

A Two-Step Approach of Feature Construction for a Genetic Learning Algorithm

David García, Antonio González and Raúl Pérez

Departamento de Ciencias de la Computación e I.A.

CITIC-UGR University of Granada (Spain)

Email: {DGarcia, A.Gonzalez,Raul_Perez}@decsai.ugr.es

Abstract—Traditionally, fuzzy rule based models work with a fixed set of features to describe a particular problem. Our proposal is to use feature construction by means of functions in order to obtain new variables that allow us to get more information about the problem. In particular, we propose the use of previously defined functions over the input variables in the antecedent of the rules. This let us to know if a combination of variables is able to provide us with more information than each one of them separately. In addition, we use a structure that helps us to manage and also restrict the number of functions under consideration by the learning algorithm. We also present a new model of rule in order to represent this kind of knowledge by extending a basic learning fuzzy rule-based model. Finally, we show the experimental study associated with this work.

Keywords—Genetic Fuzzy Systems, Feature Construction, Iterative Learning Approach, Classification.

I. INTRODUCTION

One of the most important areas in the development of applications based on fuzzy logic has been and is the design of fuzzy rule-based systems. In the literature different types of fuzzy rules have been used for different applications. The most common ones are the linguistic rule [1] and the TSK rule [2]. The first one presents a traditional model of rule whose antecedent and consequent is composed of linguistic variables. The second one proposed a model based on rules whose antecedent is composed of linguistic variables and the consequent is represented by a function of the input variables, frequently a linear function. A very well known extension of the linguistic rule is the DNF rule in which each input variable takes as a value a set of linguistic terms whose members are joined by a disjunctive operator, whilst the output variable remains an usual linguistic variable with a single value associated. Another types of extensions of fuzzy rules have been proposed, such as rules with weights, rules with double consequent, relational rules, etc. Each one of these different types of rules have been proposed with the idea of improving the system accuracy or to improve the system interpretability or both of them.

Traditionally, in fuzzy rule-based systems the number of variables in the antecedent is fixed, so now we want to explore the way to include new variables in the antecedent that result as a combination of the existing ones in order to improve the system description. That means that we need to apply feature construction techniques.

Many papers have been written regarding feature construction. In [3] a method is proposed in order to reduce

the set of features by means of arithmetics expression trees that work only with numerical attributes. They focused the importance of making a feature selection before doing the feature construction. This is because the construction of new variables is more efficient after reducing the initial searching space. In other works as [4] feature construction is used in order to create new attributes that will be the input data for another algorithm. The process of obtaining the new features is optimized by means of genetic algorithms. On the other hand, as shown in [5], feature construction is proposed for searching effective features sets that improves the quality of recognition methods. This is obtained using the composition of fuzzy relations.

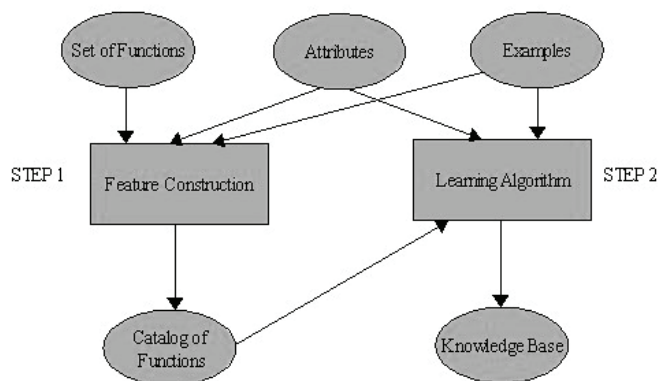


Fig. 1. Rule extraction process

In general, our proposal consists of incorporating new attributes in the learning process for obtaining an improved knowledge of the problem. In concrete, we can describe our proposal as a process with two steps: the first one is oriented to construct new attributes from the combination of the initial attributes by a set of predetermined functions. An information measurement is defined for evaluating the relevance of each new attribute. This step finishes returned the set of the best attributes that we call *catalog*. The second step takes the catalog and the training example set for obtaining a knowledge base that solves the learning problem. Figure 1 shows a graphical representation of the mentioned steps as far as the tasks taken into account in each one of them.

Next section shows a brief introduction about the rule

model we propose. Section III explains the feature construction step showing a pseudo code with the algorithm used to obtain the most relevant features and the inference process. The fourth section describes the most relevant details of the learning algorithm NSLV, and the necessary changes done in order to extract fuzzy rules with functions in the antecedent. Section V shows the experimental results obtained as well as the study related to these results. Finally, last section is devoted for conclusions and future work related to this study.

