

Genetic Learning of Fuzzy Parameters in Predictive and Decision Support Modelling

Àngela Nebot

*Dept. Llenguatges i Sistemes Informàtics,
Technical University of Catalonia (UPC),
Jordi Girona 1-3, 08034 Barcelona, Spain
angela@lsi.upc.edu
www.upc.edu*

Francisco Mugica

*Dept. Llenguatges i Sistemes Informàtics,
Technical University of Catalonia (UPC),
Jordi Girona 1-3, 08034 Barcelona, Spain
fmugica@lsi.upc.edu
www.upc.edu*

Félix Castro

*Centro de Investigación en Tecnologías de la Información y Sistemas (CITIS),
Universidad Autónoma del Estado de Hidalgo,
Hidalgo, Mexico.
fcastro@lsi.upc.edu
www.uaeh.edu.mx/investigacion/sistemas*

Jesús Acosta

*Dept. Instrumentación Industrial,
Instituto Universitario de Tecnología Alonso Gamero (IUTAG),
4101-Coro, Venezuela.
jesusacostav@hotmail.com
www.iutag.tec.ve*

Received 15 October 2010

Accepted 1 June 2011

Abstract

In this research a genetic fuzzy system (GFS) is proposed that performs discretization parameter learning in the context of the Fuzzy Inductive Reasoning (FIR) methodology and the Linguistic Rule FIR (LR-FIR) algorithm. The main goal of the GFS is to take advantage of the potentialities of GAs to learn the fuzzification parameters of the FIR and LR-FIR approaches in order to obtain reliable and useful predictive (FIR) models and decision support (LR-FIR) models. The GFS is evaluated in an e-learning context.

Keywords: Genetic fuzzy systems, fuzzy inductive reasoning, predictive models, decision support models, e-learning.

1. Introduction

Initially, e-learning was presented as the best solution to cover the needs and requirements of remote students, but also as a helping tool in the teaching-learning process, reinforcing or replacing face-to-face education. However, the undisguised truth is that many real projects have failed, or at least they have performed below expectations, due to the fact that a huge amount of time is required just in the process of providing feedback to the virtual learners, resulting in an increasing demand of teachers and, therefore, of the educational costs.

On top of that, one of the most difficult and time consuming activities for teachers in distance education courses is the evaluation process, due to the fact that the reviewing process in this kind of courses is better accomplished through collaborative resources such as e-mail, discussion forums, chats, etc. As a result, this evaluation usually has to be done according to a large number of factors, whose influence in the final mark is not always well defined and/or understood. Therefore, it would be helpful in order to reduce the intrinsic system evaluation dimensionality to identify the factors that are highly relevant for the students' evaluation. This will help teachers to provide feedback to students in function of their learning behaviour in real time.

The use of data mining methods to extract knowledge from the e-learning system available information can be an adequate approach to follow, so that the obtained knowledge can be used to fit the educational proposal better to the students' needs and requirements.

Several research projects have dealt with the integration of data mining methods focusing on e-learning systems improvement. For a deeper inside into these projects the authors recommend Refs. 1-3, where an extensive and profound analysis of different learning platforms is performed, including LON-CAPA,⁴ AHA1,⁵ ALFANET,⁶ etc. Commonly, the existing platforms perform students' classification (using supervised neural networks, decision trees, fuzzy methods, association rules, etc.), and/or students' clustering (using Kohonen's self-organizing maps, EM, etc.), however students' performance has not been addressed from the prediction of their learning behaviour point of view, i.e. predictive models are not included in the platforms.

The authors are currently developing a platform, based on the Fuzzy Inductive Reasoning (FIR) methodology that includes as part of its skills the identification of students' learning behaviour models that allow both, students and teachers, to know the future performance of the student based on the current learning behaviour, allowing teachers to give feedback to those students that need it. The platform also provides functionalities to determine the most relevant features involved in the evaluation process. Finally, it allows extracting linguistic rules that are easily understandable by experts in an educative domain and that help them to understand students' learning behaviour. This is performed by means of the Linguistic Rules extraction algorithm (LR-FIR) that is an extension of the FIR methodology.

In this paper a genetic fuzzy system (GFS) is presented that allows enhancing the performance of the above mentioned functionalities. The main goal of the GFS presented in this research is to take advantage of the potentialities of GAs to learn the fuzzification parameters of the FIR methodology, i.e. the membership functions associated to each variable. Due to the fact that it is a methodology based on fuzzy logic, FIR modelling and prediction performance is influenced by these discretization parameters.

There exists some studies on evolutionary fuzzy rule based systems (EFRBS) focused on prediction.⁷⁻¹¹ However, EFRBS are mainly focused on control¹²⁻¹⁴ and classification.^{15,16} Research developed in Refs. 7 to 11 are related to the approach proposed in this paper. In Ref. 7 the authors propose an interesting index, that is the aggregation of three metrics, that preserves the original meanings of the membership functions as much as possible while tuning their associated parameters. Ref. 8 and 9 propose a new post-processing method for the lateral and amplitude tuning of membership functions. Ref. 10 presents a fuzzy expert system trained by different genetic algorithms. In Ref. 11 a multi-objective evolutionary algorithm based on embedded genetic data base learning, is proposed, that allows the fast learning of simple and quite accurate linguistic models. These research were tested in a large set of different application areas such are weather forecasting, plastic strength, stock prices, maintenance cost of an electrical network, wind speed at a wind energy conversion system, etc., but no one is related to e-learning.

The remaining of the paper is organized as follows: section 2 presents the basics of the FIR methodology and the LR-FIR extension. The GFS developed is presented in section 3. A description of the experiments performed when the proposed GFS is applied to the CECTE didactic planning course is presented and discussed in section 4. Finally, section 5 wraps up the paper with some conclusions.

2. The Fuzzy Inductive Reasoning Methodology

The conceptualization of the Fuzzy Inductive Reasoning (FIR) methodology arises from the General System Problem Solving approach (GPS) proposed by Klir.¹⁷ This methodology of modeling and qualitative simulation is based on systems behavior rather than on structural knowledge. It is able to obtain good qualitative relations between the variables that compose the system and to infer the future behavior of that system. It has the ability to describe systems that cannot easily be described by classical mathematics (e.g. differential equations), i.e. systems for which the underlying physical laws are not well understood.

2.1 Introduction to the FIR methodology

FIR is composed of four main processes, namely: fuzzification, qualitative model identification, fuzzy forecasting and defuzzification.

The fuzzification process converts quantitative data stemming from the system into fuzzy data, i.e. qualitative triples. The qualitative model identification process is the responsible for finding causal and temporal relations between variables and therefore of obtaining the model that best represents the system.

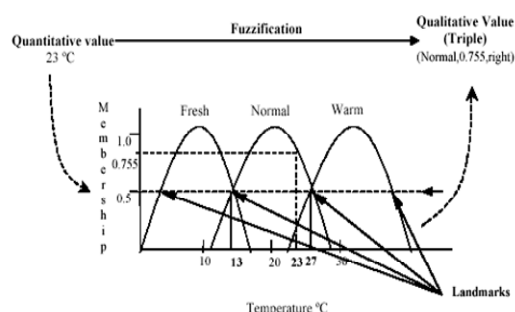


Fig. 1. FIR fuzzification process of ambient temperature variable.

Once the FIR model is available, the prediction

system can take place using the FIR inference engine. This process is called fuzzy forecast. FIR inference engine is a specialization of the k-nearest neighbor rule, commonly used in the pattern recognition field. Defuzzification is the inverse process of fuzzification. It allows to convert the qualitative predicted output into quantitative values that can then be used as input to an external quantitative model.

