Design a personalized e-learning system based on item response theory and artificial neural network approach

Ahmad Baylari, Gh.A. Montazer *

IT Engineering Department, School of Engineering, Tarbiat Modares University, Tehran, Iran

ARTICLE INFO

Keywords:
e-Learning
Adaptive testing
Multi-agent system
Artificial neural network (ANN)
Item response theory (IRT)

ABSTRACT

In web-based educational systems the structure of learning domain and content are usually presented in the static way, without taking into account the learners' goals, their experiences, their existing knowledge, their ability (known as insufficient flexibility), and without interactivity (means there is less opportunity for receiving instant responses or feedbacks from the instructor when learners need support). Therefore, considering personalization and interactivity will increase the quality of learning. In the other side, among numerous components of e-learning, assessment is an important part. Generally, the process of instruction completes with the assessment and it is used to evaluate learners' learning efficiency, skill and knowledge. But in web-based educational systems there is less attention on adaptive and personalized assessment. Having considered the importance of tests, this paper proposes a personalized multi-agent e-learning system based on item response theory (IRT) and artificial neural network (ANN) which presents adaptive tests (based on IRT) and personalized recommendations (based on ANN). These agents add adaptivity and interactivity to the learning environment and act as a human instructor which guides the learners in a friendly and personalized teaching environment.

1. Introduction

The recent applications of information communication technologies (ICT) have a strong and social impact on the society and the daily life. One of the aspects of society that has been transforming is the way of learning and teaching. In the recent years, we have seen exponential growth of Internet-based learning. The transition to online technologies in education provides the opportunities to use new learning methodology and more effective methods of teaching (Georgieva, Todorov, & Smrikarov, 2003). In a simple definition e-learning is defined as the use of network technology, namely the Internet, to design, deliver, select, administer and extend learning (Hamdi, 2007). Important features of this form of learning are the separation of learner and teacher and can take place anywhere, at any time and at any pace. Thus e-learning can take place at people's work or at home, at the time available (Kabassi & Virvou, 2004). The other perspectives of using e-learning can be generalized as follows: an opportunity for overcoming the limitations of traditional learning, such as large distance, time, budget or busy program; equal opportunities for getting education no matter where you live, how old you are, what your health and social status is; better quality and a variety of lecture materials; new consortia of educational institutions, where a lot of specialists work in collaboration, use shared resources and the students get freedom to receive knowledge, skills and experience from other universities (Georgieva et al., 2003). Due to the flexibilities that mentioned above many universities, corporations and educational organizations are developing e-learning programs to provide course materials for web-based learning. Also e-learning can be used for online employee training in business (Chen, Lee, & Chen, 2005).

From another point of view, numerous Web applications, such as portal websites (such as Google and Yahoo!), news websites, various commercial websites (such as Amazon and eBay) and search engines (such as Google) have provided personalized mechanisms to enable users to filter out uninteresting or irrelevant information. Restated, personalized services have received considerable attention recently because of information needs are different among users (Chen et al., 2005). But in web-based educational systems the structure of the domain and the content are usually presented in the static way, without taking into account the learners' goals, their experiences, their existing knowledge and their abilities (Huang, Huang, & Chen, 2007) also known as insufficient flexibility (Xu & Wang, 2006), and without interactivity means there is less opportunity for receiving instant responses and feedback from the instructor when online learners need support (Xu & Wang, 2006). Therefore, adding interactivity and intelligence to Web educational applications is considered to
be an important direction of research (Hamdi, 2007). Personalization is an issue that needs further attention, especially when it comes to web-based instruction, where the learners’ population is usually characterized by considerable heterogeneity with respect to background knowledge, age, experiences, cultural backgrounds, professions, motivation, and goals, and where learners take the main responsibility for their own learning (Huang et al., 2007).

The personalization can ensure that the system will take into account the particular strengths and weaknesses of each individual who is using the program. It is simple logic that response personalized to a particular student must be based on some information about that student; in the area of artificial intelligence (AI) in education, this realization led to student modeling, which became a core or even defining issue for the field. As a result, student modeling is used in intelligent tutoring systems (ITSs), intelligent learning environments (ILEs) and adaptive hypermedia systems (AHS) to achieve personalization and dynamic adaptation to the needs of each individual student (Kabassi & Virvou, 2004). In the last 10 years, the field of ITS research has been developing rapidly. Along with the growth of computing capabilities, more and more ITS researchers have focused on personalized virtual learning environments (PVLEs) to provide tailored learning materials, instructions and instant interaction to suit individual learners or a group of learners by using intelligent agent technology. Intelligent agent technologies facilitate the interaction between the students and the systems, and also generate the artificial intelligence model of learning, pattern recognition, and simulation, such as the student model, task model, pedagogical model, and repository technology. These models work together in a productive way to support students’ learning activities adaptively. Therefore, the properties of intelligent agents, i.e., autonomy, pre-activity, pro-activity and cooperation, support PVLEs in recognizing online learners’ learning stage and in reacting with tailored instruction including personalized learning materials, tests, instant interactions, etc. (Xu & Wang, 2006).