II. FUZZY RULES WITH FUNCTIONS IN THE ANTECEDENT

In this section we introduce the use of a new model of rule that can include functions as a component of the antecedent. First of all we will remind the linguistic fuzzy model of rules.

Linguistic rules present the following structure:

IF X_1 is A_1 and X_2 is A_2 and ... and X_n is A_n

THEN Y is B

where $X_1, X_2, \dots, X_n$ are the antecedent variables defined on the universes $U_1, U_2, \dots, U_n$, and Y is the consequent variable defined on the universe V . $A_1, A_2, \dots, A_n$ and B are fuzzy subsets defined on the respective universes.

We want to improve the capacity of knowledge representation of fuzzy models allowing that a function can be included as part of the description of a rule. To extend the rule model is easy since that a function can be considered like other variable. So, let's consider the following rule:

IF X_1 is A_1 and $SUM\{X_1, X_2\}$ is A_3

THEN Y is B

where $SUM(X_1, X_2)$ would be defined as $X_1 + X_2$ and A_3 is a fuzzy value that belongs to the function's domain.

We can interpret that $Z = SUM(X_1, X_2)$, being Z a linguistic variable defined on a fuzzy domain where A_3 is a fuzzy label that is member of that domain. Therefore, to work with functions in the antecedent of the rule does not modify the traditional inference process of the linguistic rules.

Let's suppose the following variables X_i, X_j , are defined in a fuzzy bounded domain $U_i = [inf_i, sup_i]$ and $U_j = [inf_j, sup_j]$ respectively. The range for the resultant variable Z , that represent the function $f(X_i, X_j)$, will depend on the operation selected in each case, but in general, it could be defined as $[min_{x_i \in U_i, x_j \in U_j} f(x_i, x_j), max_{x_i \in U_i, x_j \in U_j} f(x_i, x_j)]$. For this range, we consider a fixed number of uniformly distributed linguistic labels in order to define the domain of the new variable (in this work, this value is fixed to five).

Below we show an example of the reasoning to obtain the domain for a new variable obtained by applying a function over two input variables.

Given variables X_1 and X_2 , with the fuzzy domain shown in Fig. 2(a) and Fig. 2(b) respectively, we can define the

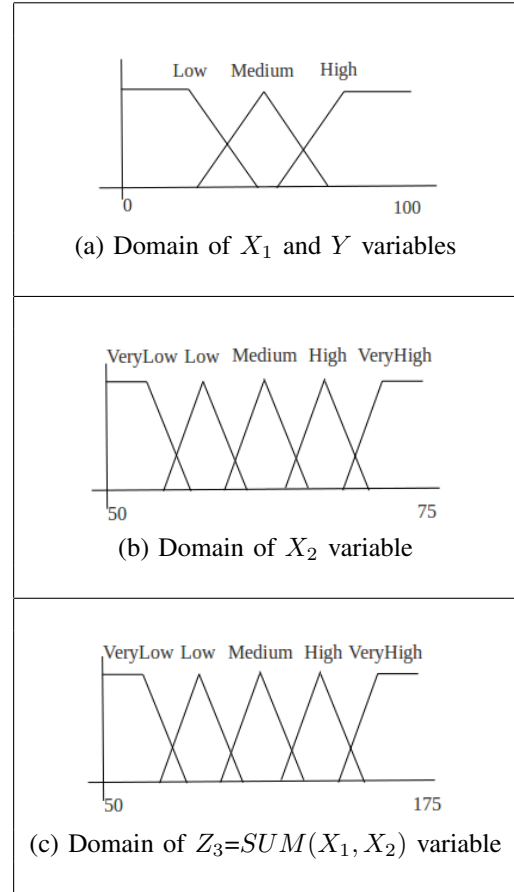


Fig. 2. Domains associated to the variables of the example

domain of the resultant variable $Z_3 = SUM(X_1, X_2)$ as is shown in Fig. 2c.

Now we can show an example of the particular type of rules we are using:

IF X_1 is *Low* and $SUM\{X_1, X_2\}$ is *Medium*

THEN Y is *Medium*

where Y is the consequent variable that takes values on the same fuzzy domain as X_1 (Fig. 2(a)).

