameters we have used are:

- $C_i = C_j = 3$ (3 labels for variable) in the Iris and $C_i = C_j = 5$ in the Pima. Number of iterations = 500. Population size = 100.
- The probabilities: For crossing is 0.5 ; for mutation is 0.3, and for apply the add_rule or delete_rule is 0.2.
- For the prefixed rule number we have used 5, 7 and 8 with IRIS, and 7, 8 and 10 with PIMA. For variable rule number, we have fixed a minimum and maximum of 2 – 10 in IRIS case, and 4 – 15 in PIMA case.
- The number of changes in mutation was fixed to 2 in IRIS, and in the decreasing case, 2 for the 250 initial iterations and 1 for the following 250 iterations. In PIMA, 3 prefixed, and in the decreasing case, 4 in 1-100 iterations, 3 in 1001-200 it., 2 in 201-300 it. and 1 in the last 200 iterations.

The obtained results can be seen in the following tables and figures, where in the bars, the media of success percentage on the 50 runs carried out is represented for each experiment, both to training and test set. In the legend of the figures appears the mean number of used rules and the results of the best execution for each experiment (learning, test)

A) IRIS PROBLEM

LING. LABEL	Media			Best		
Experiments	**% Training Set**	**% Test Set**	**N.Rules**	**% Training Set**	**% Test Set**	**N.Rules**
Exper.1	99.1	94.5	7	99	98	7
Exper.2	99.3	94.5	7	98	100	7
Exper.3	98.8	93.0	4.2	100	96	4
Exper.4	98.6	93.2	3.9	100	96	5
CERT. DEGR.	**Media**			**Best**		
Experi.	**% Training Set**	**% Test Set**	**N.Rules**	**% Training Set**	**% Test Set**	**N.Rules**
Exper.1	99.3	94.7	7	99	100	7
Exper.2	99.3	94.8	7	100	98	7
Exper.3	99.1	93.5	4.2	100	98	6
Exper.4	98.9	93.4	4.1	99	98	4

