



# **Evolving TSK fuzzy rules for classification tasks by Genetic Algorithms**

Rogério P. Espíndola & Nelson F. F. Ebecken  
*COPPE/Federal University of Rio de Janeiro.*

## **Abstract**

This paper presents a method for classification based on a genetic algorithm to evolve a Takagi-Sugeno-Kang fuzzy rule bases. It is shown how a fuzzy rule base is generated from a numerical database and how its best subset is found by the genetic algorithm. The method not only produces rule base subsets with very good performance but also reveals itself as a feature selector. Analyses are realized on the results obtained from several classification problems approached by the method and by a decision tree based algorithm.

## **1 Introduction**

Classification is one of the most important tasks of Machine Learning. Basically, it aims to associate pre-defined classes to new elements. Several Machine Learning paradigms have been applied to this problem like artificial neural networks, decision trees and production rules systems. Independently of the approach employed to construct a classifier system, some aspects must be analysed such as comprehension facility - avoiding the occurrence of application mistakes and making it reliable - and the relation between accuracy and processing time - in some cases, it may be interesting to construct a classifier slightly less accurate and much faster.

Employing Evolutionary Computation paradigms on classification tasks has been done considering these questions. For instance, these paradigms can be applied as to discover optimum relationships between the features and classes of a problem as to be combined with other Machine Learning techniques, generating hybrid classifier systems. The most disseminated area of Evolutionary



Computation is Genetic Algorithms due to the fact that it is a class of general purpose techniques able to approach efficiently very complex problems, success not obtained by some other methods - for the difficulty or impossibility of application.

Several genetic algorithms have been developed to perform feature extraction and selection from databases, which are related tasks to classification. Their aims are the identification and selection of real important features on the classification of new elements and the discarding of irrelevant or redundant ones [1,2,3,4]. The exclusion of these features frequently improves the results' quality because they may hide useful information included on the other features. The features subset used by a classifier system affects its precision, cost, processing time and the volume of examples necessary to learning, for instances [5].

It is seldom possible to construct a classifier system which is totally accurate. One of the causes to this fact is the insufficiency of a problem's representative data. Since the uncertainty and ambiguity involved on classification tasks are inevitable, the use of fuzzy logic in the classifier system makes it suitable to deal with these problems. This application induces better knowledge representation of the problem studied, as well as more simplicity and flexibility. The synergy between Genetic Algorithms and Fuzzy Systems occurs in a frequently successful way. Hybrid models that unite the characteristics of these two paradigms are some of the most robust systems of Machine Learning.

For instance, the fuzzy sets theory concepts have been applied in genetic algorithms to model new ways of encoding individuals and new genetic operators [6], to control the parameters during evolution and to study their relationships [7]. On the other hand, many genetic algorithms have been developed to improve the performance of fuzzy rule-based systems on classification and control tasks [8,9,10]. Furthermore, several works on project, learning and tuning of parameters of rule bases by genetic algorithms obtained promising results [11,12].

[13] presents a method for selecting a subset from a fuzzy rule base by a genetic algorithm to classification. The rule base is generated automatically and its rules are Mandani's model ones. After the rule base generation, a genetic algorithm is applied to find the shorter and most accurate subset of rules. A drawback of this kind of rules is that the dimension of the data base affects the rules' comprehension and the processing time of the method, reducing its applicability.

To overcome this obstacle, this work employed the same genetic algorithm to optimize decomposed zero-order Takagi-Sugeno-Kang (TSK) rule bases [14]. This kind of rule uses only one feature on the antecedent part and conclude about the class of the new pattern. The gains in processing time, simplicity of implementation and comprehension are notable.

Some databases from UCI Machine Learning Repository were studied by this method and the results compared to others obtained from a decision tree tool applied to the same problems.

In the next section, the automatic generation of rule base is shown. In section

3, the features of the genetic algorithm are exhibited. In section 4, the results of the experiments are analysed. In the last section, final comments and future developments directions are exposed.