III. STEP 1: FEATURE CONSTRUCTION

The first step of the learning process tries to find a set of new attributes that allow us to acquire more information about the problem. As it is shown in Figure 1, the input of this step consists in the initial variables set (V), the training example set (E) and a set of predetermined functions (F). The new variables are constructed applying the functions over some of the input variables. Although the general model could work with any kind of functions, from now on we will only work with binary functions, that is, functions over two variables. The relevance of each new attribute is evaluated using an information measure on the examples from the training set. The following code describes the process:

- FOREACH **f** in **F**
 - FOREACH subset **S** of **V**
 - * $info(f(S), E)$ is a measure of the relevance of the f function with S variables, calculated on the training set.
 - * IF $(info(f(S), E)) > k$, being k a threshold that establishes the minimum value for a function to be selected.
 - $f(S)$ is added to the final set of new attributes.
- RETURN the set of new attributes.

We take **F** as a set of previously defined (by an expert) functions. For each one of these functions we consider a subset of the input variables set. In order to know the relevance degree of the function over the set of examples we use an information measure. This measure is detailed below [6], [7]:

$$\rho(X, Y) = \frac{I(X, Y)}{H(X, Y)} \quad (1)$$

where

$$I(X, Y) = \sum_x \sum_y -p(x, y) \log_2 \left(\frac{p(x)p(y)}{p(x, y)} \right) \quad (2)$$

and $H(X, Y)$ is the Shannon entropy over two variables, defined as

$$H(X, Y) = \sum_x \sum_y -p(x, y) \log_2 p(x, y). \quad (3)$$

The ρ measure estimates the dependence between two generic variables X and Y in the following way: values of $\rho(X, Y)$ close to zero determine a high degree of independence of both variables, whereas values close to one demonstrate a high degree of functional dependency between them. This is the expression we calculate:

$$\rho(Z_k, Y) = \frac{I(Z_k, Y)}{H(Z_k, Y)} \quad (4)$$

where $Z_k = f_k(X_i, X_j)$ is the particular function to be evaluated and Y is the classification measure. In these expressions, Z_k is a new variable.

Once we have this information measure we compare it with a threshold (k) that establishes the minimum value for a function to be selected. The comparison method is explained as follows. We consider the value M_I as the maximum information given by any possible initial variable and it is defined by:

$$M_I = \max_p(\rho(X_p, Y)), \forall p \in \{1..n\} \quad (5)$$

being **n** the number of initial variables and **Y** the classification variable. Then, the comparison expression would be:

$$\rho(Z_k, Y) > \alpha * M_I \quad (6)$$

being $\alpha \in [0, 1]$ and $k = \alpha * M_I$ the threshold. For the experimental study we considered $\alpha = 0.9$.

So, if the function's information measure exceeds the established threshold, this new variable is added to the final set of new attributes. Finally, this set of new attributes is returned at the end of the process.

The output of the previous algorithm is used to define a new structure that we will call catalog of functions (CF). In a formal way it is an index set where each entry is a relevant function to be consider by the learning algorithm. For a given i, j, r , we define a catalog entry as a three-position vector where the first and the second values make reference to the variables involved in the function (that is, the index of the variables in the chromosome) and the last one is the function applied to these variables. An illustrative example for the previous given values would be $f_r(X_i, X_j)$. By this way, we will be able to associate to this function an integer representing the index of the function in the catalog. This integer can be described by means of the expression $\beta(i, j, k)$ because it is ordered in the catalog. Now we can define the new proposed model of fuzzy rules as follows:

IF X_1 is $A_1 \wedge \dots \wedge X_n$ is $A_n \wedge \{\wedge_{(i,j,k) \in H} f_k(X_i, X_j) \text{ is } E_k\}$
THEN Y is B

where A_i are the antecedent values, B is the consequent value and $H \subseteq CF$ is an index subset defining the specific function participating in the rule. We also define $Z_{\beta(i,j,k)} = f_k(X_i, X_j)$, as a new variable resulting of applying the function f_k over the variables (X_i, X_j) . On the other hand, $E_1, \dots, E_k$ are fuzzy values for $Z_1, \dots, Z_k$.

As we have been managing an extended version of this kind of rules (**DNF rules**), we are able to assign a set of values from its domains to each antecedent variable, so now the new rule definition is

IF X_1 is $\tilde{A}_1 \wedge \dots \wedge X_n$ is $\tilde{A}_n \wedge \{\wedge_{(i,j,k) \in H} f_k(X_i, X_j) \text{ is } \tilde{E}_k\}$
THEN Y is B

with $\tilde{A}_i$ being a set of fuzzy values on universe U and $\tilde{E}_i$ a set of fuzzy values defined on a fuzzy domain. From now on, we will refer to this kind of rules as Fuzzy Functional Rules (**FFR**). It is important to mention that the new variables Z_i will have their own domain depending on the domains of the variables involved in the function. That is, it only has sense to apply the functions to compatible variables (taking into account the units).

