An Inductive Approach for Learning Fuzzy Relational Rules*

E. Aguirre
Departamento de Ciencias
de la Computación e I.A.
Universidad de Granada
eaguirre@decsai.ugr.es

A. González
Departamento de Ciencias
de la Computación e I.A.
Universidad de Granada
A.Gonzalez@decsai.ugr.es

R. Pérez
Departamento de Ciencias
de la Computación e I.A.
Universidad de Granada
fgr@decsai.ugr.es

Abstract

An inductive approach for learning fuzzy relational rules is described. Fuzzy relational rules are rules in which fuzzy relations in the antecedent parts of the rules are allowed. These rules allow us to use an extended model of rule with a greather capability to represent knowledge in a similar human way, but with an additional complexity in the search process.

Keywords: Learning, Fuzzy rules, Fuzzy Relations, Genetic Algorithms.

1 Introduction

System modelling with fuzzy rule-based systems has been successfully applied in different problems and domains. The success of this application has to do with the research in algorithms able to learn fuzzy rules. Different types of fuzzy rules has been used in this process, as Mamdani rules [5], DNF rules [2], TSK rules [7].The use of a particular kind of rule (or in general the knowledge representation used) is related with the competence of the modeling to express the behaviour of the real systems in a understandable way (interpretability of the model) and the capability to faithfully represent the system (accuracy of the model).

The aim of this proposal consists of exploring a new model of fuzzy rule. In [8, 9] the idea of using fuzzy relations in the antecedent part of the rules and a mechanism to reason with these types of rules is introduced. The *fuzzy relational rules* allow us to obtain fuzzy models with a good interpretability-accuracy trade-off, since a fuzzy relational rule generates flexible partitioning of the input space, and moreover, the inclusion of fuzzy relations in the antecedent of the rules can simplify the description of many systems. However, a learning algorithm of fuzzy relational rules must face the increase of the search in the rule space, and therefore, the increase of the complexity of the learning algorithm.

On the basis of a previous learning algorithm, called SLAVE [1, 2], we propose an extension that is able to manage fuzzy relational rules. The complexity of the inclusion of fuzzy relations in the antecedent of the rules is limited by using a *Relation Index* that contain the relevant relations to be considered by the algorithm.

In the next section we propose a formalism to define the fuzzy relational rules and we show how to use these rules. Section 3 describes the Relation Index, a mechanism to limit the complexity of using fuzzy relations in rules. Next section outlines an extension of the learning algorithm SLAVE that is able to manage fuzzy relational rules. Finally, section 5 shows the behaviour of the extension on several examples.

2 Fuzzy Relational Rules

Tradicionally, fuzzy models are based on families of fuzzy rules of the form:

$$\text{IF } X_1 \text{ is } A_1 \text{ and } X_2 \text{ is } A_2 \text{ and } \ldots \text{and } X_n \text{ is } A_n$$

$$\text{THEN } Y \text{ is } B$$

where $X_1, X_2, \ldots, X_n$ are the antecedent variables defined on the universes $U_1, U_2, \ldots, U_n$, and Y is

*This work has been supported by the CICYT under Project TIC2002-04146-C05-02

the consequent variable defined on the universe V, $A_1, A_2, \ldots, A_n$ and B are fuzzy subsets. The basic effect of using this kind of rules can be seen as a rectangular partitioning of the input space.

In [8] the use of fuzzy models with relational antecedents is introduced. A fuzzy relational rule consists of a rule in which the antecedent requirements involve the satisfaction of a relationship among different variables, for example,

$$\text{IF } (X_1, X_2, \ldots, X_n) \text{ is R THEN } Y \text{ is } B$$

where R is a fuzzy relation, that is, a fuzzy subset of $U = U_1 \times U_2 \times \ldots \times U_n$. An example of a fuzzy relational rule is

$$\text{IF } X_1 \text{ is about the same as } X_2 \text{ THEN } Y \text{ is } B.$$

Obviously, this particular rule can be approximated by tradicional fuzzy rules, but the use of a fuzzy relational rule increases the interpretability of the knowledge, since it is described in a more similar human way, and moreover, it increases the simplicity of the model since only one rule is needed whereas to represent the same knowledge more rules are needed in the previous model (at least one for each fuzzy value of the variables).

