Neuro Fuzzy Reasoner for Student Modeling

Zoran Sevarac
Department of Information Systems, School of Business Administration FON,
University of Belgrade, Serbia and Montenegro
sevarac@gmail.com

Abstract

This paper presents a neuro-fuzzy system that can be used for student modeling. The proposed system enables classification of students based on qualitative observations of their characteristics.
The system is very flexibile and can be easily adapted to individual teacher's preferences. It is available as a Java component and it can be used in e-learning applications.

1. Introduction

The main issue with student modeling is the interpretation of the student's behaviour. The student's behaviour is any observable response to a particular stimulus during his interaction with the system. Since the communication channel between the student and the system is usually limited to a keyboard and a mouse, the information gathered about the student's behaviour is very restricted. This information is imprecise, error prone, and its interpretation is vague and uncertain.
Also, each teacher has his own teaching strategy, criteria and methods for the evaluation of the student's perfomance. So, the student model that is suitable for one teacher, may not be suitable for some other teachers.

This paper suggests a neuro-fuzzy approach to student modelling that deals with the issues mentioned above. The fuzzy model successfully handles reasoning with imprecise information, and enables representation of student modeling in the linguistic form - the same way the human teachers do. The underlying neural network enables adaptivity of the fuzzy model. The proposed neuro-fuzzy model enables creation of an easy-to-use, customized student modeling component.

The paper is organized as follows. Section 2 briefly reviews the fuzzy approach to student modeling. Section 3 presents details of the proposed neuro-fuzzy (NF) model. Section 4 describes the application of the proposed NF model to student modeling problem. Section 5 shows the evaluation of the proposed solution for student modeling. Section 6 briefly reviews related work, and compares it to the proposed model. Section 7 concludes the work and indicates future research directions.

2. Fuzzy student modeling

The fuzzy approach enables approximate reasoning and it is suitable for modeling human decision process. By using the linguistic variables and fuzzy sets, the translation from verbal to fuzzy model is straightforward. In this way, the fuzzy approach is also used for student modeling problem. The goal of fuzzy student modeling is to imitate the student modeling strategy used by human teacher.

Human teachers do not build detailed models for understanding the student performance and adapting their teaching strategy. They gather information and form general ideas of what kind of teaching might work better for each student. According to some findings students are usually classified in terms of a few underlying dimensions like motivation, intellectual ability and knowledge level on some topic.
According to the IMS LIP specification, the student classification can be based on activity evaluation.
This sort of classification can be easily expressed in terms of fuzzy logic. For example:

IF (TEST_RESULT IS LOW)
THEN STUDENT_CLASS IS BAD

This rule says that if a student has low result on a test, he is classified as a bad student. The expression (TEST_RESULT IS LOW) is the premise, and the expression (STUDENT_CLASS IS BAD) is the consequence of this fuzzy rule. TEST_RESULT and STUDENT_CLASS are linguistic variables, and their corresponding values are LOW and BAD. The premise of a fuzzy rule is always a fuzzy value, but the consequence may be a fuzzy or a crisp value. In this example, LOW is a fuzzy set, and BAD is a crisp value - class representing the classification of the student. The value of the premise is evaluated as the value of the membership function of the fuzzy set LOW.

The premise can also include several expressions and tie them with fuzzy logic operators. For example:

IF ((TEST_RESULT IS HIGH) AND
(STUDENT_SPEED IS FAST))
THEN STUDENT_CLASS IS EXCELLENT

3. Neuro-fuzzy reasoner

Neuro-fuzzy reasoner (NFR) is a software component capable of learning the set of predefined fuzzy rules. This learning capability enables creation of adaptive fuzzy-rule systems. Fig. 1 shows the structure of the NFR system.

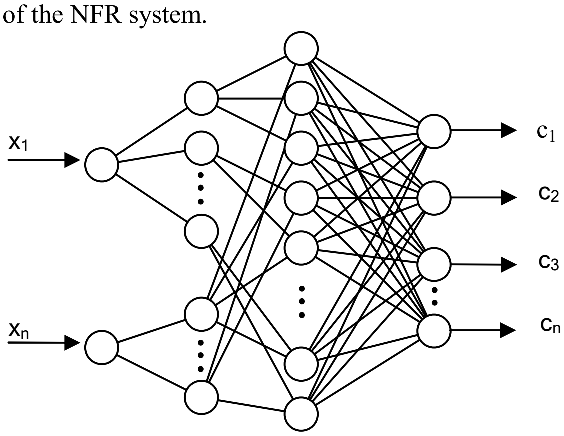

Figure 1. NFR structure

The system has a four-layer feedforward network architecture. The layers are denoted L1, L2, L3 and L4. The L1 layer introduces to the system numerical values describing the student features like test score, test time, number of correct or wrong answers, how many times student reviewed lesson, how many times student took test etc. L2 is the fuzzification layer. The units in this layer have fuzzy membership functions as transfer functions. The purpose of this layer is to fuzzify the input values – to translate them into fuzzy sets. Each unit in this layer corresponds to a single fuzzy set that appears in the premise part of a fuzzy rule.