## 2 Rule base generation

### 2.1 Rule base generation

The process of rule generation was presented in [14] which applies the decomposition scheme proposed by Kosko [15] to zero-order TSK fuzzy rules.

In this work, each attribute space was normalised and divided on five partitions defined by triangular membership functions associated to the linguistic labels *small*, *medium small*, *medium*, *medium large* and *large*. They are:

$$\begin{aligned}\mu_{\text{small}}(x) &= \max\{0, 1 - 4 \cdot |x|\} \\ \mu_{\text{medium\_small}}(x) &= \max\{0, 1 - 4 \cdot |x - 0,25|\} \\ \mu_{\text{medium}}(x) &= \max\{0, 1 - 4 \cdot |x - 0,5|\} \\ \mu_{\text{medium\_large}}(x) &= \max\{0, 1 - 4 \cdot |x - 0,75|\} \\ \mu_{\text{large}}(x) &= \max\{0, 1 - 4 \cdot |x - 1|\}\end{aligned}$$

The rules are constructed in such a way that they inform the influence of each linguistic label of an attribute in the occurrence of a determined class. This influence is a frequency estimation of a class given a record or a pattern. Given training patterns  $x^m = (x_1^m, \dots, x_n^m)$  with classes  $y^m = (y_1^m, \dots, y_K^m)$ , the influence value  $\phi_{kj}^i$  is obtained from the following expression:

$$\phi_{kj}^i = \frac{\sum_{m=1}^M \mu_{X_{ij}}(x_i^m) \cdot y_k^m}{\sum_{m=1}^M y_k^m}$$

in which  $k$  is the class influenced by the linguistic label  $j$  of the feature  $x_i$ .

Thus, the rules have the following structure:

Rule  $R_{k,j}^i$ : If  $x_i$  is  $X_{i,j}$  then class =  $k$  with influence  $\pi_{k,j}^i = \phi_{k,j}^i$

in which  $k = 1, \dots, K$  e  $j = 1, \dots, 5$ .

Considering a database with  $n$  attributes and  $K$  possible classes, a rule base generated in this way has  $5 \cdot K \cdot n$  elements.

## 2.2 Classification of new patterns

Given a pattern  $(x_1, \dots, x_n)$ , to determine its class requires the execution of the following steps.

1. for each feature  $i$ , combine the outputs of those rules related to the same class  $k$

$$\pi_k^i = \frac{\sum_{j=1}^5 \varphi_{kj}^i \cdot \mu_{X_{i,j}}(x_i)}{\sum_{j=1}^5 \mu_{X_{i,j}}(x_i)} = \sum_{j=1}^5 \varphi_{kj}^i \cdot \mu_{X_{i,j}}(x_i), \text{ in which } k = 1, \dots, K.$$

2. for each class, aggregate the previous combined outputs by the following function:

$$\pi_k = \min \left\{ \pi_k^i / i = 1, \dots, n \right\}$$

3. the class with the highest value  $\pi_k$  is associated to the pattern.

## 3 Application of Genetic Algorithms

### 3.1 Introduction

In many problems, to use all the rule base on classification of a new pattern can not be appropriate due to (among other reasons):

1. the rule base may be too large, causing an excessive delay to the pattern evaluation;
2. if the amount of patterns to be evaluated is too big, the processing time becomes too high;
3. the rule base may not produce the correct class due to the negative influence of rules with low confidences.

The aim of this genetic algorithm is to select a small subset of rules with the greatest accuracy.

### 3.2 Encoding and Fitness function

Each rules subset - candidate solution - was represented as a chromosome using

the binary alphabet  $\{0,1\}$ . Each rule is represented by a gene in the chromosome. If the rule is present on the subset, its correspondent gene receives the allele 1. Otherwise, it receives the allele 0.