Our model includes a weight for each rule. This weight, a value in $[0, 1]$, is interpreted as a measure of its prediction capability. In the inference process, the weight of each rule modifies the adaptation to the example with the antecedent of the rule by means of a product operation. The winner rule is the one that obtains the highest value in this adaptation process.

IV. STEP 2: LEARNING ALGORITHM

Our goal is to design a learning algorithm capable to obtain a knowledge base that better represents our problem. We think that the inclusion of functions in the antecedent of the rules can increase the amount of information about the set of examples we are working with. In order to do that we will take as a base algorithm the learning algorithm of fuzzy rules NSLV. Now we are going to describe the behavior of NSLV as well as the extension of this algorithm that is able to include new features, NSLV-F.

A. The learning algorithm NSLV

NSLV[8] is a genetic fuzzy rule learning algorithm that use a sequential covering strategy [9]. This strategy reduces the problem of learning a disjunctive set of rules to a sequence of simpler problems, each requiring that a single conjunctive rule be learned.

SEQUENTIAL-COVERING(T,A,E,D)

- Learned-rules $\leftarrow \{\}$
- Rule $\leftarrow \text{LEARN-ONE-RULE}(T,A,E)$
- while $\text{PERFORMANCE}(\text{Rule},E) > D$, do
 - Learned-rules $\leftarrow \text{Learned-rules} + \text{Rule}$
 - $E \leftarrow E$ -examples correctly classified by Rule
 - Rule $\leftarrow \text{LEARN-ONE-RULE}(T,A,E)$
- Learned-rules $\leftarrow \text{sort Learned-rules accord to PERFORMANCE over } E$
- return Learned-rules

where T is the target attribute, A is the attribute set, E is the set of examples and D is a threshold.

The sequential covering algorithm reduces the problem of learning a disjunctive set of rules to a sequence of simpler problems, each requiring that a single conjunctive rule be learned.

NSLV uses a genetic algorithm (GA) to implement the **LEARN-ONE-RULE** procedure. The input of this GA is a target attribute, representing the consequent variable, the complete set of antecedent variables and the set of examples, and the output is a single rule that covers at least some of the examples. The criteria for selecting the best rule in NSLV is based on obtaining a single rule that covers many of the positive examples and few of the negative examples. The idea is formulated through the consistency and completeness conditions for a rule [10]. The final criterium is the product of a consistency measure by the a completeness measure.

Therefore, each individual of the population represents a complete rule. NSLV uses a DNF model of rule and therefore each variable takes as value a set of linguistic terms whose members are joined by a disjunctive operator. The genetic algorithm represents each DNF rule by three different elements or levels (Fig. 3):

- *The variable level.* This level contains a gene for each predictive variable involved in the problem. Each gene represents a real value that it is interpreted as the relevance degree of a predictive variable on the rule. Furthermore, a special gene is added in this level that

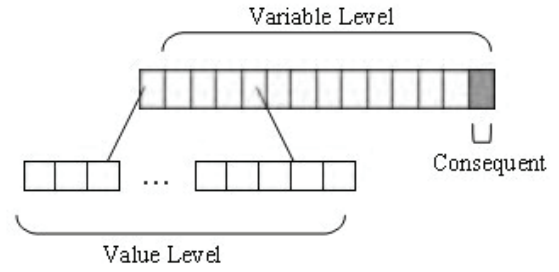


Fig. 3. A rule representation in NSLV

it is interpreted as activation threshold, that is, the variables which associated genes have values less than the threshold are not considered in the antecedent of the rule. Therefore, the use of this level allow us to develop an embedded feature selection process [11].

Two genetic operators are used in this level, two points crossover and real uniform mutation. The generation of this level in the initial population is based on information measure obtained from the training set. This measure permits to establish an initial relevance degree of each variable in the problem. The obtained values are used for all individual of population on this level. The activation threshold is randomly generated between the maximum and minimum value of the initial relevance degree obtained. A more detailed description can be found in [12].