2.1.1 Fuzzification

Figure 1 illustrates the process of fuzzification by means of an example. As mentioned earlier, a quantitative value is fuzzified into a qualitative triple. The first element of the triple is the class value, the second element is the fuzzy membership value, and the third element is the side value. The side value indicates whether the qualitative value is to the left or to the right of the peak value of the associated membership function.

The side value, that is not commonly used in fuzzy logic, is responsible for preserving, in the qualitative triple, the complete knowledge contained in the original quantitative value.

In Figure 1 a temperature of 23 degrees centigrade would hence be fuzzified into the class *normal* with a side value *right* and a fuzzy membership value of 0.755.

The point where two neighboring classes match with a membership value of 0.5 is named *landmark*. In the example of Figure 1 the membership function of the class *normal* is defined by landmarks {13,27}, being this pair the temperature values that specify the limits between the class *normal* and its adjacent classes, *fresh* and *warm*, respectively.

The result of the fuzzification process are three matrices of identical size named qualitative data matrices, one containing the class values, the second storing the membership information, and the third recording the side values. Each column represents one of the observed variables and each row denotes one time point, i.e. one recording of all variables or one recorded state.

2.1.2. Qualitative model identification

A FIR model is composed by a structure, called mask, and a pattern rule base, named behaviour matrix. A mask denotes a dynamic relationship among qualitative variables. An example of a mask is presented in Table 1.

Each negative element in the mask is called a m-input (mask input). It denotes a causal relation with the output, i.e. it influences the output up to a certain degree. The enumeration of the m-inputs is immaterial and has no relevance. The single positive value denotes the output.

Table 1. Example of a FIR mask.

	u_1	u_2	u_3	u_4	y_1
$t-2\delta t$	-1	0	0	-2	0
$t-\delta t$	0	0	0	0	-3
t	0	-4	0	0	+1

The mask of Table 1 contains four m-inputs. In this example, the first and second m-inputs, i_1 and i_2 , correspond to the input variables u_1 and u_4 two sampling intervals back, $u_1(t-2\delta t)$ and $u_4(t-2\delta t)$, whereas the third m-input, i_3 , refers to the output variable y_1 one sampling interval into the past, $y_1(t-\delta t)$, etc. It could be expressed as described in Eq. (1).

$$y_1(t) = \tilde{f}(u_1(t-2\delta t), u_4(t-2\delta t), y_1(t-\delta t), u_2(t)) \quad (1)$$

where \tilde{f} denotes a qualitative relationship. How is a mask found that, within the framework of all allowable masks, represents the most deterministic state transition matrix, i.e., optimizes the predictiveness of the model? In FIR, the concept of a mask candidate matrix is introduced. A mask candidate matrix is an ensemble of all possible masks from which the best is chosen by either a mechanism of exhaustive search of exponential complexity or by one of various suboptimal search strategies of polynomial complexity as described in Ref. 18. The mask candidate matrix contains -1 elements where the mask has a potential m-input, a +1 element where the mask has its m-output, and 0 elements to denote forbidden connections. Thus, a good mask candidate matrix to start the search for the best mask shown in Table 1 might be the one shown in Table 2.

Table 2. Example of a FIR candidate matrix.

	u_1	u_2	u_3	u_4	y_1
$t-2\delta t$	-1	-1	-1	-1	-1
$t-\delta t$	-1	-1	-1	-1	-1
t	-1	-1	-1	-1	+1

Each of the possible masks is compared to the others with respect to its potential merit, i.e., the degree of determinism associated with the state transition matrix

constructed from it. The optimality of the mask is evaluated with respect to the maximization of its forecasting power. The Shannon entropy measure is used to determine the uncertainty associated with forecasting a particular output state given any legal input state. The Shannon entropy relative to one input state is calculated from Eq. (2):

$$H_i = \sum_{\forall o} p(o|i) \cdot \log_2 p(o|i) \quad (2)$$

where $p(o|i)$ is the conditional probability of a certain m-output state o to occur, given that the m-input state i has already occurred. The term probability is meant in a statistical rather than in a true probabilistic sense. It denotes the quotient of the observed frequency of a particular state divided by the highest possible frequency of that state. The overall entropy of the mask is then computed as the sum given in Eq. (3).

$$H_m = - \sum_{\forall i} p(i) \cdot H_i \quad (3)$$

where $p(i)$ is the probability of that input state to occur. The highest possible entropy H_{max} is obtained when all probabilities are equal, and a zero entropy is encountered for relationships that are totally deterministic. A normalized overall entropy reduction H_r is defined as described in Eq. (4).

$$H_r = 1.0 - \left(\frac{H_m}{H_{max}} \right) \quad (4)$$

H_r is obviously a real-valued number in the range between 0.0 and 1.0, where high values usually indicate an improved forecasting power. The masks with highest entropy reduction values generate forecasts with the smallest amounts of uncertainty.

One problem still remains. The size of the pattern rule base increases as the complexity of the mask grows, and consequently, the number of legal states of the model grows fast. Since the total number of observed data records remains constant, the frequency of observation of each state shrinks rapidly, and so does the predictiveness of the model. The entropy reduction measure does not account for this problem. With increasing complexity, H_r simply keeps growing. Very soon, a situation is encountered where every state that has ever been observed has been observed precisely once. This obviously leads to a totally deterministic state transition matrix, and H_r assumes a value of 1.0.

Yet the predictiveness of the model will be dismal, since in all likelihood already the next predicted state has never before been observed, and that means the end of forecasting. Thus, this consideration must be included in the overall quality measure.

From a statistical point of view, every state should be observed at least five times.¹⁹ Therefore, an observation ratio, O_r , is introduced as an additional contributor to the overall quality measure, as described in Eq. (5),

$$O_r = \frac{5 \cdot n_{5x} + 4 \cdot n_{4x} + 3 \cdot n_{3x} + 2 \cdot n_{2x} + n_{1x}}{5 \cdot n_{leg}} \quad (5)$$

where: n_{leg} is the number of legal m-input states, n_{1x} is the number of m-input states observed only once, n_{2x} is the number of m-input states observed twice, etc.

If every legal m-input state has been observed at least five times, O_r is equal to 1.0. If no m-input state has been observed at all (no data are available), O_r is equal to 0.0. Thus, O_r can also be used as a quality measure. The overall quality of a mask, Q , is then defined as the product of its uncertainty reduction measure, H_r , and its observation ratio, O_r , as shown in Eq. (6).

$$Q = H_r \cdot O_r \quad (6)$$

The optimal mask is the mask with the largest Q value. Let us now address the second issue. How is the pattern rule based obtained from the mask? This process is illustrated in Figure 2. The mask can be used to ‘flatten’ dynamic relationships into pseudo-static relationships. The left side of Figure 2 shows an excerpt of the qualitative data matrix that stores the class values.

It shows the numerical rather than the symbolic class values. In the example shown in Figure 2, all the variables were discretized into three classes, except variable y_1 , that was discretized into two classes. The dashed box symbolizes the mask that is shifted downwards along the class value matrix. The round shaded ‘holes’ in the mask denote the positions of the m-inputs, whereas the square shaded ‘hole’ indicates the position of the m-output. The class values are read out from the class value matrix through the ‘holes’ of the mask, and are placed next to each other in the behavior matrix that is shown on the right side of Figure 2.