The amount of rules of a candidate solution  $S$  acts as a penalty factor on the fitness function, since it is desirable small subsets. The ability of correctly classifying patterns from a database by  $S$  is a characteristic which is more important than the amount of rules it is formed by. Thus, that characteristic must has a distinct weight on the constitution of the fitness function defined by:

$$f(S) = W_{NCP} \cdot NCP(S) - W_S \cdot \sum_{R_k \in S} 1$$

in which  $NCP(S)$  is the amount of patterns correctly classified,  $W_{NCP}$  is the weight of this characteristic,  $W_S$  is the weight of the size of  $S$  and  $0 < W_S \ll W_{NCP}$ .

### 3.3 The genetic operators

It was applied the uniform recombination with probability of 0.5, the selection of the best chromosome pair without restitution to recombination and elitism. Considering the aim of reducing the amount of rules in each individual, two strategies of mutation were defined. The first one,  $P_m(1 \rightarrow 0)$ , with higher probability of occurrence, executes the change of the allele 1 to allele 0. The other one,  $P_m(0 \rightarrow 1)$ , changes the allele 0 to allele 1.

## 4 Results analysis

### 4.1 Cases studied

In order to assess the efficiency of the method, several databases with different characteristics were studied. Moreover, a decision tree software was applied to have its results compared. The table 1 below shows the databases, their dimensions and the amount of rules generated. Before working with the databases, some changes were made to allow the analysis of the method. Repeated records or records with incomplete information were eliminated and qualitative features were converted to discrete quantitative features.

### 4.2 Genetic algorithm settings

Population size: 10  
Weight  $W_{NCP}$ : 1000  
 $P_m(1 \rightarrow 0) = 0.01$

Stopping condition: 1000 generations  
Weight  $W_S$ : 1  
 $P_m(0 \rightarrow 1) = \text{variable}$



The utilization of different values to  $P_m(0 \rightarrow 1)$  was done in order to obtain better results than the original suggested value 0,0001. Although the other parameters could be changed, it was not done because it escapes from the purposes of this work.

As genetic algorithms are based on probabilistic operations, the results obtained from two executions with given parameters setting are not necessarily equal. Then, 20 executions of the genetic algorithm were performed to each setting and the medium performance was calculated.

Table 1. Summary of databases' characteristics.

| Database                         | Valid features | Valid records | Training examples | Test examples | Rules generated |
|----------------------------------|----------------|---------------|-------------------|---------------|-----------------|
| Iris plant                       | 4              | 150           | 150               | 150           | 60              |
| Wisconsin breast cancer          | 9              | 451           | 300               | 151           | 90              |
| Credit card application approval | 15             | 653           | 416               | 237           | 150             |
| Pima indian diabetes             | 8              | 788           | 510               | 258           | 80              |

### 4.3 Results analysis

For the classification of iris plant, the results obtained by the genetic algorithm are shown on table 2. The decision tree tool was executed with some settings of two parameters: the feature selection to ramification criteria and the significance level, a ramification stopping condition. Its results are shown on table 3. The letters R, A, F and G are related to amount of rules, accuracy (%), frequency and generation, respectively.

The best classifier found by the genetic algorithm used 3 features and it was formed by 6 rules from a total of 60. On the other hand, the rules constructed from the best decision tree used all features and they had no structure in the antecedent part, making the comprehension much more difficult.

Tables 4 and 5 show the results obtained from the genetic algorithm and the decision tree applied to the diagnosis of breast cancer. As it can be noticed, the results obtained by the genetic algorithm are quite better than those obtained by the decision trees. The best fuzzy rule-based classifier, formed by only 7 rules (from 90), used only 5 features. It was more accurate and shorter than the best decision tree.

For the credit card application approval database, tables 6 and 7 show the results obtained by the genetic algorithm and the decision tree. From 150 rules, the best classifier found by the genetic algorithm was compound by 9 rules and it used 7 features. It outshines the best decision tree which had better performance than only 7 (from 60) fuzzy rule-based classifiers. Another very good result was obtained by the genetic algorithm with  $P_m(0 \rightarrow 1) = 0.01$ . This fuzzy classifier had 90.30% of accuracy and it was formed by 5 rules which used only 4 features.