Table 1. Experimental results for Iris Problem

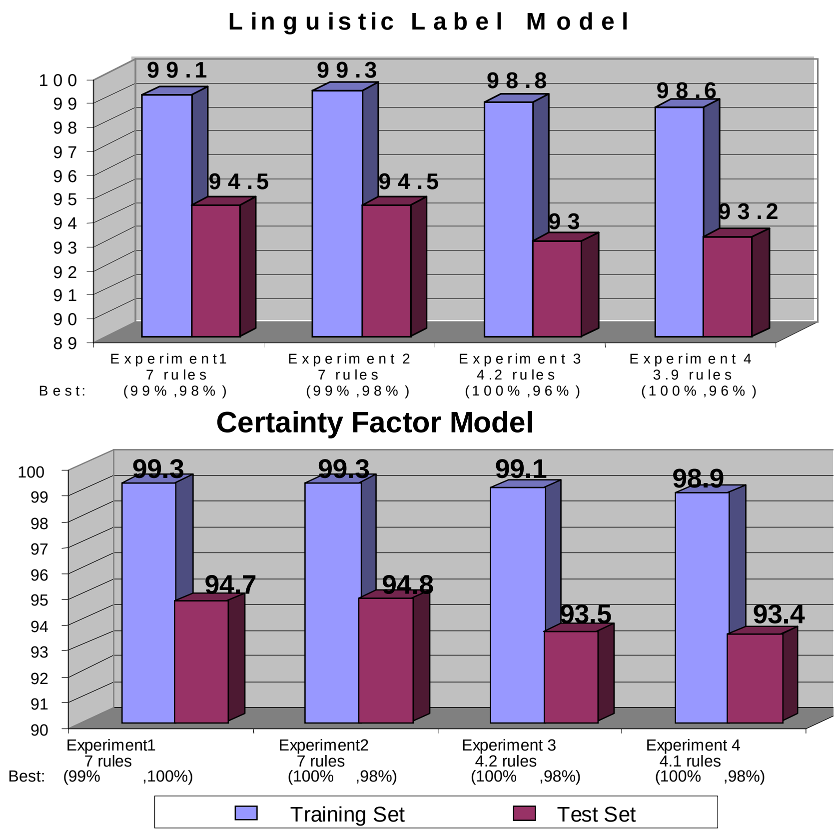


Figure 2.- Experimental results for Iris Problem

As can be observed (Table 1 and figure 2), success percentages are obtained above 99% on training set and around 94.5% on test set. Several executions, just with 1 error on training + test, have been obtained. (100% + 98% or 99% +100% of successes) . If the number of rules is not preset, the process begin with 8 rules. At the end of phase 1 (about 4 rules), the mean results of successes descend lightly, although it is no so when the initial number of rules is fixed to 4 rules.

Let us see as an example, a group of rules with linguistic labels in the consequent (after the labels, between parenthesis, the weights appear). In this group of rules that makes 2 errors in the classification of the total set of samples, three of the four entrance variables are only used (x_1 is not used). So, a characteristic selection have been made.

R_1: x_3 is high and x_4 is high ➔ y_1 is regular (0.589), y_2 is high (0.401) e y_3 is low (0.185)

R_2:x_3 is regular ➔ y_1 is low (0.833), y_2 is regular (0.201) e y_3 is high (0.884)

R_3:x_3 is high ➔ y_1 is regular (0.703), y_2 is high (0.180) e y_3 is low (0.185)

R_4:x_2 is regular and x_3 is regular ➔ y_1 is low (0.869), y_2 is regular (0.201) e y_3 is high (0.980)

B) PIMA PROBLEM

In the Pima Problem (Table 2 and Figure 3), results near to 80% of successes for the training and to 76% for the test are obtained. As better executions results are obtained above 80% so much envelope the set of training like on the test set. In the experiments 3 and 4 we can see that, leaving of 8 rules initials, they don't decrease in general the number of rules, at least in a significant way. The results stay basically

LING. LABEL	Media			Best		
Experiments	**% Training Set**	**% Test Set**	**N.Rules**	**% Training Set**	**% Test Set**	**N.Rules**
Exper.1	79.91	76.19	8	79.49	82.03	8
Exper.2	79.63	75.98	8	80.47	81.64	8
Exper.3	79.52	75.08	7.14	80.66	77.73	6
Exper.4	79.79	75.84	7.66	78.52	81.25	6
CERT. DEGR.	**Media**			**Best**		
Experi.	**% Training Set**	**% Test Set**	**N.Rules**	**% Training Set**	**% Test Set**	**N.Rules**
Exper.1	79.93	75.22	8	81.44	78.52	8
Exper.2	80.06	75.77	8	80.86	80.08	8
Exper.3	79.80	75.20	8	77.54	82.81	6
Exper.4	79.99	74.89	8.24	80.86	77.34	9

