Learning Styles and Emotion Recognition in a Fuzzy Expert System

Ramón Zatarain-Cabada, M. Lucía Barrón-Estrada, Rosalío Zatarain-Cabada

Instituto Tecnológico de Culiacán, Juan de Dios Bátiz s/n, Col. Guadalupe, Culiacán Sinaloa, 80220, Mexico

{rzatarain, lbarron}@itculiacan.edu.mx

Abstract. This paper presents a fuzzy system that recognizes learning styles and emotions using two different neural networks. The first neural network (a Kohonen neural network) recognizes the student cognitive style. The second neural network (a back-propagation neural network) was used to recognize the student emotion. Both neural networks are being part of a fuzzy system used into an intelligent tutoring system. The fuzzy system evaluates both cognitive and affective states in the student whenever he/she answers math exercises.

Keywords: Intelligent tutoring systems, affective computing, learning technologies, artificial neural networks, education.

1 Introduction

In last years, Intelligent Tutoring Systems (ITS) have integrated the ability to recognize the student's affective state, in addition to traditional cognitive state recognition. Research on affective computing includes detecting and responding to affect. Affect detection systems identify frustration, interest, boredom, and other emotions [1, 2,]. On the other hand, affect response systems transform negative emotional states (frustration, boredom, fear, etc.) to positive ones [3, 4]. Ekman's work on face analysis [5] describes a subset of emotions including joy, anger, surprise, fear, disgust/contempt and interest, which have been used in new ITS, which include the recognition and treatment of emotions and/or feelings [1, 3, 6, 7].

In this work, we present a system that combines affective computing and learning styles into a fuzzy system which is part of a ITS. We have integrated two methods for selecting the learning style and emotional state of a student and to consider them in the ITS response. For recognizing the learning style and the affective state, we implemented two neural networks. During a training session, the first network (a SOM or Kohonen network) used for detecting the learning styles, receives a number of different input patterns (the student learning style obtained from an Inventory Learning Style Questionnaire (ILSQ), the learning style of three defined courses, and the student's grade in each course), discovers significant features in these patterns and learns how to classify the input patterns. The second network (a back-propagation neural network), which is used to detect affective or emotional states, is trained with a corpus of faces representing different emotional states. The affective and learning style recognizers are used into a fuzzy system, which is part of an ITS.

2 Recognizing Learning Styles and Emotional States

By using the neural networks, the learning style and the emotional state of a student can be dynamically calculated according to evaluations and face recognition applied to the student while he/she is using the ITS.

A course can be seen as a discipline-specific knowledge space (a particular tree diagram) containing chapters, which in turn are made by subjects. The total of nodes in the tree represents the domain or expert knowledge. Figure 1 shows the knowledge domain of the math course with subjects related to arithmetic operations such as multiplication and division and topics like fractions. For each topic, different learning instances teaching the same theme under different learning styles can be observed. In this way, many combinations or paths are provided for learning the same topic. Dashed lines represent those paths. The domain module is stored as a XML-based document.

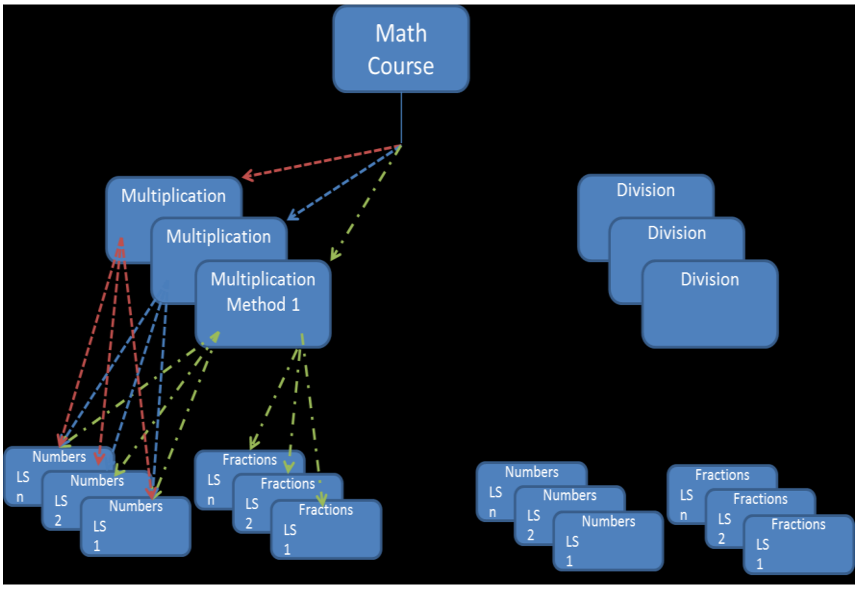

Fig. 1. Knowledge domain of a math course under different learning styles (LS)