Table 2. Results obtained by the genetic algorithm applied to iris plant database.

| $P_m(0 \rightarrow 1)$ | Best classifier |       |   | Worst classifier |       |   | Medium Performance |       |        |
|------------------------|-----------------|-------|---|------------------|-------|---|--------------------|-------|--------|
|                        | R               | A     | F | R                | A     | F | R                  | A     | G      |
| 0.0001                 | 4               | 97.33 | 3 | 6                | 90.00 | 1 | 4                  | 94.97 | 202.30 |
| 0.005                  | 6               | 98    | 1 | 6                | 96.67 | 1 | 4.2                | 97.33 | 283.50 |
| 0.01                   | 6               | 98    | 1 | 4                | 97.33 | 1 | 4.1                | 97.37 | 267.00 |

Table 3. Decision trees generated from iris plant database

| Feature selection<br>to ramification<br>criteria | Significance level (%) |      |     |      |     |      |           |      |
|--------------------------------------------------|------------------------|------|-----|------|-----|------|-----------|------|
|                                                  | none                   |      | 0.1 |      | 0.5 |      | suggested |      |
|                                                  | R                      | A    | R   | A    | R   | A    | R         | A    |
| Chi-square                                       | 9                      | 98.0 | 6   | 97.3 | 7   | 97.3 | 6         | 97.3 |
| Entropy                                          | 9                      | 98.0 | 6   | 97.3 | 7   | 97.3 | 6         | 97.3 |

Table 4. Results obtained by the genetic algorithm applied to breast cancer database.

| $P_m(0 \rightarrow 1)$ | Best classifier |     |   | Worst classifier |       |   | Medium Performance |       |        |
|------------------------|-----------------|-----|---|------------------|-------|---|--------------------|-------|--------|
|                        | R               | A   | F | R                | A     | F | R                  | A     | G      |
| 0.0001                 | 13              | 100 | 1 | 10               | 96.69 | 1 | 10.8               | 98.28 | 259.30 |
| 0.005                  | 7               | 100 | 1 | 10               | 98.68 | 1 | 10.1               | 99.40 | 545.40 |
| 0.01                   | 9               | 100 | 1 | 13               | 98.68 | 1 | 10.8               | 99.30 | 584.75 |

Table 5. Decision trees generated from breast cancer database

| Feature selection<br>to ramification<br>criteria | Significance level (%) |      |     |      |     |      |           |      |
|--------------------------------------------------|------------------------|------|-----|------|-----|------|-----------|------|
|                                                  | none                   |      | 0.1 |      | 0.5 |      | suggested |      |
|                                                  | R                      | A    | R   | A    | R   | A    | R         | A    |
| Chi-square                                       | 23                     | 92.1 | 8   | 94.0 | 16  | 92.7 | 9         | 94.0 |
| Entropy                                          | 22                     | 92.1 | 8   | 94.0 | 11  | 96.7 | 9         | 94.0 |

Table 6. Results obtained by the genetic algorithm applied to credit card database.

| $P_m(0 \rightarrow 1)$ | Best classifier |       |   | Worst classifier |       |   | Medium Performance |       |        |
|------------------------|-----------------|-------|---|------------------|-------|---|--------------------|-------|--------|
|                        | R               | A     | F | R                | A     | F | R                  | A     | G      |
| 0.0001                 | 8               | 89.45 | 1 | 4                | 76.37 | 1 | 4.2                | 86.67 | 316.90 |
| 0.005                  | 9               | 91.98 | 1 | 4                | 88.19 | 1 | 6                  | 89.89 | 657.55 |
| 0.01                   | 7               | 90.72 | 1 | 6                | 87.76 | 1 | 6.5                | 89.66 | 710.40 |



Table 7. Decision trees generated from credit card database

| Feature selection<br>to ramification<br>criteria | Significance level (%) |      |     |      |     |      |           |      |
|--------------------------------------------------|------------------------|------|-----|------|-----|------|-----------|------|
|                                                  | none                   |      | 0.1 |      | 0.5 |      | suggested |      |
|                                                  | R                      | A    | R   | A    | R   | A    | R         | A    |
| Chi-square                                       | 57                     | 82.3 | 6   | 87.3 | 32  | 85.7 | 9         | 86.9 |
| Entropy                                          | 54                     | 83.5 | 6   | 87.3 | 28  | 86.5 | 9         | 86.9 |

Tables 8 and 9 show the results obtained from the genetic algorithm and the decision tree applied to the diagnosis of indian diabetes.