Table 2. Experimental results for Pima Problem

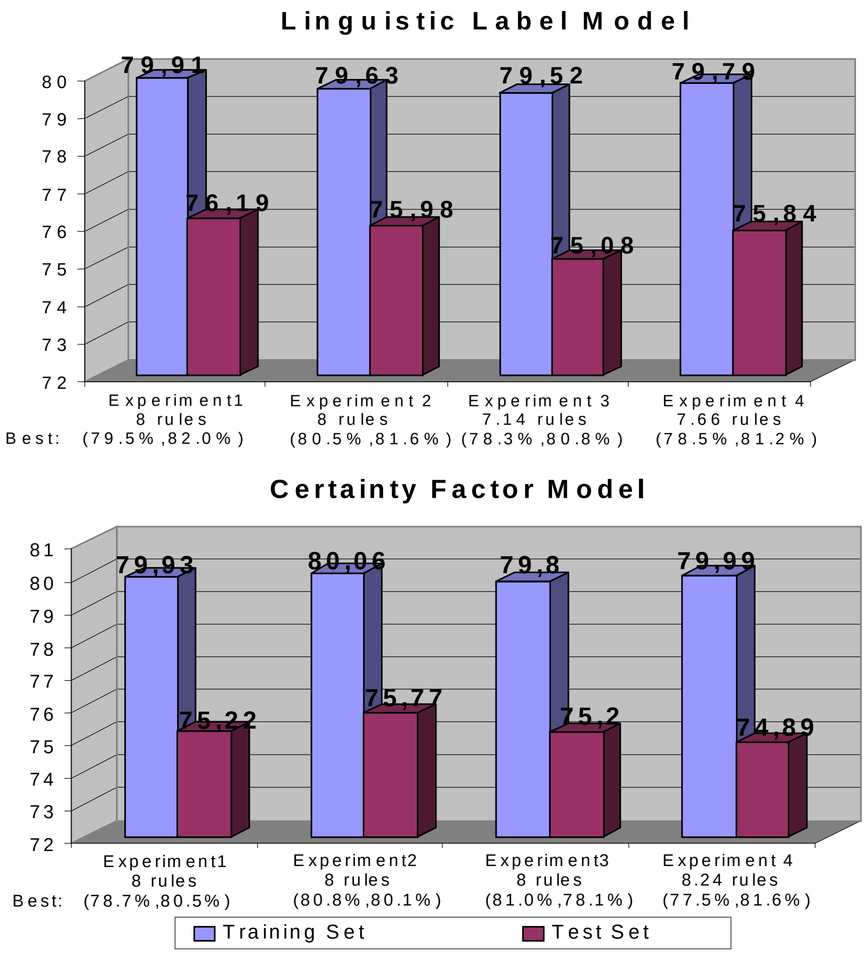


Figura 3.- Experimental results for Pima Problem

3. Phase 2. Optimization of the linguistic labels

The objective in this phase is to refine the linguistic labels of the variables in the problem, maintaining the interpretability of the rules. The used algorithm will also be genetic, where each chromosome will code the definition of the linguistic labels that use the bases of rules. The inputs will be the M better sets of rules obtained with the previous algorithm, and it will use as evaluation function the expression

$$f_2(C_i) = \max_{j=0}^{M-1} f(R_j, C_i, X)$$

with f(R,C,X) the success percentage in the classification of the set of samples X with the set of rules R and the definition of the linguistic labels C.

Each chromosome codes all linguistic labels that participate in the combined M of rules, and as we use triangular fuzzy numbers, each label is represented by 3 real numbers (a, b, c) that should complete certain restrictions:

a) Limits inferior (a) ≤ (Fashion (b) ≤ Limits superior (c)

b) a, b, c belongs to the domain of values of the variable.

For the initial population, the first chromosome will be built with the labels obtained by the previous algorithm (phase 1), and the other chromosomes will be generated starting from this, modifying the parameters p of the labels according to the expression: $p\ (t+1) = p(t) + u \quad anch$ with u ~ U([-0.2, 0.2]) and anch = lim_sup - lim_inf

The mutation operator will carry out several changes, and each one will consist on selecting, according to an uniform random variable, a parameter p of a fuzzy set (label) of a variable, and modify it, verifying the restrictions described before, and using the previous expression with u ~ U([-0.1, 0.1]).

The crossing operator, classic for 2 points, it exchanges complete linguistic variables. The genetic algorithm will be an elitist model, always conserving the best chromosome, using as selection strategy a model of stable state ("steady state") as before. The stop criteria is a fixed number of generations and also the population size is fixed.

EXPERIMENTAL RESULTS

The results obtained when applying it on the executions of the previous algorithm, can be seen in the following figures, where the improvement got with the algorithm of optimisation is also indicated.