2.1 The Kohonen Neural Network for Recognizing Emotional States

The method used for the detection of visual emotions is based on Ekman's theory (Ekman and Friesen, 1975). The recognition system was built in three stages: the first one was an implementation to extract features from face images in a corpus used to train the neural network. The second one consisted of the implementation of the neural network. The third stage integrated extraction and recognition into the fuzzy system. For training and using the neural network we used the corpus RAFD (Radboud Faces Database) [8] which is a database with 8040 different facial expressions, which contains a set of 67 models including men and women. Once the emotion state is extracted from the student the state is sent to the fuzzy system (see figure 2).

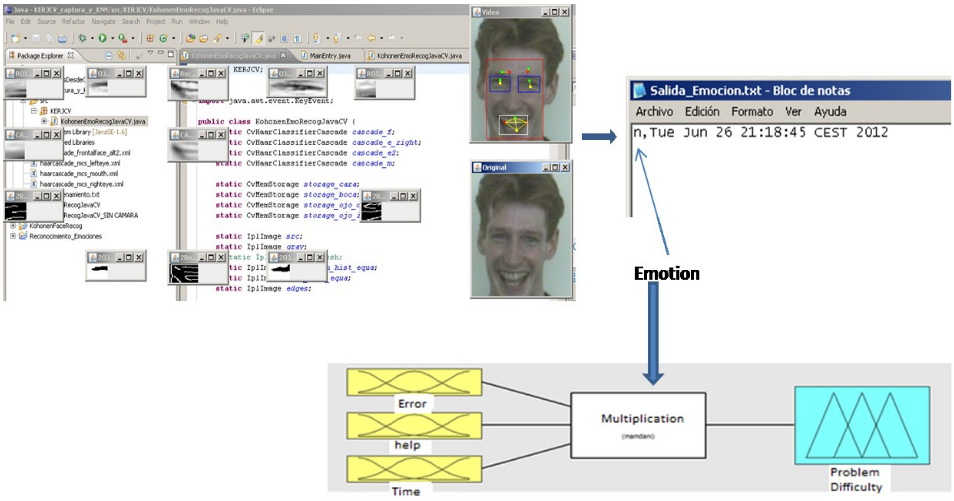

Fig. 2. Emotion extraction in the student

2.2 A Kohonen Neural Network for Learning Styles

The input layer of the neural network for learning styles has 7 neurons. The Kohonen layer has 1600 neurons, organized in a lattice of hexagonal cells with dimensions of 40x40 neurons. The signals are part of the training data space and they are vectors composed of three elements: two vectors and a scalar. The first vector is the student's learning style identified by using the ILSQ questionnaire. The second vector is the learning style of the learning material read by the student (three courses about Computer Science, Photography, and Eolithic Energy). The last element is the student’s performance. The neural network provides the student's learning style as an output value. The number of iterations used for training the neural network was 5000 iterations. Figure 3 shows part of the training of the Kohonen or SOM neural network.

3 The Fuzzy Expert System

The student module provides information about student knowledge and learning aptitudes. The module identifies what the student's knowledge is through a diagnostic test. The student knowledge can be seen as a subset (sub-tree implemented) of all knowledge possessed by the expert in the domain (module) and this is stored in a student profile.