L3 is the premise layer and its purpose is to calculate the activation of premises of the fuzzy rules. Each unit in this layer corresponds to a certain rule. The units in L3 implement AND operators by means of minimum type t-norms [3]. Minimum type t-norm is one way of implementing logical AND operation on fuzzy values, and it simply calculates min function. Connection weights between the layers L1 and L2, and L2 and L3 are fixed to 1.

The fourth layer L4 implements output units, one for each consequence (in the case of student classification one for each type of students, e.g., BAD, GOOD, EXCELLENT, etc.). The connection weights between layers L3 and L4 are trained using least mean squares algorithm. The links between premises and consequences of fuzzy rules are stored in these weights. The training set creation is based on the parameters of fuzzy membership functions from the layer L2, and the explicitly defined set of fuzzy rules.

The NFR system provides a simple way for the user to create a neuro-fuzzy classifier based on the student's prior knowledge. The corresponding NFR system is directly extracted from the fuzzy model. Since the fuzzy model is very close to verbal model, NFR makes it easy to create fuzzy rule system according to the expert's knowledge .

4. Application example

The proposed neuro-fuzzy model can be successfully used for addressing student modeling issues mentioned in the introduction. As an illustration, a NFR model for student classification based on test results and the time needed to complete the test is presented here.

The steps to take in order to apply the NFR model the student modeling problem in this case are as follows:

1. Defining input and output values;
2. Defining fuzzy sets for input values;
3. Defining fuzzy rules;
4. Creating and training the neural network

1. Input and output values

Input values:

- Test score [0..100]
- The time needed to complete the test [0..120]

Output values:

- Classes of students: {Bad, Good, Very good, Excellent}

2. Fuzzy sets

The input space is partitioned by the following fuzzy sets:

- Test score: Bad, Low, Mid, High
- The time needed to complete the test, interpreted as speed: Slow, Moderate, Fast

The corresponding membership functions are shown in figures 2 and 3.

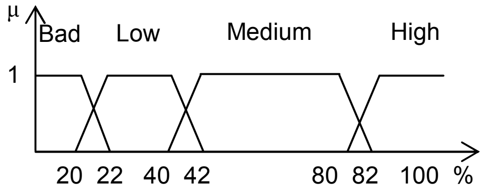

Figure 2. Test score membership functions

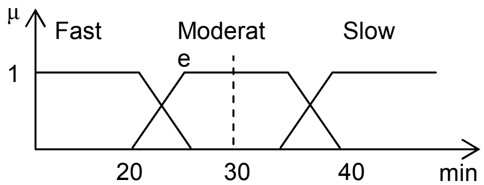

Figure 3.Test completion speed membership function

3. Fuzzy rules
The rules for student classification are taken from the human teacher. Twelve such rules are shown in Table 1. The values of linguistic variables in the premises are interpreted as the previously defined fuzzy sets, and the rules bellow are interpreted like:

IF ((TEST_SCORE IS HIGH) AND
(STUDENT_SPEED IS FAST))
THEN STUDENT_CLASS IS EXCELLENT

4. Creating and training the neural network
When the fuzzy model is defined, the construction of the corresponding NFR model is straightforward. The NFR model that corresponds to the previously defined fuzzy model is shown in fig. 4.
The network is constructed using the following principles:

1. The number of cells in the input layer L1 is equal to the number of inputs;
2. The number of cells in the fuzzyfication layer L2 is equal to the number of fuzzy sets;
3. The number of cells in the premise layer is equal to the number of rules;
4. The number of cells in the output layer is equal to the the number of classification classes;
5. The connection pattern is the same for all NFR models and it is shown in fig. 4.

