A Fuzzy-Neural Network for Classifying Learning Styles in a Web 2.0 and Mobile Learning Environment

Ramón Zatarain Cabada, María Lucía Barrón Estrada, Rosalío Zatarain Cabada
Departamento de Sistemas y Computación
Instituto Tecnológico de Culiacán
Culiacán, México
rzatarain@itculiacan.edu.mx

Carlos Alberto Reyes García
Coordinación de Ciencias Computacionales
Instituto Nacional de Astrofísica, Optica y Electrónica
Tonantzintla, México
kargaxxi@inaoep.mx

Abstract— In this paper, we present a new method for classifying student learning styles using a previously trained Fuzzy-Neural Network. We apply this method with a software tool used to produce learning objects for handheld computers like PDAs or Cell Phones, everything under a Web 2.0 collaborative learning environment. We trained the network with three different courses following Felder-Silverman model of Learning Styles. Then we add the network to an interpreter used to display learning material in Mobile devices. The interpreter show not only course material but also tracking information about learning styles used for each user or student. We present different experiments made to the network along with the tool.

Keywords-component; Web 2.0 education; Mobile Learning; Learning Styles; Fuzzy-Neural Networks

I. INTRODUCTION

An artificial neural network is a model of reasoning used to solve complex problems. There are many types of neural networks and they are divided into supervised and unsupervised neural networks depending on the way they are trained to learn. The fields or problem application where a neural network can be applied is abundant. They include areas like medicine, process control, finances, engineering, etc. [1]. However, we could not find neural network applications for learning style classification.

On the other hand, tailoring learning material to a community of learners for mobile devices like PDAs or Cell Phones is a new topic. Most work in the field of adaptive Learning Systems is oriented only to individual learners. But today, learning applications need to be centered on a community of learners, such as modern Web 2.0 applications [2]. The main benefit of using this new approach in education is collecting the information and ideas to create the store of the courses from the whole community of users (including students and instructors). Another benefit is taking advantage of technologies (wikis, blogs and social networking) that young learners are using in their spare time.

In this context, we propose a new method to create adaptive learning material in a Web 2.0 collaborative learning environment for handheld devices. The method consists of different steps where a fuzzy-neural network is trained with different courses under different learning styles, and later exported to a mobile device together with an interpreter, and the learning objects. The main contribution of this paper is the implementation of a new methodology for producing adaptive learning material for a mobile-collaborative learner community using intelligent system techniques.

The paper’s organization is as follows: In Section 2, we describe the overall process of our method. In Section 3 we present the Intelligent Module Design. Section 4 explains course displaying in mobile devices. Tests and results are shown in Section 5. Comparison to related work is given in section 6 and conclusions and future work are discussed in Section 7.
**interpreter** is added to the course. This interpreter has the task of displaying the material of the course into the mobile device according with the learning style of the learner.

The student or learner is an important author of the course and participate actively adding learning resources to the courses. The learner has a dynamic user profile with information like the GPA, particular learning style, or resources added by the user to the course.

Once a course is created, the Course Publication Module saves it into a Course Repository. Whenever a learner accesses a course, a recommender system implemented in our system tool presents interesting links or Web sites with learning material related to a current topic. Such material can be stored in the resource repository, which was searched previously by using Web mining techniques implemented also in our system.

### III. INTELLIGENT MODULE

The Intelligent Module takes as an input the learning material for four different learning styles according to Felder-Silverman Model of Learning. Then, it creates an adaptive course formed with two components: A XML file (it contains the learning material), and an Interpreter, used to display the course into mobile devices. The Interpreter also applies a previously trained Neuro-Fuzzy Network (NFn) as a dynamic classifier to show the learning material according to the best learning style of the user. The NFn acts as a planner which produces a dynamic sequence of the learning objects or material presentation.

Figure 2 shows part of the NFn [1] exported into the Interpreter. The first layer has 7 neurons representing seven fuzzy variables (time spent in a topic, student grade, etc.) used in the classification of the learning style. Every neuron of layer 1 is connected to 3 neurons of layer 2 (fuzzyfication layer). We are using triangular membership functions. Then the activation function for the layer-2 neurons is as follow:

\[
y_i(1) = \begin{cases} 
0, & \text{if } x_i(1) \leq a - b / 2 \\
1 - 2|x_i(1) - a| / b, & \text{if } a-b/2 < x_i(1) < a+b/2 \\
0, & \text{if } x_i(1) \geq a+b/2 
\end{cases}
\]

where \(a\) and \(b\) are centre and width of the triangle, \(x_i(1)\) and \(y_i(1)\) are input and output of neuron \(i\).

![Figure 2. Part of the Neuro-Fuzzy Network](image)

The output of layer 3 represents the strength of each one of the fuzzy rules. The best weight values between layer 3 and layer 4 are calculated using a genetic algorithm later explained. Layer 5 is the output of the NFn. We use a **Centroid** technique to make defuzzyfication. The value of the output is the learning style for the current user of the course.

#### A. Training the NFn with a Genetic Algorithm

At the beginning, we create a chromosome population with random values. Each decoded element of the population, represent the weights in the NFn (see figure 3).