Here, each row represents one position of the mask along the class value matrix. It is lined up with the

bottom row of the mask. Each row of the behavior matrix represents one pseudo-static qualitative state or qualitative rule (also called pattern rule). For example, the shaded rule of Figure 2 can be read as follows: ‘If all the m-inputs (i_1, i_2, i_3, i_4) have a value of 2 (corresponding to *medium*) then the m-output, O_1 , assumes a value of 1 (corresponding to *high*)’.

The qualitative rules can be invoked during qualitative simulation to predict new qualitative outputs. Clearly, these rules can be written in any order, i.e., the sequencing of the rows of the behavior matrix has become irrelevant. They can be sorted alphanumerically. The sorted behavior matrix is called state transition matrix.

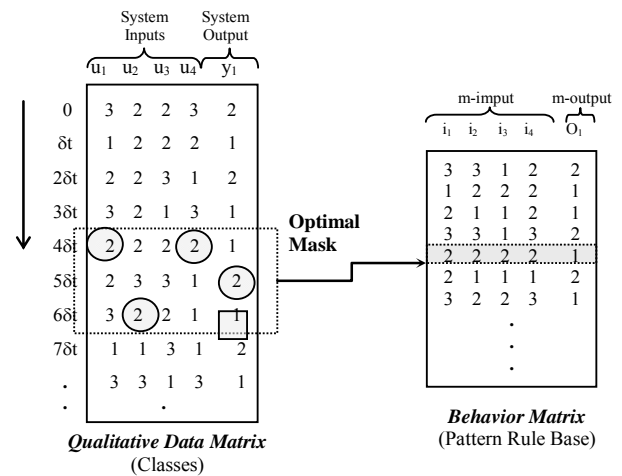


Fig. 2. Process of flattening dynamic relationships into pseudo-static relationships using a mask.

2.1.3. Fuzzy forecasting

Once the best model is obtained by means of computing the quality measure presented above, future output states can be predicted using the inference engine that is at the heart of the qualitative simulation module inside FIR. This process is described in Figure 3.

Using the five-nearest-neighbors (5NN) fuzzy inferencing algorithm the membership and side functions of the new input are compared with those of all previous recordings of the same qualitative input. The input with the most similar membership and side functions is identified. For this purpose, a normalized defuzzification is computed for every input variable of the new input set, using Eq. (7).

$$pos_i = class_i + side_i \cdot (1 - Memb_i) \quad (7)$$

The pos_i values are stored in a vector, **pos**. The index i represents the i^{th} input variable in the input state of the current observation. $Memb_i$ is the membership value, and $class_i$ and $side_i$ are the numeric class and side values associated to those inputs, respectively.

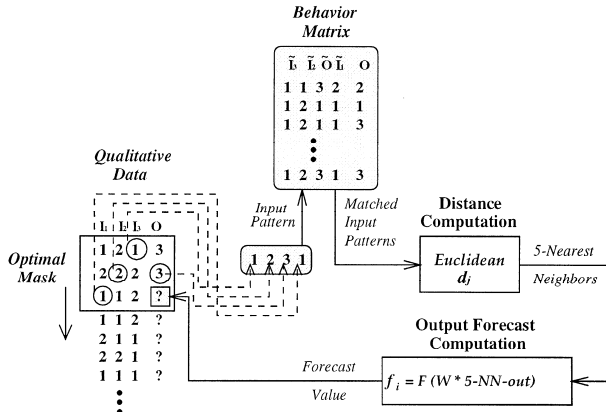


Fig. 3. Fuzzy forecasting FIR Process.

The position value, pos_i , can be interpreted as a normalized defuzzification of the i^{th} input variable. Irrespective of the original values of the input variable, pos_i assumes values in the range (1.0-1.5) for the lowest class, (1.5-2.5) for the next higher class, etc. The defuzzification is repeated for all previous recordings of the same input state, as shown in Eq. (8).

$$pos_{ij} = class_{ij} + side_{ij} \cdot (1 - Memb_{ij}) \quad (8)$$

where the index j denotes the j^{th} previous observation of the same input state. Also the pos_{ij} values are stored in a vector, **pos_j**. Then, the L_2 norms of the difference between the **pos** vector of the new input state and the **pos_j** vectors of all previous recordings of the same input state are computed following Eq. (9),

$$dis_j = \sqrt{\sum_{i=1}^N (pos_i - pos_{ij})^2} \quad (9)$$

where N is the number of m -inputs. Finally, the previous recording with the smallest L_2 norm is identified. The class and side values of the output state associated with this input state are then used as forecasts for the class

and side values of the new output state. Forecasting of the new membership function is done a little differently. Here, the five previous recordings with the smallest L_2 norms are used (if at least five such recordings are found in the input/output matrix), and a distance weighted average of their fuzzy membership functions is computed and used as the forecast for the fuzzy membership function of the current state.

2.1.4. Defuzzification

The defuzzification engine of the FIR methodology is responsible for converting each qualitative predicted output triple back to a quantitative output value. It is the inverse operation of the previously described fuzzification engine.

For a deeper and more detailed insight into the FIR methodology, the reader is referred to Ref. 20.

2.2 Linguistic rules extraction from FIR models: LR-FIR

The LR-FIR algorithm was developed with the goal to be a useful tool for decision makers. With this purpose in mind the ultimate goal of LR-FIR is to obtain interpretable, realistic and efficient behavioural rules, describing complex systems.

Previous efforts have been made in this direction giving major attention to the interpretability associated to the fuzzy rule-based systems, mainly in linguistic ones. Some examples of these works are Ref. 21 to 23.

LR-FIR performs an iterative process that compact the pattern rule base, i.e. input/output relationships, obtained by FIR. In order to obtain a set of rules congruent with the model previously identified by FIR, LR-FIR is based on its initial discretization (performed in the *fuzzification* process). Figure 4 shows, in a schematic way, the main phases of the LR-FIR algorithm. In the rule extraction process, the mask features and the pattern rule base obtained are used. The algorithm can be summarized as a set of ordered steps:

2.2.1. Basic compaction

The main goal of this step is to transform the pattern rule base, R , into a reduced set of rules, R' . Notice that R , is usually very large being almost as large as the number of training data available. Consist on an

iterative step that evaluates one at a time, all the rules and each of their premises in a pattern rule base. The pattern rule base, R , is compacted on the basis of the “knowledge” obtained by FIR. A specific subset of rules, R_c , can be transformed in the form of a compacted rule r_c , when all premises P but one (P_a), as well as the consequence C share the same values. Premises, in this context, represent the input features, whereas consequence is the output feature in a rule. If the subset contains all legal values LV_a of P_a , all these rules can be replaced by a single rule, r_c , that has a value of -1 in the premise P_a . A -1 value in a variable means that this variable is not relevant in the rule and, therefore, it is not considered. Let see this concept by means of an example. Assume that we have the subset R_c of three rules presented in Table 3, and that the algorithm is working with the third premise, i.e. $v_j = 3$.

Table 3: Example 1: Set of candidate rules to be compacted.

Rule	Premises	Consequent
1	1 2 2	1
2	1 2 1	1
3	1 2 3	1

As the third premise is discretized into only 3 classes and all of them are represented in R_c , it can be concluded that this variable is not relevant in this set of rules and therefore R_c can be compacted to the rule $r_c = 1 \ 2 \ -1 \ 1$.

Notice that the algorithm performs two iterations, an external that deals with each one of the premises and an internal that deals with each rule of the pattern rule base. Several considerations to this algorithm should be mentioned here.