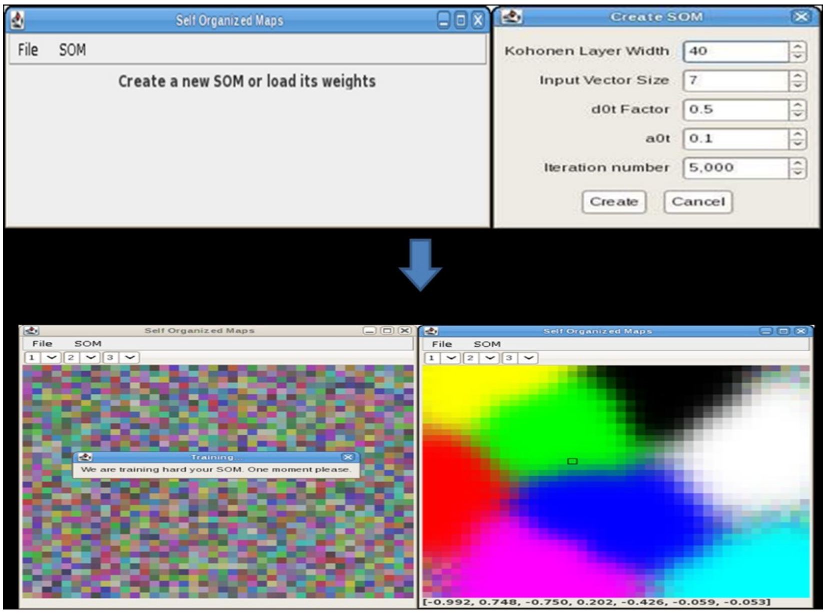

Fig. 3. Training of the Kohonen network for recognizing learning styles

For every student there are a static and a dynamic profile, which store particular and academic information like affective states, learning styles, and scoring results. In the ITS, a fuzzy expert system was implemented with a new knowledge tracing algorithm, which is used to track student's pedagogical states, applying a set of rules. The benefit of using fuzzy rules is that they allow inferences even when the conditions are only partially satisfied. The fuzzy system uses input linguistic variables such as *error, help, time*, *Emotion*, and *Learning Style* (Figure 4). These variables are loaded when the student solves an exercise. The output variable of the fuzzy system is the difficulty and type of the next exercise. The type is defined according the learning style assigned to the student.

3.1 The Fuzzy Sets

The proposed fuzzy sets, for each linguistic variable are:

- Error = {low, normal, many}
- Help = {little, normal, helpful}
- Time = {very fast, fast, slow, very slow}
- Emotion = { anger, disgust, fear, happiness, sadness, surprise, and neutral }
- Learning Style = { Visual, Verbal, Sequential, Global, Sensitive, Intuitive }
- Difficulty = {very easy, easy, basic, hard, very hard}

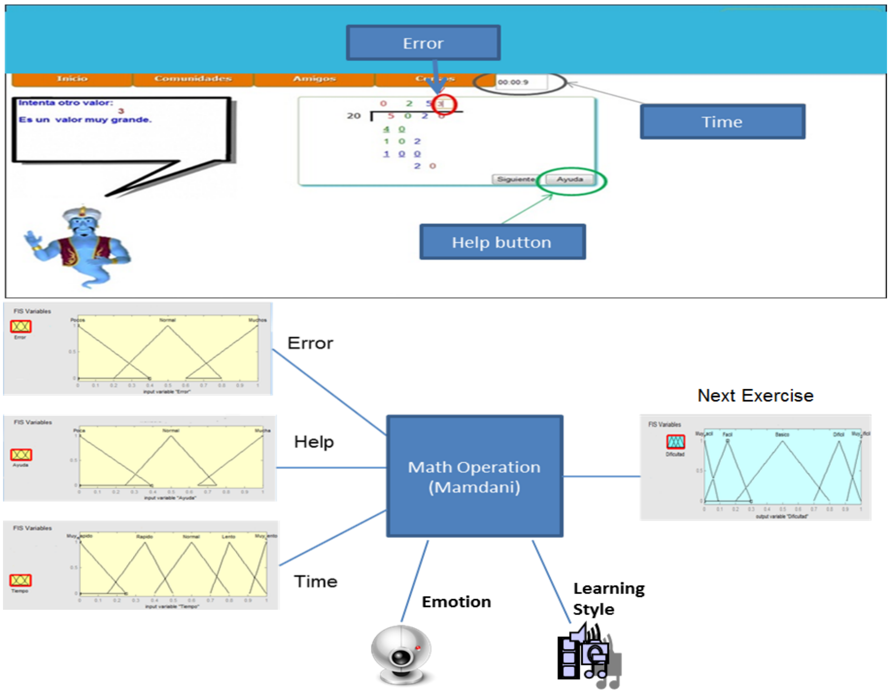

Fig. 4. Input and Output of fuzzy variables

3.2 Rule Evaluation

One important step of our fuzzy expert system is to evaluate the fuzzy values of the input variables. Table 1 shows a sample of some of the fuzzy rules that are used in the system.