![Figure 3. Decoding the weights in the NFn into a chromosome](image)

For encoding the chromosome, the weights of the NFn are sorted using a Bucket Sort Algorithm, ordering first by layer, and then by neurons. The output of the algorithm is the chromosome. Each gene of the chromosome represents a weighted link in the NFn.

To evaluate the chromosome’s performance, we assign each weight (gene) contained in the chromosome to the links of a respective NFn. Then we test the network with a set of training values. Last, the sum of squared errors is evaluated.
We always try to find the smaller sum. This technique gives the best chromosome which represents the best weights in a particular NFN.

B. The NFN training Data

In order to train the network, we create three sets of courses for high school students. Each course was presented in eight different styles for Felder-Silverman model. The chosen courses were Teaching Digital Photography, Eolic Energy, and Introduction to Computers. In order to get their best learning style we applied Felder-Silverman Index of Learning Style Questionnaire to every student. We also designed an exam to evaluate the course taken by the student. The input data to the network was the performance of the student under each course and the learning style of the taken course (randomly chosen). The Desired Output was the learning style calculated from Felder-Silverman Questionnaire. We made tests with two groups of 40 students.

IV. USING THE NFN ON MOBILE DEVICES

Whenever a course is created and exported to a cell phone or PDA (using XML format), the NFN is also exported to those devices, along with a XML parser or interpreter. This program run any course stored in the mobile by reading the XML file where the course is stored. At the beginning, a learning style is assigned to the student or user of the course. Then, depending on the results from questions prepared to the students inside the learning objects, the learning material is adapted to the best learning style (Visual, Verbal, Sequential, etc.) of the student (the NFN is consulted in order to find the best appropriated learning style to the student).

On the other hand, usability standards in desktop computers are different from handheld devices like PDAs or cell phones. Some of the usability problems found with respect to mobility are: small screen size, limited memory space, software navigation, and operating system compatibility. We followed the standards proposed in [5], and implemented new interfaces for visualization of the courses on mobile devices like cell phones. Figure 4 shows two examples with the old and new version.

An important feature is the possibility of tracing the different learning styles that a user or student follow during lesson learning. The interpreter of the course optionally displays the current student learning styles plus the values of fuzzy variables. This information is relevant for doing different types of analysis with respect to behaviors of the students and the way they learn. Figure 5 shows two snapshots of a handheld device (a cell phone) showing values of each learning style in a range between -100 to +100 (left device) and a dynamic sequence of the style behavior during the execution of the course (right device).

V. TEST AND RESULTS

We tested the tool with 15 professors/teachers and their respective students of different teaching levels. They developed different kinds of courses like a GNU/Linux course, a Basic Math Operation course, and learning material for preparation to the University Admission test EXANI-II. Each one of the courses had from 4 to 10 units of learning. They included evaluations on each unit. The students participated by reading, evaluating and adding material (Web resources) to the courses.

Next, we present an example of how an author creates/updates learning material for a Basic Math course (figure 6).
We first create the structure of the course (top-left) with units, sub-units, and quizzes. Then, we add learning material for each learning style defined in the model (top-right is verbal style and bottom-left is visual style). In this moment, we also define fuzzy sets to each fuzzy variable (center of figure), and recommended and current resources in the course (bottom-left). Last, we export the course in one of two formats: SCORM (for e-learning environments) or MLT (for m-learning environments). Last, the learner in a mobile can rate the material he is reading. This is important for recommending learning material be added to some course.

VI. RELATED WORK

There are software tools used to create collaborative and web 2.0 learning systems like SHAREK [6], and SPRITS [7]. SHAREK allows community learners to share resources for creating learning material. However the learning material is oriented to e-learning systems with no concerns to adaptation possibilities. On the other hand, SPRITS is more oriented to sharing knowledge. Khribi, Jenni, and Nasraoui present an approach of an automatic recommendation system for personalized e-learning [8]. With respect to Identifying learning styles there have been different approaches like Bayesian networks, decision trees, or Hidden Markov Models. However, the learning styles are calculated based only on the ILSQ questionnaire and none of those works are authoring tools [9, 10, 11]. The main advantages of our tool are dynamic adaptation to student style of learning, and possibilities of creating m-learning and e-learning material everything under an integrated environment.

VII. CONCLUSIONS AND FUTURE WORK

The work presented in this paper provides different aspects of our research. First, we illustrate an original architecture of the tool oriented to the Web 2.0 technology. Second, we discuss a novel technique for managing different styles of learning. This technique use a genetic algorithm for training a NFN used to classify learning styles for Felder-Silverman models. Third, we talk about different points of the visualization on mobile devices, like usability and traceability of the information. Last, we show an exercise of creating learning material using our tool.

We are now committed to test the system with unsupervised neural networks. The reasons are related with providing less assistance of teachers in the network training.
On the other hand, we also are working with some aspects of usability and mobile device adaptation.

ACKNOWLEDGMENT

The work described in this paper is fully supported by a grant from the DGEST (Dirección General de Educación Superior Tecnológica) in México under the program “support and development of academic bodies” [Academic Body: Research in Software Engineering].

REFERENCES