- *The value level.* It is composed of the sequence of assignments to the predictive variables, where each assignment variable/value is represented by a binary string. The complete level is composed by concatenation of the binary strings representing the assignments variable/values for all input variables. The assignment of a certain variable will appear in the description of the rule if the value of its associated gene in the variable level permits to be considered to the variable relevant for this rule.
 - *The consequent level.* It codes the value of the classification variable of the rule. This level is composed by one gene that is represented by a integer value.
- The integer uniform mutation is the unique genetic operator considered on this level and in the initial population this level is randomly generated.

NSLV uses a steady state genetic algorithm. In this case, the selection process is the following: two individuals of the

population are selected, the crossover operators on levels are applied between them obtaining two new individuals. The mutation operators are used for modifying the new individuals. They are evaluated and interchanged by the two worst individuals of the population. Therefore, in each generation of the genetic process only two new individuals are generated and evaluated.

B. NSLV-F: A genetic learning algorithm of fuzzy functional rules

In the current section we explain the changes made in the latest version of NSLV in order to be able to learn FFR. Basically we need to modify the learning algorithm code so it is possible to learn this new type of rules. Thus, the first change takes place in one's own representation of each rule. In this way, we have created a new substructure called function level that encodes the active functions that appear in the rule.

The function level. This level represents from the total set of functions, those that are part of the antecedent of the rule. The individual keeps the same number of levels that the older version apart from this new one. The maximum number of functions that can be included in a single rule is defined by a parameter. In the experiments we have considered up to 10 functions in each rule. This function level use an integer coding where 0 indicates no function considered and any other positive value will refer to the index of the function in the catalog (Fig. 4).

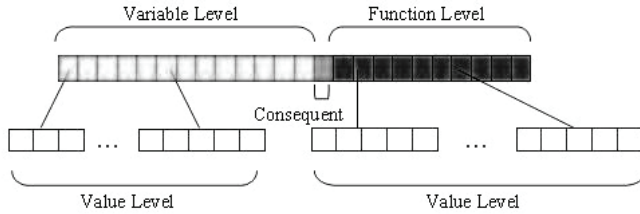


Fig. 4. A rule representation in NSLV-F

Paying attention to genetic operators, we keep the same operators for variables, antecedent values and consequent values used in NSLV. As the function level uses an integer codification, we use two points crossover and a mutation operators for integer coding. The mutation operation defined for this level has associated a probability. This probability establishes if one value of the structure is modified (in the experimental study this value has been set to **0,2**). When the previous fact occurs, a position of the level is randomly selected. If that position has zero value a new random value is generated for selecting a new function of the Catalog. If the position has a different value from zero, then zero value is assigned to that position.

As now we have a new value level associated to the function level which uses a binary coding, we also define the same two genetic operators that were defined for the value level associated to the variable level. That is, two points crossover and binary uniform mutation.

In the initial population all individuals start having the genes of their function level with zero value. That means that at the beginning of the genetic process no functions are considered in the chromosome.

V. EXPERIMENTAL STUDY

In this section we compare the behavior of the new learning algorithm with respect to the base algorithm NSLV. We also compare it with other classical fuzzy rule learning algorithms. The experimental study was performed on 9 databases of the UCI Repository of Machine Learning Databases and Domain Theories [13] and also on Tao database [14].

TABLE I describes the previously mentioned databases, where each row represents the number of examples (#E), variables (#V), continuous variables (#C), nominal variables (#N), classes (#Cl) and missing values (M).

TABLE I
DATABASE USED IN THE EXPERIMENTAL STUDY.

Database	#E	#V	#C	#N	#Cl	M
bpa	345	6	6	0	2	No
brd	106	11	4	7	7	No
h-c	303	13	6	7	2	Yes
irs	150	4	4	0	3	No
pim	768	8	7	1	2	No
son	208	60	60	0	2	No
tao	888	2	2	0	2	No
trf	748	4	4	0	2	No
wdbc	569	30	30	0	2	No
wpbc	198	33	33	0	2	Yes

We have considered in all databases five uniformly distributed linguistic labels to define the domain of the continuous variables. In both algorithms, we use the missing values treatment described in [15]. The results have been obtained by ten-fold cross validation, using the same partition for all algorithms. Parameter *m* determines the minimum number of individuals in the population that must be in a class. In all experiments, this value was fixed to $\frac{|pop|}{|cl|+1}$, where *|pop|* is the size of the population and *|cl|* is the number of classes. The size of the population was fixed to 100.

First of all we are going make a brief description of the functions considered by the algorithm in the experimental study.