The idea is to include fuzzy relations in order to improve the interpretability and simplicity of fuzzy models but minimally extending the traditional model of fuzzy rule, that is, using fuzzy relations only when there are a clear improvement of the model due of its use. Moreover, in a first step, we only explore a restrictive definition of the previous one, in which, we only consider antecedent requeriments involving individual variables or the satisfaction of relationships between two variables, for example,

$$\text{IF } (X_1, X_2) \text{ is R and } X_3 \text{ is } A_3 \text{ THEN } Y \text{ is } B$$

where now R is a fuzzy subset of $U = U_1 \times U_2$.

The use of fuzzy relational rules allows more general partitions than the rectangular one, and obviously this is an extension of the tradicional models.

An interesting study of the use of these rules can be found in [9]. In the general case, the reasoning with this kind of rules need complex calculus, but when we use singletons as inputs the process is simpler. For an input like (x_1, x_2, x_3) with $x_i \in U_i$ for the previous rule, the output value is

$$B^*(y) = \tau \wedge B(y)$$

with $\tau = R(x_1, x_2) \wedge A_3(x_3)$ and $\wedge$ a t-norm like the minumum operator. When several rules are used $B^*(y)$ is taken as the maximum value among the different $B^*(y)$ for each rule.

The aim of this work is to develop a learning algorithm for fuzzy relational rules. However, the complexity of the problem increases considerately, since the possible binary relation between two selected variables among the set of the n original variables is obviously infinite, or in a practical case, too big. Therefore, a previous reduction of the relation candidates is needed, and in the next section we address a formalism to represent such reduction.

3 Relation Index

The start point is a set of possible antecedent variables $\{X_1, X_2, \ldots, X_n\}$ defined on universes $U_1, U_2, \ldots, U_n$ respectively, and a consequent variable Y defined on the universe V. We permit rules in which appear any of these previous variables and additionally some binary relationship between this variables. In order to restrict the possible relation to be considered by the learning algorithm we introduce a Relation Index in which we include the relevant relations to be considered in our problem. At this point, an expert definition of the relation index is needed, and it requires an effort for selecting relation candidates to be used by the learning algorithm.

The Relation Index (RI) is really managed as a index set. Let us suppose we numerate all the candidates relation. Thus, we define the index set RI as: if $\{(X_i, X_j) \text{ is } R_k\}$ is a relevant relation to be consider for the learning algorithm, then we put (i,j,k) in the index set RI.

Using the RI set, the formalism for a fuzzy relational rule we manage is

$$\text{IF } X_1 \text{ is } A_1 \wedge \ldots \wedge X_n \text{ is } A_n \wedge \{\wedge_{(i,j,k) \in H} [(X_i, X_j) \text{ is } R_k]\}$$
$$\text{THEN } Y \text{ is } B$$

where A_i are the antecedent values, B is the consequent value and $H \subseteq RI$ is an index subset defining the concrete relation participating in the rule. We denote this rule as $R_B(A, H)$, with $A = (A_1, A_2, \ldots, A_n)$.

Actually, we will manage an extended version of this rule where it is possible to assign a set of values from its domain to each antecedent variable of the rule, that is,

$$\text{IF } X_1 \text{ is } \tilde{A}_1 \wedge \ldots \wedge X_n \text{ is } \tilde{A}_n \wedge \{\wedge_{(i,j,k)\in H}[(X_i, X_J) \text{ is } R_k]\}$$

$$\text{THEN } Y \text{ is } B$$

with $\tilde{A}_i$ being a set of fuzzy values on universe U_i. The inference process associated to this rule is a simple extension of the general case. For an input $x = (x_1, x_2, \ldots, x_n)$, $x_i \in U_i$ the output is