A) IRIS PROBLEM

LING. LABEL	Media			Best		
Experi.	**% Training Set**	**% Test Set**	**N.Rules**	**% Training Set**	**% Test Set**	**N.Rules**
Exper.1	99.8	96.9	7	100	100	7
Exper.2	99.9	96.5	7	99	100	7
Exper.3	99.6	96.0	4.2	100	98	4
Exper.4	99.3	95.5	3.9	100	100	6
CERT. DEGR.	**Media**			**Best**		
Experi.	**% Training Set**	**% Test Set**	**N.Rules**	**% Training Set**	**% Test Set**	**N.Rules**
Exper.1	99.8	97.0	7	100	100	7
Exper.2	99.8	96.6	7	100	98	7
Exper.3	99.7	96.1	4.2	100	100	5
Exper.4	99.6	95.9	4.1	100	98	3

Table 3. Results for Iris Problem, after Optimisation Phase

After running the optimisation algorithm, an improvement of the previous results takes place (see table 3 and Figure 4), getting near 100% successes on the training set and around 97% on the test set. As better result, it has been obtained in several executions 100% success, both on training and test sets.

With the definition of the linguistic labels made a priori, it made 3 errors in the classification of the total group of samples (training + test). The same group of rules, with a definition of the linguistic labels obtained with the algorithm of optimisation doesn't make any error in the classification of the total group of samples.