Table 8. Results obtained by the genetic algorithm applied to indian diabetes database.

| $P_m(0 \rightarrow 1)$ | Best classifier |       |   | Worst classifier |       |   | Medium Performance |       |        |
|------------------------|-----------------|-------|---|------------------|-------|---|--------------------|-------|--------|
|                        | R               | A     | F | R                | A     | F | R                  | A     | G      |
| 0.0001                 | 5               | 82.17 | 1 | 8                | 73.64 | 1 | 6.7                | 78.64 | 322.80 |
| 0.005                  | 9               | 84.88 | 1 | 8                | 79.07 | 1 | 9.7                | 82.27 | 640.75 |
| 0.01                   | 9               | 84.88 | 5 | 9                | 81.01 | 1 | 9.3                | 83.70 | 785.50 |

Table 9. Decision trees generated from indian diabetes database

| Feature selection<br>to ramification<br>criteria | Significance level (%) |      |     |      |     |      |           |      |
|--------------------------------------------------|------------------------|------|-----|------|-----|------|-----------|------|
|                                                  | none                   |      | 0.1 |      | 0.5 |      | suggested |      |
|                                                  | R                      | A    | R   | A    | R   | A    | R         | A    |
| Chi-square                                       | 100                    | 72.1 | 7   | 75.6 | 58  | 74.8 | 19        | 74.4 |
| Entropy                                          | 93                     | 70.9 | 7   | 75.6 | 32  | 73.3 | 17        | 72.9 |

The best decision tree had better performance than only 2 (from 60) fuzzy rule-based classifiers. The best fuzzy rule-based classifier was found 6 times by the genetic algorithm. It was a surprising fact due to the complexity of this problem. The fuzzy classifier was formed by 9 rules (from 80) and it used 6 features. It was much more accurate than the best decision tree.

## 5. Final considerations

As it can be easily noticed, the genetic algorithm employed to select subsets of fuzzy rule bases is very simple. Its main characteristics are:

1. it just used the basic genetic operators;
2. it worked on populations with very few individuals;
3. there was no interference in the initiation of the population in order to



introduce diversity;

4. it was not executed any action to avoid the occurrence of premature convergence.

In spite of its simplicity, the genetic algorithm obtained very good results in every case studied. The quality of a classifier was estimated by some characteristics considered important in this work: accuracy, compacity and comprehensibility. In these three characteristics the fuzzy rule-based systems surpassed the decision tree. In almost all settings, the medium performances of the genetic algorithm were better than the best ones achieved by the decision tree software.

A broader comment must be made to comprehensibility. The superiority of linguistic fuzzy rules to numerical ones is irrefutable. The antecedent part of a rule obtained from a decision tree branch is more complex: inconstant amount of conditions and multiple evaluation of a feature. On the other hand, the linguistic fuzzy rules studied in this work has just one condition, using only one feature. On databases with high dimensions this characteristic is more important. Thus, acquiring knowledge from these linguistic rules is easier and more natural than by means of a decision tree.

An interesting fact is the implicit role of using the genetic algorithm to select a subset of decomposed zero-order TSK fuzzy rules. As this method generates rule bases with one condition and the evolutive process is responsible for finding the best subset of rules, only few features are used to classify new patterns, suggesting that these fields are the most important ones in the database. For instance, in the diagnosis of breast cancer, only 5 from 9 features are needed to a correct diagnosis. Thus, beyond the classification task, the method performed the selection of the most important attributes.