Table 1. Fuzzy rules for student classification

SCORE	SPEED	CLASS
Bad	Slow	Bad
Bad	Moderate	Bad
Bad	Fast	Bad
Low	Slow	Bad
Low	Moderate	Good
Low	Fast	Good
Medium	Slow	Good
Medium	Moderate	Very good
Medium	Fast	Very good
High	Slow	Very good
High	Moderate	Excellent
High	Fast	Excellent

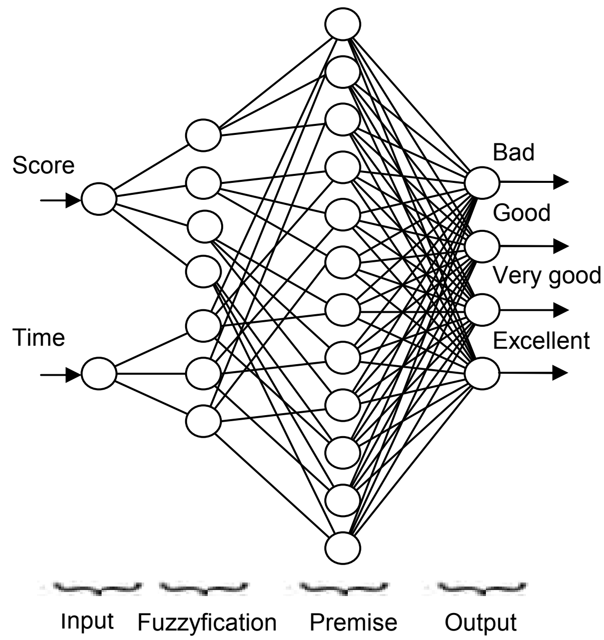

Figure 4. The NFR model for student classification

In our practical implementation of this student modeling system, we used the our JAVA neural network framework *Neuroph*. The training set for this example was automatically generated from the rule set and the parameters of the membership functions, and it contained 120 training elements. The training was completed in 6-12 iterations with no error, which is considered as a very good result. The training diverged only if there were contradictory rules.
Figures 5 and 6 show the developed test application. Figure 5 presents the user interface for the underlying neuro-fuzzy system.

The user (the teacher, or the student (in case the student model is open to the student)) sets the test score (points) and time, and clicks the *Classify* button. In the example in Fig 5. 60 points correspond to the medium test score, and 36
minutes to moderate speed. The student with such a performance is classified as a *Very good* student according to rule from Table 1.

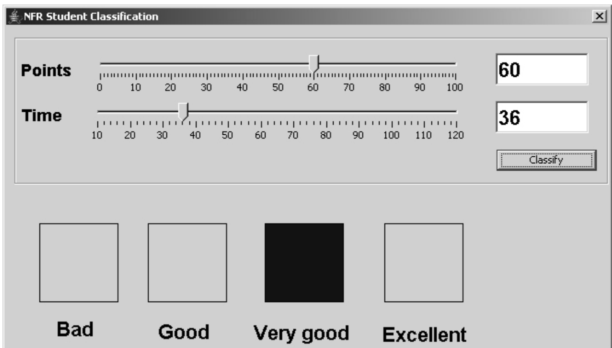

Figure 5. User interface of the student classification neuro-fuzzy component

Figure 6 shows a network view of the implemented neuro-fuzzy system in the *SmartNet* application. The application is a part of our JAVA neural network framework *Neuroph,* and is used for testing purposes. The active cell in the output layer marked with darker color corresponds to the *Very good* class. The basic principle is clearly shown in this example: if a student belongs to some predefined class, the corresponding cell for that class in output layer is activated.

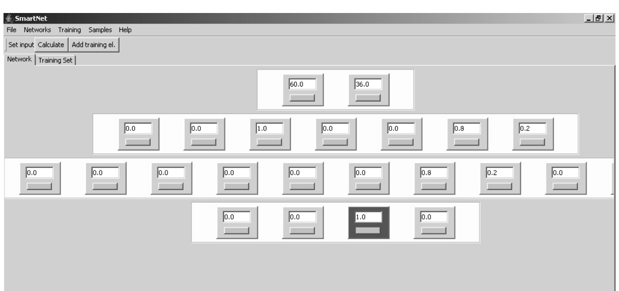

Figure 6. Network view of the nero-fuzzy system for student classification

5. Evaluation

The system was tested with two generated test sets, each one having 50 test elements. Each test element was given input and desired output values. The first test set contained values whose membership degrees were clearly defined. This means that there were no cases with membership degrees equal to 0.5 (when the membership degree of some value is 0.5 it is undecidable to which fuzzy set that value belongs). The second test set included these cases also.
With the first test set the success rate was 100%, even with the values that had membership degree close to 0.5. The elements from the second test set with values that had membership degree equal to 0.5 were classified into two categories, which was not a desirable behaviour. These elements were than included into a training set with a clearly defined classification category. After repeating the training these elements, the system reached the success rate of 100% with the second training set. Therefore, the solution for the misclassification problem can be one of the following:

1. including the values that can be misclassified into the initial training set with the defined classification
2. defining principle for solving these cases: classify them into lower or higher order fuzzy set

The system had the success rate of 100% also for the data that were not used during the training.

After this initial testing, the system was tested within a real world e-learning application called Multitutor [1]. The group of 15 students took Multitutor test in basic C programming. The test had 10 questions, and lasted 30 min. Test results and individual students' times were used as input for the NFR classifier, and the classification matched the decisions of a human teacher.