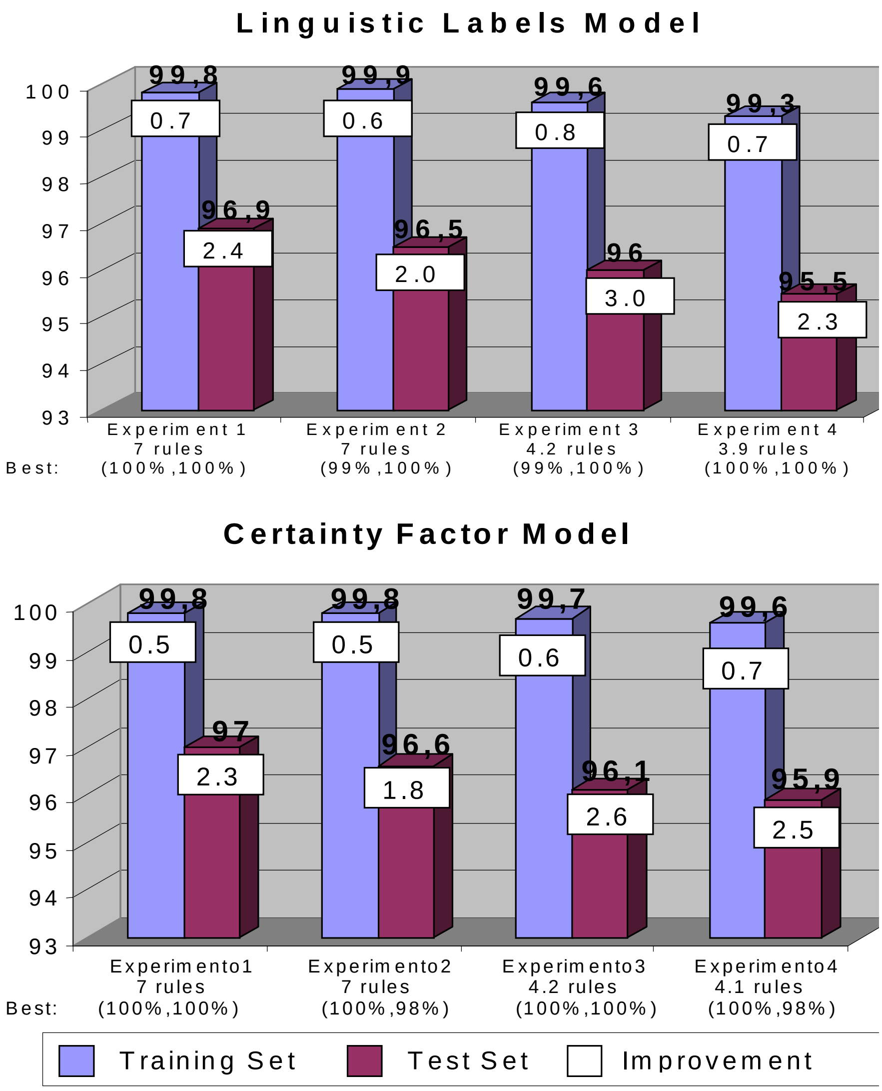


Figure 4.- Results for Iris Problem, after Optimisation Phase

In the following table 4, the definitions of the linguistic labels (triangular fuzzy numbers) of some variables, can be seen before and after the optimisation. If a label is not used in the rules, then it is no optimised (it is say “no used”).

Variable / labels		**a priori**	**after optimisation**
X_1	Low	4.000000,5.000000,6.000000	No used
	Regular	5.000000,6.000000,7.000000	5.075925,5.684425,7.415382
	High	6.000000,7.000000,8.000000	6.072806,7.268104,7.795695
X_2	Low	2.000000,2.750000,3.500000	2.453243,2.653426,3.326715
	Regular	2.750000,3.500000,4.250000	2.554554,4.070342,4.270333
	High	3.500000,4.250000,5.000000	No used
X_4	Low	0.000000,0.750000,1.500000	0.197051,0.740059,1.716486
	Regular	0.750000,1.500000,2.250000	0.817952,1.133237,2.096435
	High	1.500000,2.250000,3.000000	1.606774,2.432081,2.881521
Y_2	Low	0.000000,1.000000,2.000000	0.093782,1.104570,1.866771
	Regular	1.000000,2.000000,3.000000	1.396730,1.671366,2.790037
	High	2.000000,3.000000,4.000000	2.486762,2.584316,3.798983
Y_3	Low	0.000000,1.000000,2.000000	0.006532,0.683632,2.150017
	Regular	1.000000,2.000000,3.000000	No used
	High	2.000000,3.000000,4.000000	1.914575,2.832996,3.713253

Table 4. Labels definition of some variables, before and after of optimisation phase

As an example of the result of the optimisation one can observe the following group of rules:

R_1: *x_3 is regular and x_4 is regular → y_1 is low(0.709), y_2 is high (0.407), y_3 is low(0.337)*

R_2: *x_3 is high → y_1 is low(0.009), y_2 is regular (0.350), y_3 is high (0.218)*

R_3: *x_3 is high and x_4 is regular → y_1 is regular(0.039), y_2 is regular (0.351), y_3 is high (0.218)*

R_4: *x_2 is low and x_4 is high → y_1 is low(0.663), y_2 is low (0.222), y_3 is high (0.967)*

R_5: *x_1 is high and x_3 is high and x_4 is low → y_1 is regular(0.721), y_2 is low (0.051), y_3 is high (0.342)*

R_6: *x_1 is regular and x_2 is regular and x_3 is high and x_4 is regular → y_1 is low (0.975), y_2 is low(0.39), y_3 is low (0.622)*

B) PIMA PROBLEM

The values of the parameters used in the phase 2, are basically the same ones that in the phase 1.

After the algorithm of optimisation, in the PIMA an improvement also takes place, getting results above 82% on the set of training and around 76.5% on the test set. As better result, in all the experiments they have been obtained so much in several executions percentages of success superiors to 80% in the set of training like on the test set, as can be seen in the following table 5 and figure 5.