A. Functions

In this empirical part we have taken into account only three possible functions:

- SUM (X_i, X_j), defined as $X_i + X_j$
- PRODUCT (X_i, X_j) defined as $X_i * X_j$
- SUBT_ABS (X_i, X_j) defined as $|X_i - X_j|$.

The adding and subtract functions will not be considered with variables with non compatible units. It would not have sense to consider the $SUM(X_1, X_2)$ where X_1 and X_2 measure time and weight respectively. It is important to remark that the domain of the new variables generated will

TABLE II
RESULTS OBTAINED FOR TRAINING.

	NSLV	NSLV-F
bpa	66.9 (2)	71.2 (1)
brd	94.7 (1)	86.6 (2)
h-c	91.2 (2)	92.1 (1)
irs	96.8 (2)	97 (1)
pim	80.2 (2)	80.3 (1)
son	90.4 (2)	96.6 (1)
tao	82.4 (2)	85.9 (1)
trf	77.1 (1)	77 (2)
wdbc	95.8 (2)	97.1 (1)
wpbc	80.1 (1)	79.5 (2)
Rnk	1.7	1.3
Pos	2	1

TABLE III
RESULTS OBTAINED FOR TEST.

	NSLV	NSLV-F
bpa	56.7 (2)	59.9 (1)
brd	55.6 (2)	57.5 (1)
h-c	77.8 (2)	78.4 (1)
irs	94 (2)	94.6 (1)
pim	72.6 (2)	74.8 (1)
son	69.1 (2)	73.9 (1)
tao	81.6 (2)	85.1 (1)
trf	75.9 (2)	76 (1)
wdbc	93.6 (2)	95.6 (1)
wpbc	74.2 (2)	77.3 (1)
Rnk	2	1
Pos	2	1

TABLE IV
RESULTS OBTAINED FOR RULES.

	NSLV	NSLV-F
bpa	6.1 (1)	8.9 (2)
brd	19.2 (1)	20.1 (2)
h-c	11.4 (1)	13.3 (2)
irs	4 (2)	3.4 (1)
pim	12.8 (1)	13.3 (2)
son	9.9 (2)	9.3 (1)
tao	3.4 (1)	4.1 (2)
trf	3.4 (2)	3 (1)
wdbc	4.5 (1)	5 (2)
wpbc	3.4 (2)	3.3 (1)
Rnk	1.4	1.6
Pos	1	2

TABLE V
A RULE SET FROM TAO DATABASE

```

IF SUM(var0,var1) IS {High or Very High}
THEN class IS C1 Weight=0.884278

IF var0 IS {Very Low or Low or Medium} AND
   SUM(var0,var1) IS {Very Low or Low}
THEN class IS C0 Weight=0.908361

IF var0 IS {Very High}
THEN class IS C1 Weight=1

```

also be divided into five uniformly distributed linguistic labels.

B. Comparing NSLV with NSLV-F

We have focused our study in the following indicators: accuracy on training set (Training), accuracy on test set (Test) and number of rules (Rules).

Training and Test figure the accuracy of the learned model. On the other hand, Rules define the size and the interpretability of the knowledge base.

To perform the comparison was used Bonferroni-Dunn test [16]. It establishes that two classifiers have differences if they differ in at least one critical distance. This distance depends on the number of databases, the number of learning algorithms to compare and a critical value q_α , with error α . In that case we took 0.05, $q_\alpha = 1.960$. Finally, for 10 databases and two classifiers, the critical distance (CD) is **0,619**.

The results are shown in TABLE II, TABLE III and TABLE IV. The first column indicates databases evaluated. The second and the third columns represent the results for NSLV and NSLV-F algorithms respectively. First row shows indicators evaluated and the last one contains the value estimated for Bonferroni-Dunn test.

As we can see, there are significant differences in test (TABLE III). This means that NSLV-F improves the accuracy with respect to NSLV. Another important fact is that the results obtained for Training (TABLE II) and Rules (TABLE IV) have no significant differences, so the number of learned

rules are approximately the same and they cover almost the same number of examples. Anyway, the improvement obtained in the Test results could be due to the greater ability to generalize of NSLV-F. The inclusion of new variables created by input variables combinations (by means of functions), allow us to obtain more general rules that increase the prediction capacity of the model.