Although the test sample was small, an assumption is that the system can easily handle a larger number of samples without any difficulties. This assumption is based on experience with the system during the testing.

6. Related work

Other authors have been applying as well neuro-fuzzy systems to the student modeling problem. The basic principle is the same for all, including our system: translate the verbal model to the fuzzy one, and then use a neural network to implement and adapt the fuzzy model.

Among the other researchers who worked in this field are Stathacopoulou [2], and more recently Mir Sadique Ali [3]. Stathacopoulou has developed a well-defined framework for neuro-fuzzy student modeling. It accepts observable student behavior as input, and outputs an estimation of the student's characteristics.

This neuro-fuzzy model has input, relation, aggregation, and output blocks. The input and output blocks perform fuzzyfication and defuzzyfication respectively, while the relation and aggregation blocks implement fuzzy inference.

The NFR model has a similar but simpler architecture with less cells. Of course, it gives more rough approximations than the model by Stathacopoulou but this can be overcome by using a few NFRs in parallel and changing transfer functions in the output layer from step to linear. Comparing to the Stathacopoulou's model, NFR is simpler but gets the job done.

Anfis was one of the first rule-based neuro–fuzzy systems for function approximation. Sadique used it in his work on student modeling [3]. Anfis is not a good solution for applications where interpretation is important because it uses fuzzy inference of Sugeno type [4]. This is a major drawback in its application to student modeling. However, Sadique tested this system with his rule base and got satisfactory results. NFR rules are very similar to the one used by Sadique, but NFR also has the interpretation capability because it uses Mamdani type rules.

The NEFCLASS system is a general-purporse neuro-fuzzy classifier introduced by Nauck and Kruse [5]. It can be used to determine the correct class or category of a given input pattern. The NEFCLASS learning algorithm changes the values of membership function parameters, and it is not suitable for fuzzy modeling used here since the values represent the encoded expert's knowledge and should not be changed. The idea of the NEFCLASS system was the initial inspiration for the NFR system.

The NFR system combined some ideas from all three mentioned approaches into a new neuro-fuzzy system that can be used for classification and reasoning problems.

Some popular uncertain classification models [6] are based on the Bayes classification algorithm. The main drawback of this approach is that the teacher must describe classification in terms of conditional probabilities. This makes creating classification model too complicated for end users, and the created model is hard to inspect.

Comparing to some other statistical testing schemas like adaptive testing, NFR offers a customized, multi-dimensional classification, and a well structured, semantically rich classification model. But it is important to note that NFR does not offer real-time student evaluation like adaptive testing, nor it is its purpose. The NFR can be used to evaluate the total test result by observing several parameters defined by the teacher.

The idea of fuzzy student modeling is not new and there are many fuzzy student models that have been successfully applied in various e-learning applications. By using a simple neural network, the NFR adds one more ingredient to the proven concept of fuzzy student modeling: adaptivity. This way, the evaluation strategy (classification) is not defined at development time. It is very flexible and it can be entirely defined by the end user (teacher). This is possible because the appropriate NFR model can be automatically generated from the set of high-level rules, using a wizard. Maybe the lack of adaptivity was the main reason why the fuzzy approach was not widely used in practice so far, despite the good results and the fact that it could be the natural solution for the student modeling problem.

7. Conclusions

This paper proposed a neuro-fuzzy system that can be used for student modeling. The proposed system is relatively simple, supports creation of higl-level pedagogical strategies, and can be easily adapted to individual teacher's preferences.

The system is available as a JAVA component and can be used in e-learning applications, but also in any other type of applications thar require user modeling, classification or reasoning.

Further development will include a specialized software tool for creating and adjusting student modeling, classification, or reasoning components.

8. References

[1] G. Simic, The Multi-course Tutoring System Design, Journal Computer Science and Information Systems, Vol.1, February 2004

[2] R. Stathacopoulou, G. D. Magoulas , M. Grigoriadou, M. Samarakou, Neuro-fuzzy knowledge processing in intelligent learning environments for improved student diagnosis, 1999.

[3] Mir Sadique Ali, Ashok A. Ghatol, An Adaptive Neuro Fuzzy Inference System For Student Modeling In Web-Based Intelligent Tutoring Systems, 2005.

[4] Marian B. Gorzalczany, Computational Intelligence Systems and Applications, Springer Verlag, 2002.

[5] U. Nauck, R. Kruse, Design and Implementation of a Neuro-Fuzzy Data Analysis Tool in Java, 1999.

[6] K. VanLehn, Evaluation of an assessment system based on Bayesian student modeling, International Journal of Artificial Intelligence in Education, 8(2) , 1998.