LING. LABEL	**Media**			**Best**		
Experi.	**% Training Set**	**% Test Set**	**N.Rules**	**% Training Set**	**% Test Set**	**N.Rules**
Exper.1	82.40	76.70	8	81.64	82.42	8
Exper.2	82.16	76.52	8	82.23	81.64	8
Exper.3	82.37	75.71	7.14	80.08	82.42	7
Exper.4	82.25	76.47	7.66	81.64	82.03	5

CERT. DEGR.	Media			Best		
Experi.	**% Training Set**	**% Test Set**	**N.Rules**	**% Training Set**	**% Test Set**	**N.Rules**
Exper.1	82.27	76.12	8	83.20	80.08	8
Exper.2	82.48	76.54	8	82.617	82.81	8
Exper.3	82.51	75.72	8	80.47	82.81	6
Exper.4	82.60	75.66	8.24	80.08	82.03	7

Table 5. Results for Pima Problem, after Optimisation Phase

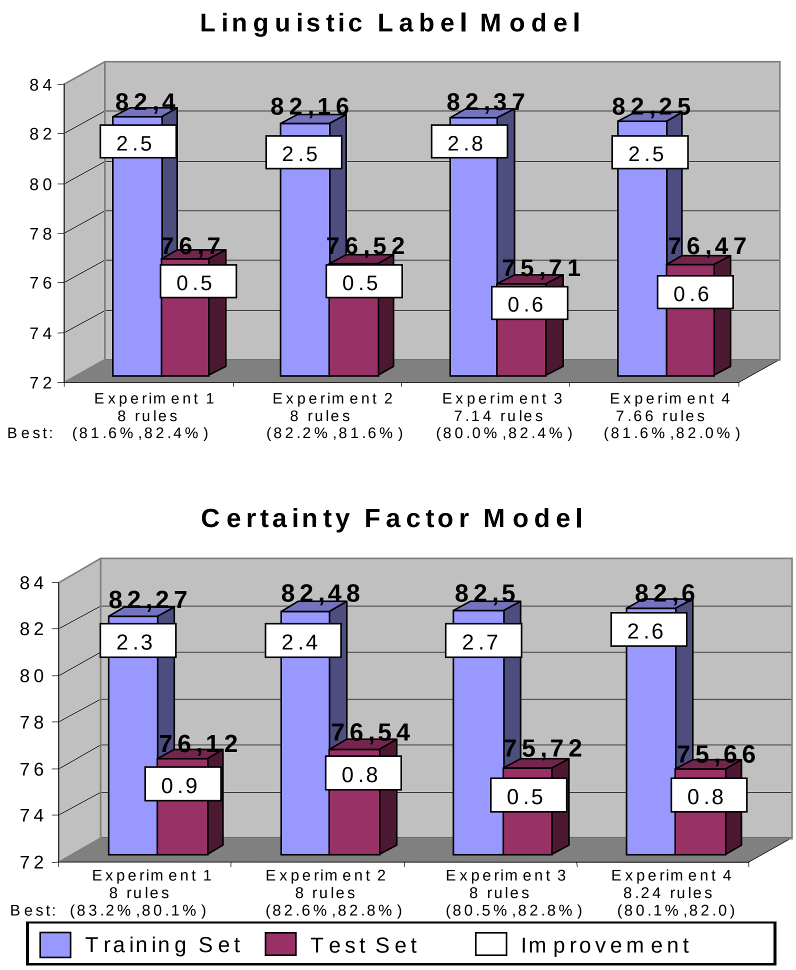


Figure 5.- Results for Iris Problem, after Optimisation Phase

In order to show that the proposed methodology obtains good results, in the following table 6, are showed the results of other methodologies, that have been published.

IRIS					PIMA			
Methodology	**Training**	**Test**	**N. rules**		**Methodology**	**Training**	**Test**	**N. rules**
[Ben98]	100%	96.6%	3		**[Ben98]**	83.2%	78.2%	13
CN2 [Clar86]	98.92%	94.16%	6.4		**CN2 [Clar86]**	85.4%	74.5%	38
[Cor97b]	99.63%	95.63%	38.2		**[Cor97b]**	82.56%	75.3%	103
[Per97]	97.51%	96.21%	4.0		**[Per97]**	79.37%	80.9%	23

Table 6. Results published, obtained with some methodologies

4. Conclusions

In this work, a methodology for the construction of FRBS for classification has been presented, represented as a fuzzy network. The method allows to develop FRBS, using two types of rules: rules with linguistic labels and rules with degrees of certainty in the consequent.