$$B^*(y) = \tau(\tilde{A}_1, \ldots, \tilde{A}_n, H) \wedge B(y)$$

with

$$\tau(\tilde{A}_1, \ldots, \tilde{A}_n, H) = \tilde{A} \wedge \{\wedge_{(i,j,k)\in H} R_k(x_i, x_j)\}$$

and

$$\tilde{A} = \tilde{A}_1(x_1) \wedge \ldots \wedge \tilde{A}_n(x_n)$$

and

$$\tilde{A}_i(x_i) = \frac{max_{j\in S} A_{ij}(x_i)}{max_{j\in 1\ldots n_i} A_{ij}(x_i)}$$

where the domain of X_i is defined by n_i values $\{A_{i1}, A_{i2}, \ldots, A_{in_i}\}$ and $\tilde{A}_i$ is defined by the set of values contained in the index set S, that is,

$$\tilde{A}_i = \{A_{ij} | j \in S\}.$$

Obviously, when $S = \{1, 2, \ldots, n_i\}$ then $\tilde{A}_i(x_i) = 1$ and the component i doesn't affect to the calculus of τ. Therefore, the situation is the same to that in which this variable doesn't appear in the rule.

4 Learning Algorithm

In previous papers [1, 2] we have developed SLAVE an inductive learning algorithm of fuzzy rules based on the strategy of learning one rule, removing the data it covers, then iterating the process. This kind of algorithms are called *sequential covering* algorithms. A prototypical description of this family of algorithms can be found in [6].

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.

SLAVE 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 SLAVE 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 [2]. The final criterium is the product of a consistency measure by the a compleltness measure. The aim of this paper is to extend this algorithm to fuzzy relational rules. The structure of the algorithm is the same, and even the criterium for selecting rules. But several particular aspects need to be adapted to the new kind of rule.

One important aspect to solve with this new representation is the calculus of the number of positive and negative examples to a rule. This calculation is needed in the consistency and completeness definitions. It is necessary to adapt these definitions to the new structure of a rule.

Given a fuzzy relational rule $R_B(\tilde{A}, H)$, the number of positive examples of this rule is defined as the cardinal on a set of examples E of the fuzzy set of positive examples to the rule. This set is defined by given a membership degree to each example about the concept of "be positive" to the rule. This membership degree is defined by

$$\tau(\tilde{A}_1, \ldots, \tilde{A}_n, H) \wedge \frac{B(y)}{max_{B'} B'(y)}$$

representing the simultaneous adaptation of the example to the antecedent and the consequent of the rule. The number of negative examples is the cardinal of the fuzzy set of the negative examples to the rule. The membership degree of each example to this set is defined by

$$\tau(\tilde{A}_1, \ldots, \tilde{A}_n, H) \wedge \frac{max_{B' \neq B} B'(y)}{max_{B'} B'(y)}$$

representing the adaptation to the antecedent and any of the other possible value of the consequent variable different to that considered in the rule.

The genetic algorithm used in the learning process has a similar genetic representation to the representation proposed in SLAVE [1]. Each individual of the population represents a complete rule. An individual is composed by four different representations or levels (Fig. 1):

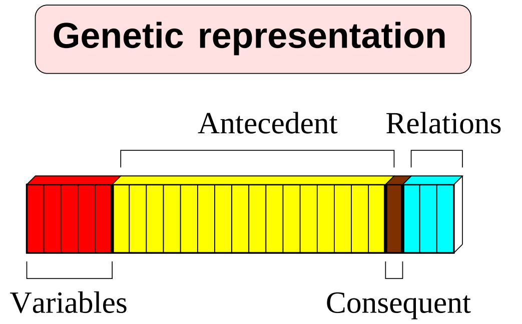

Figure 1: Representation of a individual rule

- *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.

- *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 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, this level can be considered a embedded feature selection process.

  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 maximun and mininum value of the initial relevance degree obtained. A more detailed description can be found in [3].

- *The antecedent 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.