Table 1. A sample of fuzzy rules of the expert system

Rule #	Rule
Rule 1	If(**Error** is low) and (**Help** is **little**) and (**Time** is **very fast**) and (Emotion is **neutral**) then (**Difficulty** is **very_hard**)
Rule 2	If(**Error** is low) and (**Help** is **little**) and (**Time** is **fast**) and (Emotion is **neutral**) then (**Difficulty** is **very_hard**)
Rule 3	If(**Error** is low) and (**Help** is **little**) and (**Time** is **normal**) then (**Difficulty** is **very_hard**)
Rule 4	If(**Error** is low) and (**Help** is **little**) and (**Time** is **slow**) then (**Difficulty** is **hard**)
Rule 5	If(**Error** is low) and (**Help** is **little**) and (**Time** is **very-slow**) then (**Difficulty** is **hard**)
Rule 6	If(**Error** is low) and (**Help** is **normal**) and (**Time** is **slow**) and (Emotion is **sadness**) then (**Difficulty** is **basic**)
…	…
Rule 41	If(**Error** is **many**) and (**Help** is **helpful**) and (**Time** is **very-slow**) and (Emotion is **fear**) then (**Difficulty** is **very_easy**)

In order to evaluate the conjunction of the rule antecedent, we applied the following equation:

$$\mu_{A\cap B\cap C\ldots\cap Z}(x) = \min[\mu_A(x), \mu_B(x), \mu_c(x), \ldots, \mu_Z(x)] \quad (1)$$

To evaluate disjunction, we applied equation:

$$\mu_{A\cup B\cup C\ldots\cup Z}(x) = \max\,[\mu_A(x), \mu_B(x), \mu_c(x), \ldots, \mu_Z(x)] \quad (2)$$

For instance, to evaluate the next fuzzy rule:

IF *Error is low (0.3)*
AND *Help is little (0.2)*
AND *Time is very-fast (0.1)*
AND *Emotion is neutral (0.2)*
THEN *Difficulty is very-hard (0.1)*

Equation 1 is applied:

$\mu_{\text{very-hard}}(\textit{Difficulty}) = \min[\mu_{\text{low}}(\textit{Error}), \mu_{\text{little}}(\textit{Help}), \mu_{\text{very-fast}}(\textit{time}),$
$\mu_{\text{neutral}}(\textit{emotion})] = \min[0.3,0.2,0.1,0.1] = 0.1$

3.3 The Implementation of an Exercise

Next, we present the structure of the XML-based file of six exercises about integer divisions and how the student solves the exercises with the help and support of the intelligent tutoring system.


```
Division ([
    {"divisor":9,"dividend":[1,0,8],"quotient":[0,1,2],"reminder":[1,0],"mul":[9,18]},
    {"divisor":2,"dividend": [4,2],"quotient":[2,1],"reminder":[0,0],"mul":[4,2]},
    {"divisor":11,"dividend":[1,0,0],"quotient":[0,0,9],"reminder":[1],"mul":[99]},
    {"divisor":10,"dividend":[5,0,0],"quotient":[0,5,0],"reminder":[0,0],"mul":[50,0]},
    {"divisor":20,"dividend":[5,0,2,0],"quotient":[0,2,5,1],"reminder":[10,2,0],"mul":[40,100,20]},
    {"divisor":14,"dividend":[1,3,2],"quotient":[0,0,9],"reminder":[6],"mul":[126]}
    ]);
```


The basic structure of the XML-based file consists of an array of objects which contains the "divisor" and "dividend" attributes which are shown to the student. The "quotient", "reminder" and "mul" attributes contain the correct answers.

An initial exercise is presented to the student through the interface; students can enter answers they think are correct, while the intelligent tutor dynamically check the corresponding XML file to verify the answer and to provide responses to them. The initial exercise has a difficulty level that was set for each student profile according the result in the diagnostic test completed by the own student. The difficulty level of the next exercises can be modified depending on the student's performance in solving each math exercise. The functionality of how responses are evaluated and the path considered by the solution process are shown in Figure 5. In this context, the premise is simple. The ITS waits for an entry value *t*, and verifies that the value is correct. When a correct value is entered, the ITS moves to the next box; then it will wait for the next input value. Otherwise, the ITS sends a message through a pedagogical agent about the type of error found in the answer and then it waits for a student response. This action is repeated until the division is complete. During this process the student can make use of two buttons located below the division operation. The "Help" button sends tips or advices to the student through the pedagogical agent. The "Next" button moves to the next exercise.