- When more than one -1 value is present in a compacted rule r_c , it is compulsory to evaluate the existence of conflicts by expanding all the premises to all their legal values LV_a , and comparing the resultant rules with the original pattern rules R . If conflicts exist, the compacted rule r_c is rejected, and otherwise accepted. In the latter case, the previous subset, R_c is replaced by the compacted one r_c . Conflicts occur when one or more extended rules have the same values in all its premises but different values in the consequence.
- When a -1 value appears in any of the variables that are not the one evaluated at this moment (v_j), the -1 takes in the class values of the other rules in that

premise. Assume that we have the subset R_c of three rules presented in Table 4, and that the algorithm is working with the third premise, i.e. $v_j = 3$.

Table 4: Example 2: Set of candidate rules to be compacted when a premise has a -1 value.

Rule	Premises	Consequent
1	1 2 1	1
2	-1 2 2	1
3	1 2 3	1

In this case, the -1 value of the first premise of rule #2 takes in class 1 of rules #1 and #3, and therefore, all three rules are compacted into the rule: -1 2 -1 1. Notice that in this case, the algorithm will check for conflicts before the compacted rule is accepted.

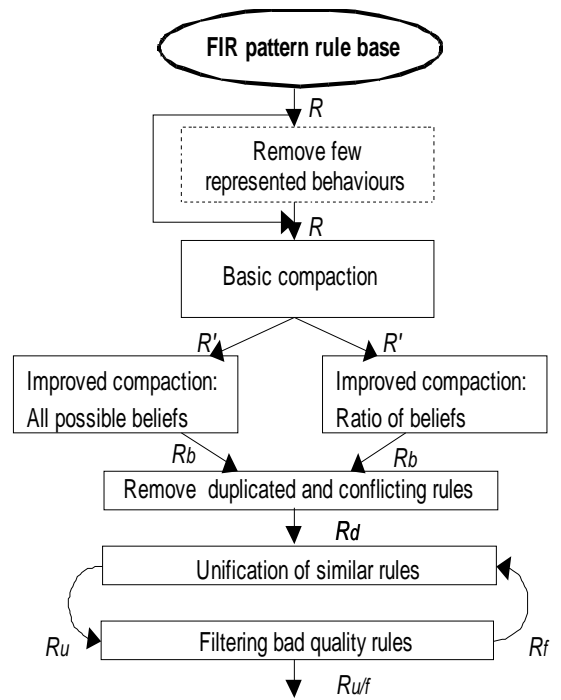


Fig. 4. Main steps of the LR-FIR algorithm.

2.2.2. Improved compaction

Whereas the previous step only structures the available knowledge and represents it in a more compact form, the improved compaction step extends the knowledge base R to cases that have not been previously used to build the model: R_b . Thus, whereas the step 1 leads to a

compacted data base that only contains knowledge; the enhanced algorithm contains undisputed knowledge and uncontested belief. Two options are studied: *all possible beliefs* and *ratio of beliefs*. In the first option, *improved compaction with all possible beliefs*, using the compacted rule base R' obtained in step 1, all input features P (premises) are visited once more in all the rules r that have nonnegative vales (not compacted), and their values are replaced by -1. An expansion to all possible full sets of rules Xr and their comparison with the original rules R are carried out. If no conflicts, Cf , are found, the compacted rule, r_c , is accepted, and otherwise, rejected.

The second option, *improved compaction with a ratio of beliefs*, is an extension of the basic compaction, where a consistent minimal ratio, MR , of the legal values LVa should be present in the candidate subset R_c , in order to compact it in the form of a single rule r_c . This latter option seems more suitable because, although a consistent ratio was used to compact R_c in a single rule r_c , the assumed beliefs are minimal and do not compromise the model previously identified by FIR. Instead, in option 1, beliefs are assumed to be consistent with the original rules; nevertheless, this could compromise the agreement with model identified, especially when the training data are poor and do not describe well all possible behaviours.

2.2.3 Remove duplicated and conflicting rules

In this step all duplicated rules are removed. It can be found, also, in R_b , ambiguous or conflicting rules, i.e. rules with the same values in all its premises but different value in the consequent. In this case, the rules that are involved in conflicts are analyzed, and those with lower quality, Q_r , are eliminated. The quality of a rule is assessed using the well known specificity and sensitivity measures that are standard metrics often applied in the machine learning field. Specificity and sensitivity measures are in the range $[0..1]$. High quality means high values (closer to 1) of specificity and sensitivity measures; therefore the rule with highest Q_r remains and the other conflicting rules are eliminated. In order to maintain a robust and consistent set of rules, those conflicting rules sharing contiguous input space (adjacent classes) in the consequent are not removed since these rules should be unified in the next step of the algorithm.

2.2.4. Rule unification

The rule unification step can be performed before or after rules filtering. If it is done before, rules with low quality are usually unified with rules of better quality deriving, more often than not, to a set of rules that preserve as much as possible the full behaviour of the system but with a large level of generalization, not really useful for decision support systems. Contrarily, if it is done after, low quality rules that have conflicts with better quality rules are eliminated and therefore, the resulting set of rules explain in a more synthesized and clear way the more often behaviours of the system. Although, obviously, some information is lost, the set of rules obtained is usually more suitable for decision support. The user should decide in which order these two steps are executed depending on the goals of his/her research.

Rule unification is an iterative process that evaluates, one at a time, each rule with respect to the remaining ones to find similar candidate rules to be unified in a single one R_u . This is carried out in two phases. In the first phase the rules that share, in a same variable (premise or consequent), contiguous input spaces and the same values in the remaining ones should be unified in a unique rule R_u . In order to maintain a consistent set of rules and do not compromise the previous steps a subset of rules, R_d , should not be unified when the contiguous input space of the candidate rules R_d , cover all the legal values of that variable. This condition is included because when this happens a conflict surely exist, otherwise these candidate rules R_d , would have been compacted in the basic or improved compaction steps.

There are four options to perform this step: *wise*: a subset of rules R_d , is unified in a unique rule R_u , if and only if the quality Q_r , of the unified rule R_u , is higher than the best quality of the candidate rules; *blind*: a subset of candidate rules R_d , is unified without verifying the quality Q_r of the unified rule R_u .

These two alternatives can be combined with *repetitions* and *without repetitions* options. In the first one, a rule can be unified with several rules, i.e. whenever possible, whereas in the second one, a rule can be unified only once. Optionally, those rules not unified in the first stage are evaluated with the goal to discover new unifications with the already unified rules. The unification is performed only in the consequent

value. In this step the default option is *without repetitions*, i.e. a rule can be unified only once.

2.2.5 Rule filtering

The obtained set of rules R_d or R_u is evaluated using the Sensitivity and Specificity metrics. These metrics allow an objective and realistic assessment of the resulting rules. A parameter, chosen by the user, determines the minimum quality value to be accepted. The rules that have associated lower qualities for, at least, one of the metrics are eliminated. These standard metrics that assess the quality of the obtained rules are explained in section 4. For a deeper insight into LR-FIR, the reader is referred to Ref. 24.

3. The GFS Proposed (GA+FIR+LR-FIR)

The GFS proposed in this research is presented in Fig. 5. A genetic algorithm (GA)^{25,26} is used to obtain the membership functions associated to the classes for each system variable. These parameters are used in the fuzzification process of FIR methodology, and have direct influence in the identification process of the FIR predictive model. Depending on the membership functions defined, the model structure (mask) obtained changes and the pattern rules derived from it change as well.