TABLE V shows a rule set obtained by NSLV-F from Tao database. This database is a synthetic problem composed by two real predictive variables, $var0$ and $var1$, and two classes, $C0$, $C1$. We can see that the knowledge base is composed by three rules and it only uses $var0$ and a combination of both initial variables $SUM(var0, var1)$. The point is that the variable $var1$ does not appear alone in any rule. So that, we would be able to conclude that, in this particular case, the new variable $SUM(var0, var1)$ makes possible to obtain more general rules or even gives more information that the variable $var1$ on its own. So we find one of the main reasons why this new model is proposed.

C. Comparing NSLV-F with other fuzzy rule learning algorithms on accuracy

In this section we compare NSLV-F with other well-known fuzzy rule learning algorithms. We have used KEEL platform [17] and for all algorithms we have used the recommendation settings by KEEL. The fuzzy algorithms are the following:

- Fuzzy Ishibuchi99 [18]: Its genetic algorithm handles each fuzzy rule as an individual, and a fitness value is assigned to each rule. It uses linguistic values with

TABLE VI
COMPARING NSLV-F WITH OTHER FUZZY RULE-BASED ALGORITHMS

	GP	GAP	Ishibushi	Chi	SAP	NSLV-F
bpa	55.32 (6)	60.82 (2)	58.25 (4)	57.87 (5)	63.49 (1)	59.9 (3)
brd	55.09 (3)	56.9 (2)	51.72 (5)	43.9 (6)	54 (4)	57.5 (1)
h-c	78.17 (2)	73.49 (5)	77.87 (3)	50.77 (6)	76.88 (4)	78.4 (1)
irs	86.66 (6)	90.66 (5)	95.33 (1.5)	92.6 (4)	95.33 (1.5)	94.6 (3)
pim	73.58 (3)	73.83 (2)	68.75 (6)	73.05 (4)	72.53 (5)	74.8 (1)
son	70.69 (2)	61.07 (6)	61.59 (5)	61.74 (4)	69.14 (3)	73.9 (1)
tao	82.82 (4)	76.15 (6)	81.19 (5)	84.36 (3)	86.91 (1)	85.1 (2)
trf	76.46 (3)	76.19 (5)	76.2 (4)	77.28 (1)	76.47 (2)	76 (6)
wdbc	93.15 (3)	81.89 (5)	92.56 (4)	95.06 (2)	71.17 (6)	95.6 (1)
wdbc	76.83 (4)	88.23 (1)	76.35 (5)	69.6 (6)	80.91 (2)	77.3 (3)
Rank	3.6	3.9	4.25	4.1	2.95	2.2

fixed membership functions as antecedent fuzzy sets, so a linguistic interpretation of each fuzzy rule is easily obtained.

- Fuzzy GAP [19]: It classifies an instance as a result of a quadratic combination of its features. The weights of such combination are fitted as a quadratic discriminant. An instance is classified with the class that has the better value for the quadratic combination of its features.
- Fuzzy SAP [19]: A Simulated Annealing algorithm is used to learn a fuzzy classifier. For each hypothesis, the number of labels and number of rules must be given. In same way, as it is possible to manage any combination of conjunction and/or disjunctions in the antecedent part of a fuzzy rule, a maximum deep tree size must be given too. Those parameters are, in conjunction with Simulated Annealing typical parameters, of capital importance in the evolution of this method.
- Fuzzy GP [20]: Combining genetic programming operators with Simulated Annealing Search to evolve fuzzy rule based classifiers.
- Fuzzy Chi-RW: This algorithm takes as base algorithm the Chi et al. [21] but adding weights to rules. The algorithm is an adapted version of Wang-Mendel algorithm[22] for classification problems.

In TABLE VI appears the result of comparison with the fuzzy methods mentioned before. In this case, for 10 databases and 6 classifiers, the critical distance is **1.05**. Bonferroni test reveals that there are significant differences in test parameter and that NSLV-F has the best result, being significantly better than GP, Chi and Ishibushi. There are not significant differences with the rest of the algorithms.

VI. CONCLUSIONS AND FUTURE WORK

In this work we have presented a learning algorithm that introduces the concept of feature construction by using functions as a way of creating new attributes. The basic idea is to consider not only the knowledge given by the input variables but also the knowledge that can be extracted from the combination of these input variables. In order to do that we combine the variables by using some previously defined functions. These functions will be managed by means of a structure that will also contain the most promising ones taking into account an information measure. A new model

of rule has been also proposed in order to represent this kind of knowledge. In addition, we took as basic algorithm a learning fuzzy rule-based one and we extended it to obtain this new model. Finally, the experimental results show that our proposal gives us the possibility of finding more general rules and also increases the prediction capability of the model. Our purpose is to go on improving the current model in order to consider more functions in the future that could give us better results. Anyway, this is an initial approximation and it will be necessary to go on searching for other functions that allow us to improve the accuracy and efficiency of the process.