  We use a binary representation and two genetic operator on this level, two points crossover and binary uniform mutation. The generation of this level in the initial population is the following: known the consequent value (generated previously in the consequent level), an example of this class is selected. The more specific antecedent for this example is obtained and it is used for coding the individual. This process is repeated for all individuals of the population. More information can be found in [3].

- *The relation level*. This level represents the relation set included in the antecedent of the rule. Each gene codes a possible relation of the RI. In this work, we have considered three as the maximun number of relations permitted in each rule.

  Each gene takes an integer value in $[1,...,N_{RI}]$, being N_{RI} the number of relations of the RI table. The zero value is included in this range and it is interpreted as it does not necessary to include any relations. A value j different of zero is interpreted that the j relation of the RI is added to the rule.

  The genetic operators use on this level are two points crossover and a variation of the integer uniform mutation. The variation consists of considering that the probability of change to the zero value is 0.5 and $\frac{0.5}{N_{RI}}$ for the rest of values. The modification of this operator is included for increasing the possibility to remove irrelevant relations in the rule.

  In the initial population, all individuals begin with all the genes of this level with zero value, that is, at the beginning of the genetic process, the relations are not considered.

In this work, we use a state steady 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.

5 Experimental Studies

In this section, we will describe the relations used on the selected learning problem. We have applied the proposed algorithm to two classification problems: the first one was artificially generated and represents the kind of problems that are not easy to describe using attributed valuated rule because it need a high number of rule in the knowledge database. The goal of this example is to show the capacity of the algorithm for extracting the relations involved in the problem; the second is a real problem and the goal is to find a knowledge base that describes, in a more understandable way, the relations among the input variables.

5.1 Relations defined

In this empirical part we are only considered two relations, the crisp order relation *less than (<)* and the fuzzy relation *approximately equal to (≈)*. For the first of them, remarks that we can consider that the relation *greater than (>)* is included, since $X_i > X_j$ is equivalent to $X_j < X_i$, and only is needed to include both possibilities in the RI table.

The "approximately equal" relation can be implemented in different ways. In our case, this fuzzy relation for variables with a bounded domain is defined as:

$$\mu_{X_i \approx X_j}(x_i, x_j) = \begin{cases} (1 - |\frac{x_i - x_j}{\lambda\Delta}|)^2 & \text{if } |x_i - x_j| < \lambda\Delta \\ 0 & \text{otherwise} \end{cases}$$

where x_1 and x_2 are values of the domain of the variables, Δ represents the different between the upper and lower limits of the domain and λ is a parameter that takes value in (0,1]. In our experimentation, we consider λ=0.25.

5.2 Artificial database

For testing the behaviour of the proposed algorithm, we have generated a example database using the following set of rules:

if *($X_2 \approx X_8$)* **and** *(X_{10} >0.7)* **then** *class 1*
else if *($X_3 \approx X_4$)* **and** *(X_9 >0.7)* **then** *class 0*
else if *($X_5 \approx X_6$)* **and** *($X_6 \approx X_7$)* **then** *class 2*

All the predictive variables take values in [0,1] and we have used domains with five uniformly distributed labels for all variables.

The main problem of this database is the high number of attribute valuated rules needed for describing the system when relations are not permitted in the antecedent. In this case, we can see the relations like a way for compacting a subset of attribute valuated rules and therefore, reducing the number of rules and improving the interpretability of the knowledge base.

We have generated a database with 150 examples, 50 examples per class, with 10 predictive variables and a classification variable with three possible values. Containing 200 possible relations in the RI table, we have run the algorithm using ten hold one cross-validation. The results are shown in Table 1 where $SLAVE$ represent the basic version of the algorithm and $SLAVE^R$ represents the version with fuzzy relations.

Table 1: Results obtained with the two versions of the SLAVE learning on the artificial datatabase

	Training %	Test %	Rules
$SLAVE^R$	98.97 ± 0.6	98.64 ± 1.1	3.2 ± 0.3
$SLAVE$	79.29 ± 4.5	64.93 ± 14.8	6.4 ± 2.3

From the experimentation, we can conclude that the system is able to extract the existing relations in the example sets and this database presents problems for obtaining knowledge without to include relations in the antecedent of the rules.