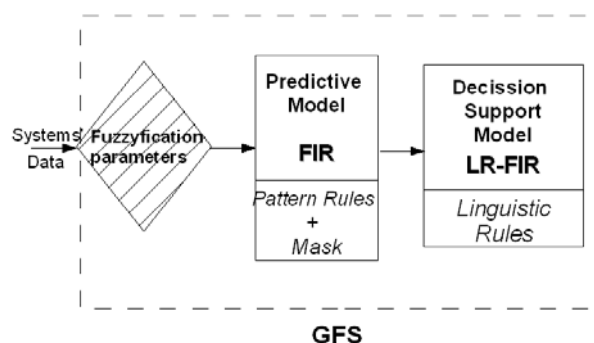


Fig. 5. Scheme of the GFS.

The optimization performed by the GA is based on the prediction power of the FIR model obtained. The pattern rules that compose (together with the mask) the FIR model are then used to derive a set of linguistic rules, by means of the LR-FIR algorithm. The set of linguistic rules constitutes a decision support model that helps teachers and students to understand student's learning behaviour. Due to the fact that the LR-FIR algorithm

has as input the pattern rules of the predictive model, variations in the fuzzification parameters affect directly the decision support model, i.e. the set of linguistic rules. Therefore, it is of great importance to determine in an automatic and reliable way good fuzzification parameters in order to maximize the powerfulness of both predictive and decision support models.

3.1. Determination of the membership functions

3.1.1. Genetic representation

The genetic representation chosen takes into account the number of samples registered for each variable. A specific variable is represented by the proportion of data samples that contains each class, codified in the range [0..1]. An example of chromosome representation for a unique variable that has 4 classes could be (0.2,0.3,0.4,0.1), meaning that the membership function of the first class contains the 20% of the data samples available for this variable, and the second, third and fourth membership functions contain 30%, 40% and 10% of the data records, respectively. Obviously, the sum of the proportions for each class must be 1. Therefore, if we denote by D_{ij} the data proportion of the variable i and class j , and C_{2i} the information of the data proportion for all the classes of the variable i , the membership representation, C_2 , for a system of N variables (including inputs and outputs), is defined by: $C_2 = (C_{21}, C_{22}, \dots, C_{2N})$, where $C_{2i} = (D_{i1} \dots D_{iE_i})$. Note that each time the distribution of the landmarks changes due to the action of the genetic operators, it is mandatory to re-compute the proportions of the new distribution. A clear advantage of this representation is the facility to compute the landmarks from it.

3.1.2. Fitness or objective function

In order to evaluate a chromosome, the following steps are considered:

- Decode the information of the chromosome, building the associated fuzzy partition in the FIR structures.
- Execute the *qualitative model identification* process of FIR methodology with the training data set, using the partition built in the previous step. Therefore, the mask associated to that partition with the highest quality measure is obtained.
- Compute an objective function. The cost function is

defined as the prediction error of a portion of the training data set, called validation data set. The normalized mean square error in percentage (MSE), given in Eq. (10), is used for this purpose,

$$MSE = \frac{E[(y(t) - \hat{y}(t))^2]}{y_{var}} \cdot 100\% \quad (10)$$

where $\hat{y}(t)$ is the predicted output, $y(t)$ the system output and y_{var} denotes the variance of $y(t)$. The idea is to use part of the training data set to identify the model and the rest of the data set (validation set) to evaluate the prediction performance of that model. It is important to remember that the FIR model is composed of the optimal mask and the pattern rule base (behaviour matrix). Therefore, both must be generated in the evaluation process of a certain fuzzy partition for this cost function. The size of the validation data set used for cost function evaluation purposes is defined with respect to the size of the whole training data set.

3.1.3. Genetic operators and parameters

In this work, the same genetic operators presented in Ref. 27 are used. The genetic parameters are described in section 4. The initial population is composed by an individual distributed using the equal frequency partition (EFP) method, and the rest of the chromosomes have their values chosen randomly. No repeated chromosomes are allowed. As known, the EFP algorithm defines the classes in such a way that the same number of data records is included in each membership function.

4. E-Learning Models

The Centre of Studies in Communication and Educational Technologies (CECTE: Spanish acronym) is a partially virtual campus, offering postgraduate courses and continuous education (graduate, workshops and specific courses) to Latin-American students. The CECTE is part of the international Latin-American Institute of Educative Communication (ILCE: Spanish acronym), whose main goal is to offer postgraduate courses.

The most demanded CECTE courses follow a hybrid, semi-presential model, in which students take courses online (WCECTE) but also attend weekly TV sessions

through the National System of Educative Television (EDUSAT). Through WCECTE, students can access the course materials and communicate and interact with each other through an e-mail system and a discussion forum. The environment also includes an agenda, a news system, virtual classrooms, a digital library, interactive tutorials, and other related tools.

In the CECTE educational model the tutor is a very relevant actor, as he or she interacts directly with students, assigning learning activities, answering doubts, opening topics in discussion forums, evaluating the activities performed by learners, and verifying that the teaching-learning process be adequate, taking advantage of all the tools provided by WCECTE.

In this section the *didactic planning* CECTE course is studied. The obtained results were evaluated and validated by educative experts of the CECTE.

For the experiments in this study, a set of 700 students, enrolled in the “Didactic Planning” graduate course, was selected. The course is addressed to second term high school teachers offering specialized subjects, namely Mathematics, Chemistry, Mexican History, Computer Science, English, as well as Reading and Writing, Ethics and Values. The students are meant to perform a set of activities throughout the course with the main purpose of learning new methods and strategies for planning the classes that they teach. This is the reason why these activities are centred on the so-called “class plan”.

A class plan is a document where a set of strategies are suggested to develop a teaching-learning session, taking into account different factors that appear in the educational process, such as students’ characteristics, teaching style, teachers’ experience, etc. The data features available for this study are detailed in Table 5.

In the didactic planning course, two novel evaluation topics, not often used in e-learning environments, were incorporated: co-evaluation and experience report. In co-evaluation, the advisor grades how well the student evaluates the class plans of his/her course mates. The experience report is a student description of his/her perception of the course. It can be viewed as a self-evaluation of the student’s own learning process.

The aim of this study is to analyze the usefulness of the GFS proposed with two main focuses: FIR models that are capable of predicting student’s performance and LR-FIR models that help to understand student’s learning behaviour patterns.

In the didactic planning course, two novel evaluation topics, not often used in e-learning environments, were incorporated: co-evaluation and experience report. In co-evaluation, the advisor grades how well the student evaluates the class plans of his/her course mates. The experience report is a student description of his/her perception of the course. It can be viewed as a self-evaluation of the student's own learning process.

The aim of this study is to analyze the usefulness of the GFS proposed with two main focuses: FIR models that are capable of predicting student's performance and LR-FIR models that help to understand student's learning behaviour patterns.

Table 5. Data features collected for the didactic planning course.

Feature	Description
ACT	Percentage of the activities performed by the student with respect to the total activities of the course.
ASS	Percentage of student's session assistance with respect to the total number of sessions of the course.
MAIL	Average mark obtained by the student in the activities sent by e-mail.
COEV	Average mark of the co-evaluation performed by the student of the class plan of other students.
F	Average mark of the student's forum participation (referring to topics related to the course).
FCP	Average mark of the forum class plan (referring only to topics related to the class plan exclusively).
FC	Average mark obtained by the student in his/her final class plan.
IC	Average mark obtained by the student in his/her initial class plan.
ER	Average mark obtained by the student in the experience report. In this report the student evaluates his/her learning process and describes the main concepts learned.
BR	Average mark of the work (activities) performed in the branch.
MARK	Final mark obtained by the student in the course.