ACKNOWLEDGMENT

This work is partially supported by the projects P09-TIC-04813.

REFERENCES

- [1] E. H. Mamdani, S. Assilian, "An experiment in linguistic synthesis with fuzzy logic controller". *International Journal of Man-Machine Studies*, vol. 1, pp.1-13, 1975.
- [2] T. Takagi, M. Sugeno, "Fuzzy identification of systems and its application to modeling and control". *IEEE Transactions on systems, Man and Cybernetics*, vol. 15, pp. 116-132, 1985.
- [3] K. Drodz, H. Kwasnicka, "Feature Set Reduction by Evolutionary Selection and Construction". *Lecture Notes in Computer Science 6071 LNAI (PART2)*, pp. 140-149, 2010.
- [4] R. Alfred, "Optimizing Feature Construction Process for Dynamic Aggregation of Relational Attributes". *Journal of Computer Science* 5 (11), pp. 864-877.
- [5] H. A. Gvenir, M. akir, "Voting features based classifier with feature construction and its application to predicting financial distress". *J.Expert Systems with Applications* 37 (2), pp. 1713-1718.
- [6] S. Kullback, *Information Theory and Statistics*, Gloucester, Mass, 1978.
- [7] L. Wehenkel, "On uncertainty measures used for decision tree induction", *Proc. of IPMU'96*, pp. 413-418, 1996.
- [8] A. González, R. Pérez, "Improving the genetic algorithm of SLAVE", *Mathware & Soft Computing*, 16, pp. 59-70, 2009.
- [9] T. Mitchell, *Machine Learning*. Ed. MacGraw-Hill, 1997.
- [10] A. González, R. Pérez, "Completeness and consistency conditions for learning fuzzy rules". *Fuzzy Sets and Systems*, vol. 96, pp. 37-51, 1998.
- [11] A. González, R. Pérez, "An analysis of the scalability of an embedded feature selection model for classification problems". *Proceedings of the Information Processing and Management of Uncertainty on Knowledge-Based Systems IPMU 2006*, pp. 1949-1956, 2006.
- [12] A. González, R. Pérez, "Selection of Relevant Features in a Fuzzy Genetic Learning Algorithm". *IEEE Transactions on System, Man, and Cybernetics Part B*, vol. 31(3), pp. 417-425, 2001.
- [13] A. Asuncion, D.J. Newman, "UCI Machine Learning Repository" Irvine, CA: University of California, School of Information and Computer Science, 2007. Available: <http://www.ics.uci.edu/mllearn/MLRepository.html>.

- [14] E. Bernardó-Mansilla, X. Llorá, J. Garrel (2002) XCS and gale: "A comparative study of two learning classifier". In: P. Lanzi, W. Stolzman, S. Wilson (eds) *Advances in Learning Classifier Systems, LNCS*, vol. 2321. Springer, Heidelberg, pp. 115-132.
- [15] A. González, R. Pérez, "SLAVE: to genetic learning system based on an iterative aproach". *IEEE Transaction on Fuzzy System*, vol. 27 (2), pp. 176-19, 1999.
- [16] O. Dunn, "Multiple Comparisons among means". *Journal of the American Stadistical Association*, 1961.
- [17] Alcalá J. et al, 2009. KEEL: A Software Tool to Assess Evolutionary Algorithms to Data Mining Problems. *Soft Computing* 13:3 307-318
- [18] Ishibuchi, H. et al, 1999. Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 29:5 pp. 601-618.
- [19] Sánchez, L. and Couso, I., 2001. Combining GP Operators With SA Search To Evolve Fuzzy Rule Based Classifiers. *Information Sciences* 136:(1-4), pp 175-192.
- [20] Otero, J. and Sánchez, L., 2006. Induction of descriptive fuzzy classifiers with the Logitboost algorithm. *Soft Computing* 10:9, pp 825-835.
- [21] Chi, Z., Wu, J., Yan, H., Handwritten Numeral Recognition Using Self-Organizing Maps and Fuzzy Rules. *Pattern Recognition*, 28:1, pp. 59-66, 1995.
- [22] Wang, L. X., Mendel, J. M., Generating Fuzzy Rules by Learning from Examples. *IEEE Transaztiond on Systems, Man and Cybernetics*, 22:6, pp. 1414-1427, 1992.