Future studies may consider the following topics, some of them being the focus of our ongoing research: apply other methods to generate fuzzy rule bases from numerical data; redefine the linguistic labels used and introduce new ones; employ transformation techniques in order to deal with incomplete records; employ techniques to manipulate great databases; study some metrics able to establish how interesting a rule is; develop the genetic algorithm on parallel processing computers; study the effects of other strategies of encoding and population initiation; apply other genetic operators; allow the genetic algorithm to self-adapt.

Due to the very good results obtained from the cases studied, the method presented in this work reveals itself as a promising and fast tool to generate efficient, comprehensive and relatively short classifiers in diverse application areas.

## References

- [1] PEI, M., GOODMAN, E. D., PUNCH, W. F., DING, Y. Genetic Algorithms for Classification and Feature Extraction. *Classification Society of North America Annual Meeting*, June 1995.



- [2] GUERRA-SALCEDO, C., CHEN, S., WHITLEY, D., SMITH, S. Fast and Accurate Feature Selection Using Hybrid Genetic Strategies. *Proceeding of the 1999 Congress on Evolutionary Computation*, Washington, pp. 177-184, 1999.
- [3] VAFAIE, H., DE JONG, K. Robust feature selection algorithms. *Proceedings of the 5th IEEE International Conference on Tools for Artificial Intelligence*, Boston, pp. 356-363, 1993.
- [4] RICHELDI, M., LANZI, P. L. Improving Genetic Based Feature Selection by Reducing Data Dimensionality. *International Workshop on Evolutionary Computation*, Bari, July 1996.
- [5] YANG, J., HONAVAR, V. *Feature subset selection using a genetic algorithm*. Technical Report TR 97-02a, Iowa State University, May 1997.
- [6] HERRERA, F., LOZANO, M., VERDEGAY, J. L. *Fuzzy Connectives Based Crossover Operators to Model Genetic Algorithms Population Diversity*. Technical Report DECSAI-95110, Universidad de Granada, February 1995.
- [7] LEE, M. A., TAKAGI, H. Dynamic control of genetic algorithms using fuzzy logic techniques. *Proceedings on Fifth International Conference on Genetic Algorithms*, San Mateo, pp. 76-83, July 1993.
- [8] WONG, C.-C., LIN, N.-S. Rule extraction for fuzzy modeling. *Fuzzy Sets and Systems*, 88(1), pp. 23-30, 1997.
- [9] YUAN, Y., ZHUANG, H. A genetic algorithm for generating fuzzy classification rules. *Fuzzy Sets and Systems*, 84(1), pp. 1-19, 1996.
- [10] LEKOVA, A., MIKHAILOV, L., BOYADJIEV, D., NABOUT, A. Redundant fuzzy rules exclusion by genetic algorithms. *Fuzzy Sets and Systems*, 100, pp. 235-243, 1998.
- [11] CORDÓN, O., HERRERA, F., LOZANO, M. A Three-Stage Method for Designing Genetic Fuzzy Systems by Learning from Examples. *Proceeding of 4th International Conference on Parallel Problem Solving from Nature*, Berlin, pp. 720-729, 1996.
- [12] HERRERA, F., LOZANO, M., VERDEGAY, J. L. *A Learning Process for Fuzzy Control Rules using Genetic Algorithms*. Technical Report DECSAI-95108, Universidad de Granada, February 1995.
- [13] ISHIBUCHI, H., MURATA, T., TANAKA, H. Construction of Fuzzy Classification System with Linguistic If-Then Rules Using Genetic Algorithms (Chapter 11). *Genetic Algorithms for Pattern Recognition*, CRC Press, pp. 227-251, 1996.
- [14] EVSUKOFF, A., BRANCO, A.C.S., GENTIL, S. A Knowledge Acquisition Method for Fuzzy Expert System in Diagnosis Problems. *Sixth IEEE International Conference on Fuzzy Systems*, Barcelona, July 1997.
- [15] KOSKO, B. *Neural Networks and Fuzzy Systems - A Dynamical Systems Approach to Machine Intelligence*, Prentice Hall: New Jersey, 1992.