A method has been implemented for the obtaining of a set of rules, once fixed the definition of the linguistic labels that participate in the problem that obtains good initial results in the problems Iris and Pima. Specific genetic operators have been used for the algorithm, i.e., the one used to modify the number of the initial rules, or the one that considers for mutation a variable number of modifications as the execution of the algorithm advances.

A method for refinement the linguistic labels has been implemented, based on the evaluation of several groups of rules, with the objective of avoiding a possible overlearning of the training set and, simultaneously, giving to the resulting label a meaning of a more global search. When applying this method to the IRIS and PIMA problems we have obtained very good results that are comparable to the best published ones.

With each example, different experiments have been carried out, using a fixed or variable number of rules, or using linguistic variables instead of degrees of certainty in the consequents of the rules. As can be observed in the experiences carried out, not all the problems with the same characteristics have the best results. Although in a classification problem it seems reasonable not to use linguistics labels in the consequents of the rules the obtained results indicate that in the problem of the Pima, this type of rules behave lightly better than the certainty factor model, mainly in the generalisation (test group).

A line for future works will be to apply the model with degrees of certainty for obtaining fuzzy classifications, to improve the used genetic algorithms, to uses similarity measures in the selection of the combined sets M of the rules in the phase of optimization, so that these sets are “sufficiently different”, and finally to consider the first algorithm as a characteristic selection method and then to apply methods of direct construction of the linguistic labels in the second phase.

5. References

[Ben98] Benítez Sánchez, J.M. "Extraction of fuzzy rules with and of artificial networks neural." Ph. D. Dissertation (in spanish) . 1998.. Universidad Granada (Spain)

[Clar86] Clark, P.; Niblett, T. *"Learning If Then Rules in noisy domains"* TIRM 86-019 The Turing Institute. Glasgow. 1986

[Cor97a] Cordón, O.; Herrera, F.; Lozano, M. *"On the combination of Fuzzy Logic and Evolutionary Computation: A Short Review and Bibliography"*. En Pedrycz (ed.) "Fuzzy Evolutionary Computation". Kluwer Academic Pub. 1997

[Cor97b] Cordón, O; del Jesus, M.J.; Herrera, F.; López. E. *"Selecting fuzzy rule-based classification systems with specific reasoning methods using genetic algorithms"*. 7th IFSA World Congress. 1997. 424-429

[Feld93] Feldman, D.S. *"Fuzzy Network Synthesis with Genetic Algorithms"*. Fifth International Conference on Genetics Algorithms. Urbana (Illinois). 1993. 312-317

[Gold89] Goldberg, D.E. *"Genetic Algorithms in Search,Optimisation and Machine Learning"*. Addi.-Wesley, N.Y. 89

[Her95] Herrera, F.; Lozano, M.; Verdegay, J.L. *"Algoritmos genéticos: Fundamentos, Extensiones y Aplicaciones"*. ARBOR, Septiembre 1995.

[Mam75] Mamdani, E.H.; Assilian, S. *"An experiment in linguistic synthesis with a fuzzy logic controller"* Int. Journal of Man-Machine Studies, 7(1). 1-13. 1975

[Pal92] Pal, S.K.; Mandal, D.P. *"Linguistic Recognition System Based on Aproximate Reasoning"* Information Sciences 61 (1992) 135-161

[Per97] Pérez Rodríguez, F.G.R. *"Aprendizaje de Reglas Difusas Usando Algoritmos Genéticos"* Tesis doctoral. Departamento de Ciencias de la Computación e Inteligencia Artificial. Universidad de Granada. 1997

[Zad75] Zadeh, L.A. *"The concept of a linguistic variable and its applications to aproximate reasoning"*. *Part I* Information Sciencies 8. Pags 199-249. *Part II* Information Sciencies 8. 301-357. *Part III* Information Sciencies 9. 43-80