4.1. Previous work

In Ref. 24 FIR and LR-FIR models of the Didactic Planning course were identified with the main goal to provide valuable knowledge to teachers about the course performance and student's learning behaviour understanding. To this end, the FIR platform and the LR-FIR algorithm used the discretization parameters defined by the experts, i.e. course teachers. These parameters are described in Table 6. ACT, ASS, IC and BR variables were discretized into two classes whereas the rest of the variables were discretized into 3 classes. C1, C2 and C3 refer to classes 1, 2 and 3 of that

variable, respectively. As has been explained in section 2, the landmarks define the membership functions.

In this research 7-fold cross-validation was used. Each test set was composed of 100 samples whereas the training sets contained 600 samples. The optimal mask obtained by the FIR model identification process is presented in Table 7. Notice that in this case the mask has only one row meaning that no temporal relation is allowed. This is due to the fact that we are dealing with students, i.e. each data sample contains information of a specific student.

The best model identified for the didactic planning course includes the average marks of the co-evaluation (COEV), the initial class plan (IC), and the experience report (ER) features as the most relevant features to predict the final mark of the course (MARK) for each student.

Table 6. Fuzzification parameters defined by experts.

Feature	Landmarks	Granularity
ACT	C1:[24-42]; C2:(42-66]	2
ASS	C1:[13-55]; C2:(55-100]	2
MAIL, COEV, F, FCP, FC	C1:[0-4.9]; C2:(4.9-7.9]; C3:(7.9-10]	3
IC	C1:[0-5.1]; C2:(5.1-10]	2
ER	C1:[0-6.9]; C2:(6.9-8.9]; C3:(8.9-10]	3
BR	C1:[0-8.1]; C2:(8.1-10]	2
MARK	C1:[0-4.9]; C2:(4.9-7.9]; C3:(7.9-10]	3

An interesting result is the selection of the co-evaluation variable as a relevant feature for the prediction of a student final mark. In COEV the advisor grades how well the student evaluates the class plans of his/her course mates. A student that is able to evaluate the work of other people is capable to evaluate correctly his/her own work and, therefore, to execute a good work, i.e. a good final class plan. Therefore, the information conveyed by this feature is fundamental to predict the final performance of the student in the course. This conclusion has been corroborated by the advisors responsible for the course. It is also interesting to point out that co-evaluation normally is not a feature used in the e-learning environment. However, in this experiment, its high level of predictive power has been shown. On the other hand the experience report (ER) is also a relevant feature for the prediction of the students' final mark. The experience report can be viewed as a self-evaluation of his/her own learning process.

Table 7. FIR optimal mask (fuzzification parameters defined by experts).

ACT	ASS	MAIL	COEV	F	FCP	FC	IC	ER	BR	MARK
0	0	0	-1	0	0	0	-2	-3	0	+1

The root mean square error (RMS) described in Eq. (11) is used to determine the validity of the model.

$$RMS = \sqrt{\frac{\sum_{i=1}^N (y_i(t) - \hat{y}_i(t))^2}{N}} \quad (11)$$

where $\hat{y}(t)$ is the predicted output, $y(t)$ the system output and N the number of samples. The RMS errors obtained for each fold are summarized in Table 8.

Table 8. RMS errors for the 7 folds (fuzzification parameters defined by experts).

FOLD1	FOLD2	FOLD3	FOLD4	FOLD5	FOLD6	FOLD7	MEAN
0.364	0.343	0.544	0.498	0.311	0.340	0.436	0.405

The set of pattern rules derived from the FIR model is then used as input to the LR-FIR algorithm which obtains the set of linguistic rules presented in Table 9.

Table 9. Linguistic rules FIR model and its metrics (fuzzification parameters defined by experts).

Rule	Spec.	Sens.	Acc.
IF IC IN 2 AND ER IN 1 THEN MARK IN 1	0.97	0.56	0.91
IF IC IN 1 AND ER IN 1 THEN MARK IN 1	1	0.19	0.88
JOINT QUALITY CLASS 1	0.97	0.75	0.94
IF IC IN 2 AND ER IN 3 THEN MARK IN 2	0.29	0.38	0.32
IF COEV IN 3 AND ER IN 2 THEN MARK IN 2	0.91	0.32	0.72
IF COEV IN 3 AND IC IN 1 AND ER IN 3 THEN MARK IN 2	0.99	0.17	0.73
JOINT QUALITYCLASS 2	0.18	0.87	0.40
IF COEV IN 3 AND IC IN 2 AND ER IN 2-3 THEN MARK IN 3	0.60	0.99	0.81
JOINT QUALITY CLASS 3	0.60	0.99	0.81
JOINT QUALITY ALL RULES	0.56	0.92	0.69

The rules are described in class notation. The range of values for each class is defined in Table 6.

For example the first rule that defines class 1 can be also written as: IF IC IN [5.1..10] AND ER IN [0..6.9] THEN MARK IN [0..4.9]

The second, third and fourth columns show the specificity, sensitivity and accuracy measures for each rule. Specificity is defined as one minus the ratio of the number of out-of-class data records that the rule

identifies to the total number of out-of-class data. Sensitivity is the ratio of the number of in-class data that the rule identifies to the total number of in-class data. Accuracy is defined as the number of correct results to the total number of data points evaluated. The set of rules obtained were validated by the course coordinator, teachers and educative experts and conclude that the obtained results were consistent with their own perception of course student's learning behaviour. The number of linguistic rules extracted from the set of 600 pattern rules is 6. Therefore, a huge reduction has been obtained giving comprehensible explanation to the educative actors, making easiest the understanding of student's learning behaviour.

4.2. Optimization of the membership functions

The GA described in section 3 is used to obtain the membership functions (landmarks) given as granularities the ones listed in the third column of Table 6. The GA uses a population size of 50 individuals, a crossover probability of 0.6, a mutation probability of 0.1 and as stop criteria a number of chromosome evaluations of 1000 and 1500. Thirty executions have been performed for each stop criteria. A summary of the results is shown in Table 10. The first column contains a 'B' or a 'W'. B stands for best and W for worse, meaning that the membership function shown in that row is one of the best or one of the worse obtained by the GA. The second column shows these membership functions in data proportion notation. The data proportion notation is described in section 3.1.1. The third column presents the optimal mask, in position notation, encountered by FIR for these fuzzification parameters. Finally, the fourth column contains the value of the objective function.

As can be seen from Table 10, the MSE_{val} obtained for the best and worse solutions are not very far away (14.5 vs. 16.6), meaning that the GA is able to obtain good suboptimal solutions.

An interesting outcome is that almost all the FIR models obtained have the same optimal mask, i.e. (4,6,7,9,11), that corresponds to the mask shown in Table 11.

Notice that in this case COEV and ER variables are found relevant, as in the case when the discretization parameters were defined by experts (see section 4.1).

Table 10. Membership functions results obtained by the GA.

Res.	Data Proportion	Opt. Mask	MSE _{val}
B	COEV:(0.47,0.21,0.32);FCP:(0.34,0.23,0.43);FC:(0.51,0.28,0.21);ER:(0.27,0.73);MARK:(0.45,0.24,0.31)	(4,6,7,9,11)	14.4
B	COEV:(0.47,0.21,0.32);FCP:(0.38,0.21,0.41);FC:(0.50,0.26,0.24);ER:(0.32,0.68);MARK:(0.43,0.42,0.15)	(4,6,7,9,11)	14.5
W	COEV:(0.36,0.47,0.17);FCP:(0.31,0.27,0.42);FC:(0.36,0.19,0.45);ER:(0.5,0.5);MARK:(0.60,0.25,0.15)	(4,6,7,9,11)	16.5
W	COEV:(0.35,0.25,0.40);FCP:(0.26,0.27,0.47);FC:(0.39,0.28,0.33);ER:(0.46,0.54);MARK:(0.58,0.24,0.18)	(4,6,7,9,11)	16.6

Table 11. FIR optimal mask (fuzzification parameters defined by the GA).