5.3 Infarction database

We study the performance of the learning algorithm proposed on a real problem based in a medical domain. The infarction database is made up by 14 different variables for determining the nominal diagnostic variable. There are 2 nominal variables and the rest of them are continuous. These variables are divided into two different type of tests, morphological methods and biochemical analysis. These tests are accepted as the best markers for the postmortem diagnosis of myocardial infarction [4]. The database contains 71 examples where there are missing values and we run the algorithm using ten hold one crossvalidation. Table 2 shows the obtained results considering that in the RI table have been taken into account of the possible relations amongs the variables with the same universe of discurse.

Table 2: Results obtained by the two versions of the SLAVE algorithm on the Infarction database

	Training %	Test %	Rules
$SLAVE^R$	92.18 ± 3.1	90.40 ± 8.4	2.6 ± 0.6
$SLAVE$	96.70 ± 5.2	84.05 ± 11.2	4.8 ± 1.1

We compare this results with other well-known learning algorithms. The learning algorithms choiced are *C4.5* that represents the knowledge learned by a decision tree, *CN2* that represents the knowledge by a set of classical DNF rules and *LVQ* that is a adaptative classification method based on the Kohonen self-organizated maps. Table 3 shows the results obtained with these algorithms. Furthermore, the CN2 algorithm obtains on this databases an average number of rule of 5.4.

Table 3: Predictive performance the others learning algorithm on Infarction database

C4.5	CN2	LVQ
81.4 %	76.16 %	56.41 %

With these results, we can see that the inclusion of relations improve the predictive capacity of the generated knowledge base reducing the number of rules. Furthermore, this knowledge is simpler and more understandable for a human. Now, we show an example of rule base obtained by the proposed algorithm:

Attributes	Diagnosys
K/Na(A) is $\{$Low$\}$ & K/Na(C) is $\{$Low$\}$	YES
ApoA is $\{$Low,Medium$\}$ & HDL $<$ LDL	NO
K/Na(B) $<$ K/Na(D) &K/Na(D) $<$ K/Na(A)	YES

In this example appears the attributes (such as K/Na attributes, HDL and LDL) and relations between variables (such as K/Na(B)$<$ K/Na(D) and K/Na(D)$<$K/Na(A)) that are very important for determining the infarction diagnosys [4]. So, we can consider that the knowledge learned by the proposed algorithm can be better interpreted for a medical expert in this field.

References

[1] L. Castillo, A. González, R. Pérez (2001). Including a simplicity criterion in the selection of the best rule in a genetic fuzzy learning algorithm, *Fuzzy Sets and Systems*, volumen 120, pages 309-321.

[2] A. González, R. Pérez (1998). Completeness and consistency conditions for learning fuzzy rules, *Fuzzy Sets and Systems*, volumen 96, pages 37-51.

[3] A. González, R. Pérez (2001). Selection of Relevant Features in a Fuzzy Genetic Learning Algorithm, *IEEE Transactions on System, Man, and Cybernetics Part B*, volumen 31(3), pages 417-425.

[4] H.P. Hougen, E. Villanueva, A. Valenzuela (1992), Sudden cardiac death: a comparative study of morphological, histochemical and biochemical methods, Forensic Science Internacional 52, pages 161-169.

[5] E.H. Mamdani, S. Assilian (1975). An experiment in linguistic synthesis with fuzzy logic controller. *International Journal of Man-Machine Studies*, volumen 7, pages 1-13.

[6] T. Mitchell (1997). *Machine Learning* , Ed. MacGraw-Hill.

[7] T. Takagi, M. Sugeno (1985). Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics*, volumen 15, pages 116-132.

[8] R.R. Yager (1991). The representation of fuzzy relational production rules, *Journal od Applied Intelligence*, volumen 1, pages 35-42.

[9] R.R. Yager, D. P. Filev (1996). Relational partitioning of fuzzy rules, *Fuzzy Sets and Systems*, volumen 80, pages 57-69.