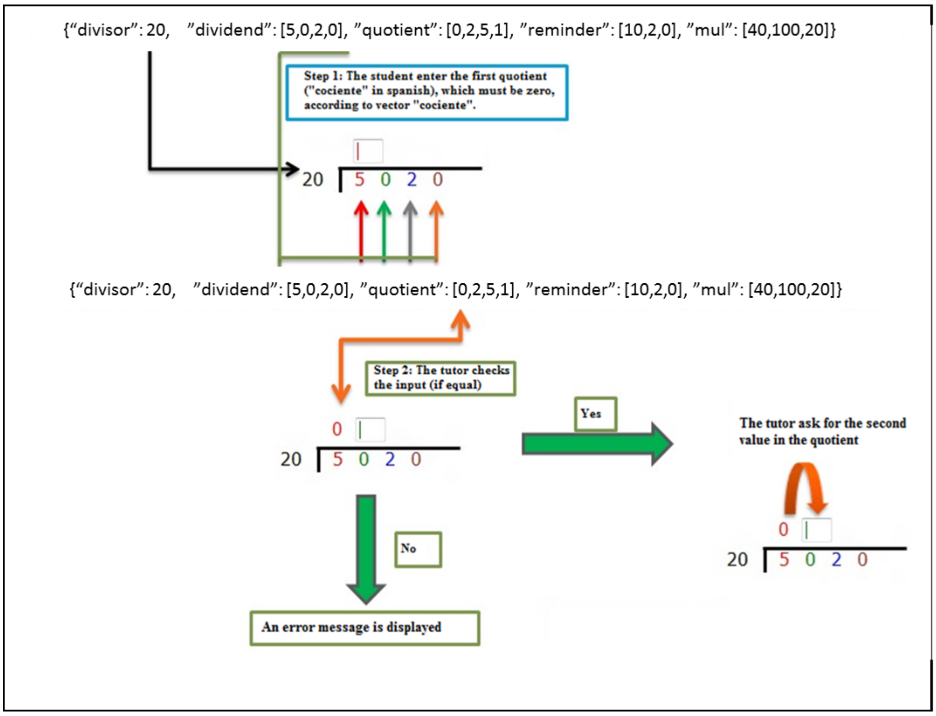

Fig. 5. Evaluation of an example of the Division Arithmetic Operation

4 Results and Conclusions

We still are working with the integration of the neural networks to the ITS. We have evaluated the ITS (with no emotion and learning style recognition) with a group of children from third grade. There were 72 children from public and private schools who tested the tool and the tutoring system. We evaluated the subject of multiplication. We applied a test before and after the students used the software tool. We obtained from the results a good improvement in most students (more in students with lower initial grades) using one of the two teaching methods for multiplication: traditional and lattice.

The results up to now are encouraging. The next step is to integrate the neural networks with the ITS. We also are integrating as an application, the affective ITS into the Facebook social network. The intelligent tutoring system and recognizers were implemented by using different software tools and programming languages.

References

1. Arroyo, I., Woolf, B., Cooper, D., Burleson, W., Muldner, K., Christopherson, R.: Emotions sensors go to school. In: Proceedings of the 14th international conference on artificial intelligence in education. pp. 17−24. IOS press, Amsterdam (2009)
2. Conati C., Maclaren, H.: Empirically Building and Evaluating a Probabilistic Model of User Affect. User Modeling and User-Adapted Interaction, 19, 267−303 (2009)
3. D'Mello, S.K., Picard, R.W., Graesser, A.C.: Towards an affective-sensitive AutoTutor. Special issue on Intelligent Educational Systems IEEE Intelligent Systems, 22(4) 53−61 (2007)
4. Du Boulay, B.: Towards a motivationally intelligence pedagogy: how should an intelligent tutor respond to the unmotivated or the demotivated? In: R. Calvo & S. D'Mello (Eds.) New perspectives on affect and learning technologies, New York: Springer, pp. 41−52 (2011)
5. Ekman, P., Friesen, W.: Unmasking the face: a guide to recognizing emotions from facial clues. Englewood Cliffs, NJ: Prentice-Hall (1975)
6. Forbes-Riley, K. & Litman, D.: Adapting to Student Uncertainty Improves Tutoring Dialogues. In: Proceedings of the 14th International Conference on Artificial Intelligence in Education (AIED), pp. 33−40, Brighton, UK (2009)
7. D'Mello, S.K., Graesser, A.C.: AutoTutor and Affective AutoTutor: Learning by Talking with Cognitively and Emotionally Intelligent Computers that Talk Back. ACM Transactions on Interactive Intelligent Systems, 2(4), 23:2-23:39 (2012)
8. Langner, O., Dotsch, R., Bijlstra, G., Wigboldus, D., Hawk, S., van Knippenberg, A.: Presentation and validation of the Radboud Faces Database. Cognition & Emotion, 24(8), 1377−1388, DOI: 10.1080/02699930903485076 (2010)