ACT	ASS	MAIL	COEV	F	FCP	FC	IC	ER	BR	MARK
0	0	0	-1	0	-2	-3	0	-4	0	+1

Therefore, it is once again established that these variables are the more relevant ones. The initial class plan (IC) is now replaced by the forum class plan (FCP) and the final class plan (FC). It seems that IC and FCP/FC contain similar information related to the class plan being, probably, redundant.

4.3. New FIR models

The discretization proposed by the GA with a MSE_{val} of 14.5 and the optimal mask presented in Table 11 are used in this section to obtain FIR models for each fold as was done in previous work (section 4.1). The prediction errors obtained for each of the seven folds are shown in Table 12.

If we compare Tables 12 and 8 it can be seen that the mean RMS error is lower (0.342 vs. 0.405) when the discretization parameters are proposed by the GA. However, the membership functions obtained for the output variable (MARK) are not intuitive for educative actors, and will probably reduce the usefulness of the set of linguistic rules.

The experts discretized the variable MARK in the following way: [0-4.9] for class 1 (fail), (4.9-7.9] for class 2 (pass) and (7.9-10] for class 3 (excellent). The GA obtained the following discretization: [0-8.9] for class 1, (8.9-9.8] for class 2 and (9.8-10] for class 3. It is clear that the meaning that the experts give to the variable (fail, pass, excellent) is not preserved in the GA discretization. However, the GA gets a better discretization from the prediction performance point of view and the discretization is coherent with the

distribution of the data. In the course under study only a 7% of the students have a MARK lower or equal than 4.9; a 15% have a MARK between 5 and 7.9 and a 78% have a MARK higher or equal than 8.

Table 12. RMS errors for the 7 folds (fuzzification parameters defined by the GA).

FOLD1	FOLD2	FOLD3	FOLD4	FOLD5	FOLD6	FOLD7	MEAN
0.7	0.152	0.222	0.327	0.309	0.226	0.462	0.342

Trying to preserve the meaning of the MARK variable given by the experts, the following experiment is performed: the discretization parameters of the MARK variable are fixed and the GA is used only to obtain the discretization parameters of the input variables. The same GA parameters described in section 4.2 are used in this case. The optimal mask obtained in almost all the executions is the one shown in Table 11.

The prediction errors obtained for each of the seven folds when the best discretization found is used are shown in Table 13.

Table 13. RMS errors for the 7 folds (fuzzification parameters defined by the GA with variable MARK Fixed).

FOLD1	FOLD2	FOLD3	FOLD4	FOLD5	FOLD6	FOLD7	MEAN
0.693	0.211	0.302	0.259	0.342	0.229	0.397	0.347

Notice that the mean error is in between the error obtained when fuzzification parameters are defined by experts, 0.405 (Table 8), and the error obtained when the GA is used without restrictions, 0.342 (Table 12), being much closer to the last one. This makes sense since in this case an optimization of the inputs is performed.

4.4. New LR-FIR models

The discretizations proposed by the GA and the GA with the variable MARK fixed are here used to obtain LR-FIR models. As in section 4.1 an improved compaction based on the minimal ratio is chosen. Also, a filtering threshold of 0.1 is applied before and after unification, which means that the rules with specificity or sensitivity values lower or equal to 0.1 were deleted. The same filtering procedure has been applied in the rules presented in Table 9. The set of rules obtained when the fuzzification parameters proposed by the GA and the GA with MARK fixed are used, are shown in Tables 14 and 15, respectively.

Analyzing the joint quality of all rules shown in Tables 9, 14 and 15 (last row), it is quite clear that the LR-FIR models that use the discretization parameters given by the GA are able to define more reliable rules, due to the fact that the associated metrics are balanced. This is not the case for the set of rules obtained when the discretization parameters are defined by experts. In that case (Table 9) specificity (0.56), sensitivity (0.92) and accuracy (0.69) are clearly unbalanced, reducing the consistency of the set of rules.

Table 14. Linguistic rules FIR model and its metrics (fuzzification parameters defined by the GA).

Rule	Spec.	Sens.	Acc.
IF COEV IN 1 AND ER IN 1 THEN MARK IN 1	1	0.26	0.69
IF FCP IN 1 AND ER IN 1 THEN MASK IN 1	1	0.25	0.68
IF FCP IN 1 AND FC IN 2 AND ER IN 2 THEN MARK IN 1	0.88	0.19	0.59
IF FCP IN 1 AND FC IN 1 AND ER IN 2 THEN MARK IN 1	0.97	0.14	0.64
JOINT QUALITY CLASS 1	0.85	0.68	0.78
IF FCP IN 3 AND FC IN 2 AND ER IN 2 THEN MASK IN 2	0.64	0.46	0.57
IF FCP IN 1 AND FC IN 2 AND ER IN 2 THEN MARK IN 2	0.87	0.17	0.59
IF FCP IN 3 AND FC IN 1 AND ER IN 2 THEN MARK IN 2	0.93	0.15	0.61
IF FCP IN 2 AND FC IN 2 AND ER IN 2 THEN MARK IN 2	0.96	0.13	0.62
JOINT QUALITYCLASS 2	0.41	0.90	0.51
IF COEV IN 3 AND FCP IN 3 AND FC IN 2 AND ER IN 2 THEN MARK IN 3	0.91	0.91	0.91
JOINT QUALITY CLASS 3	0.91	0.91	0.91
JOINT QUALITY ALL RULES	0.66	0.81	0.72

Moreover, the specificity measure is rather low, 0.56, meaning that there are a quite large number of students that do not fit in the set of rules that define his/her MARK.

The linguistic rules shown in Table 15 are the more consistent and reliable ones, being also the more useful for decision support purposes because the output (MARK) discretization preserves the meaning defined by the experts.

The set of rules obtained when the discretization parameters are learned (Tables 14 and 15) have quite good metrics, however higher metrics would be desirable. To this end, future work is focused on the development of a GFS that uses as GA objective function a combination of the three metrics used in this work, i.e. specificity, sensitivity and accuracy. Moreover, it would be interesting to use a

multiobjective GA to perform this task. It is also crucial that the objective function takes into account not only the accuracy but also the linguistic quality of the discretization, in order to allow the MARK variable be tuned by the GFS.

Table 15. Linguistic rules FIR model and its metrics (fuzzification parameters defined by the GA with variable MARK Fixed).

Rule	Spec.	Sens.	Acc.
IF COEV IN 1 AND FC IN 1 AND ER IN 1 THEN MARK IN 1	0.97	0.78	0.96
JOINT QUALITY CLASS 1	0.97	0.78	0.96
IF FCP IN 1 AND FC IN 1 AND ER IN 2 THEN MARK IN 2	0.94	0.19	0.83
IF FCP IN 1 AND FC IN 1 AND ER IN 1 THEN MARK IN 2	0.95	0.17	0.83
IF FCP IN 1 AND FC IN 2 AND ER IN 1 THEN MARK IN 2	0.99	0.16	0.86
IF FCP IN 1 AND FC IN 2 AND ER IN 2 THEN MARK IN 2	0.85	0.13	0.74
JOINT QUALITYCLASS 2	0.73	0.65	0.72
IF FCP IN 3 AND FC IN 2 AND ER IN 2 THEN MARK IN 3	0.97	0.50	0.61
IF FCP IN 1 AND FC IN 2 AND ER IN 2 THEN MARK IN 3	0.90	0.16	0.33
IF FCP IN 3 AND FC IN 1 AND ER IN 2 THEN MARK IN 3	0.93	0.11	0.30
JOINT QUALITY CLASS 3	0.80	0.78	0.78
JOINT QUALITY ALL RULES	0.77	0.76	0.77

5. Conclusions

A GFS in which a GA learns fuzzification parameters within the Fuzzy Inductive Reasoning (FIR) methodology with the goal to enhance predictive (FIR) and decision support (LR-FIR) models identification is presented in this paper. The GFS is used to obtain FIR and LR-FIR models in an e-learning application, i.e. Didactic Planning graduate course, with the goal of predicting students' performance and to understand students' learning behaviour. The results obtained are encouraging, because better predictive and decision support models are obtained when discretization parameters are learned by GAs. The next step is the development of GFSs with objective functions based on specificity, sensitivity and/or accuracy, in such a way that a better learning could be performed for decision support models. The objective function should also take into account the linguistic quality of the discretization.

References

1. F. Castro, A. Vellido, A. Nebot and F. Mugica, Applying Data Mining Techniques to e-Learning Problems, in *Evolution of Teaching and Learning Paradigms in Intelligent Environment. Studies in Computational Intelligence* **62**, (L.C. Jain, R.A. Tedman & D.K. Tedman, Ed. Springer-Verlag Germany, 2007), pp. 183-221.
2. M. Croock, F. Mofers, M. Van Veen, P. Van Rosmalen, F. Brouns, J. Boticario, C. Barrera, O. Santos, A. Ayala, E. Gaudio, F. Hernández, C. Arana and I. Trueba, State-of-the-Art, ALFanet/IST-2001-33288 Deliverable D12. Open Universiteit Nederland. URL: <http://learningnetworks.org/> 2002.
3. P. Van Rosmalen, F. Brouns, C. Tattersall, H. Vogten, J. van Bruggen, P. Sloep and R. Koper, Towards an Open Framework for Adaptive Agent-Supported e-Learning, in *International Journal Continuing Engineering Education and Lifelong Learning* **15**(3-6) (2005), pp. 261-275.
4. B. Minaei-Bidgoli, P.N. Tan and W.F. Punch, Mining Interesting Contrast Rules for a Web-based Educational System, in *Proceedings of The 2004 International Conference on Machine Learning and Applications, ICMLA'04*. Louisville, KY (2004).
5. C. Romero, S. Ventura, P. De Bra and C. De Castro, Discovering Prediction Rules in AHA! Courses,” in *Proceedings of the User Modelling Conference*, Johnstown, Pennsylvania (2003), pp. 35-44.
6. M. Van der Klink, J. Boon, E. Rusman, M. Rodrigo, C. Fuentes, C. Arana, C. Barrera, I. Hoke and M. Franco, Initial Market Study, ALFanet/IST-2001-33288 Deliverable D72. Open Universiteit Nederland. URL: <http://learningnetworks.org/downloads/alfanet-d72-initialmarket-studies.pdf> (2002).
7. M.J. Gacto, R. Alcalá and F. Herrera, Integration of an Index to Preserve the Semantic Interpretability in the Multi-Objective Evolutionary Rule Selection and Tuning of Linguistic Fuzzy Systems, in *IEEE Transactions on Fuzzy Systems* **18**(3) (2010), pp. 515-531.
8. R. Alcalá, J. Alcalá-Fdez, F. Herrera and J. Otero, Genetic Learning of Accurate and Compact Fuzzy Rule Based Systems Based on the 2-Tuples Linguistic Representation, in *International Journal of Approximate Reasoning* **44**(1) (2007), pp. 45-64.
9. R. Alcalá, J. Alcalá-Fdez, M.J. Gacto and F. Herrera, Rule Base Reduction and Genetic Tuning of Fuzzy Systems based on the Linguistic 3-Tuples Representation, in *Soft Computing* **11**(5) (2007), pp. 401-419.
10. I.G. Damousis and P. Dokopoulos, A fuzzy expert system for the forecasting of wind speed and power generation in wind farms, in *Proc. 4th Int. Conf. Power Industry Computer Applications (PICA 2001)* (Sydney, 2001), pp. 63-69.
11. R. Alcalá, M.J. Gacto and F. Herrera, A Fast and Scalable Multi-Objective Genetic Fuzzy System for Linguistic Fuzzy Modeling in High-Dimensional Regression Problems, in *IEEE Transaction on Fuzzy Systems*, in press (2011), doi:10.1109/TFUZZ.2011.2131657.
12. H.B. Gürocak, A genetic-algorithm-based method for tuning fuzzy logic controllers, in *Fuzzy Sets Syst.* **108**(1), (Nov. 1999), pp. 39-47.
13. F. Hoffmann, Evolutionary Algorithms for Fuzzy Control System Design, in *Proc. IEEE* **89**(9) (Sept. 2001), pp. 1318-1333.
14. I-F. Chung, C.J. Lin, and C.T. Lin, A GA-based fuzzy adaptive learning control network, in *Fuzzy Sets Syst.* **112**(1), (May 2000), pp. 65-84.
15. H.A. Camargo, M.G. Pires, and P.A.D. Castro, Genetic Design of Fuzzy Knowledge Bases – a study of different approaches, in *Proc. IEEE Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS'04)* **2** (June 2004), pp. 954-959.
16. M.J. del Jesús, F. Hoffman, L.J. Navascués, and L. Sánchez, Induction of Fuzzy-Rule-Based Classifiers with Evolutionary Boosting Algorithms, in *IEEE Trans. Fuzzy Syst.* **12**(3) (June 2004), pp. 296-308.
17. G. Klir and D. Elias, *Architecture of systems problem solving*, (2nd ed. New York, NY: Plenum Press, 2002).
18. A. Jerez and A. Nebot, Genetic algorithms versus classical search techniques for identification of fuzzy models, in *5th European congress on Intelligent Techniques and Soft Computing*, Aachen (1997), pp. 769-773.
19. A. Law and D. Kelton, *Simulation Modeling and Analysis*, (2nd ed., McGraw-Hill, New York, 1990).
20. A. Nebot, F.E. Cellier and M. Vallverdú, Mixed quantitative/qualitative modeling and simulation of the cardiovascular system, in *Computer Methods and Programs in Biomedicine* **55** (1998), pp. 127-155.
21. C. Mencar and A.M. Fanelli, Interpretability constraints for fuzzy information granulation, in *Information Sciences* (2008).
22. S.M. Zhou and J.K. Gan, Low-level interpretability and high-level interpretability: a unified view of data-driven interpretable fuzzy system modelling, in *Fuzzy Sets and Systems* 2008.
23. M.J. Gacto, R. Alcalá and F. Herrera, Interpretability of Linguistic Fuzzy Rule-Based Systems: An Overview of Interpretability Measures, in *Information Sciences*, in press (2011), doi:10.1016/j.ins.2011.02.021.
24. F. Castro, A. Nebot and F. Múgica, On the extraction of decision support rules from fuzzy predictive models, in *Applied Soft Computing* **11**(4) (2011), pp. 3463-3475.
25. J. Holland, *Adaptation in Natural and Artificial Systems* (Ann Arbor, MI, USA: University of Michigan Press, 1975).

26. D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning* (Upper Saddle River, NJ, USA: Addison-Wesley Professional, 1989).
27. J. Acosta, A. Nebot, P. Villar and J.M. Fuertes, Optimization of Fuzzy Partitions for Inductive Reasoning using Genetic Algorithms, in *Int. Journal of Systems Sciences* **38**(12) (2007), pp. 991-1011.