In this study, we will propose a framework for constructing adaptive tests that will be used as a post-test in our system. Thus a multi-agent system is proposed which has the capability of estimating the learners’ ability based on explicit responses on these tests and presents him/her a personalized and adaptive test based on that ability. Also the system can discover learner’s learning problems via learner responses on review tests by using an artificial neural network (ANN) approach and then recommends appropriate learning materials to the student. So, the organization of this paper is as follows. The following section briefly reviews the relevant literature on personalization for e-learning. After describing item response theory (IRT) in Section 3, the test construction process will be presented in Section 4. Section 5 describes the system architecture. Agent design and system evaluation are addressed in Sections 6 and 7, respectively. The final section provides a conclusion for our work.

2. Personalization concept in e-learning systems

Many researchers have recently endeavored to provide personalization mechanisms for web-based learning (Chen et al., 2005). Therefore, personalized learning strategy is needed for most e-learning systems currently. Learners enjoyed greater success in learning environments that adapted to and supported their individual learning orientation (Xu & Wang, 2006). Nowadays, most recommendation systems consider learner/user preferences, interests, and browsing behaviors when analyzing learner/user behaviors for personalized services. These systems neglect the importance of learner/user ability for implementing personalized mechanisms. On the other hand, some researchers emphasized that personalization should consider different levels of learner/user knowledge, especially in relation to learning. Therefore, considering learner ability can promote personalized learning performance (Chen et al., 2005). Also most e-learning systems lack the presence and control of an instructor to the learning process to assist and help learners in these environments (also known as human touch). Thus modeling the behavior of the instructor and provide instant feedbacks is an important task in these environments. Shi et al. have proposed a method to personalize the learning process by a SOM agent which replaces the human instructor and controls in an e-learning environment (Shi, Revithis, & Chen, 2002). Xu and Wang developed a multi-agent architecture for their personalized model (Xu & Wang, 2006). Chen et al. proposed a personalized e-learning system based on IRT (PEL-IRT) which considers both course material’s difficulty and learner’s ability to provide individual learning paths for learners (Chen et al., 2005). So far he and his colleagues have presented a prototype of personalized web-based instruction system (PWIS) based on the modified IRT to perform personalized curriculum sequencing through simultaneously considering courseware difficulty level, learner’s ability and the concept continuity of learning pathways during learning (Chen, Liu, & Chang, 2006).

In all systems described above, the importance of tests, constructing and adapting them to learner’s ability have been neglected. They have only adapted the content and personalized part of their systems is sequence and organization of the content. So in this paper, designing the tests and adapting these tests to learners and simulating an instructor that control the learning process is desired. To achieve these goals, a multi-agent system will be proposed. Theses agents play an important role to provide personalization for the system. Agents are a natural extension of current component-based approaches and should at least be able to model the preferences, goals, or desires of their owners and to learn as they perform their assigned tasks (Xu & Wang, 2006). In our system, we have three tests: pre-test, adaptive post-tests and review tests. Post-tests are adaptive and will be adapted to the learner’s ability. By review tests the instructor can diagnose learner’s learning problems and will recommend personalized learning materials to the learner. Nice features of our system are dynamic and personalized tests and personalized recommendations of a simulated human instructor.

3. Item response theory (IRT)

Item response theory (IRT) was first introduced to provide a formal approach to adaptive testing (Fernandez, 2003). The main purpose of IRT is to estimate an examinee’s ability (θ) or proficiency (Wainer, 1990) according to his/her dichotomous responses (true/false) to test items. Based on the IRT model, the relationship between examinee’s responses and test items can be explained by so-called item characteristic curve (ICC) (Wang, 2006). In the case of a typical test item, this curve is S-shaped (as shown in Fig. 1); the horizontal axis is ability scale (in a limited range) and the vertical axis is the probability that an examinee with certain ability will give a correct answer to the item (this probability will be smaller for examinees of low ability and larger for examinees of high ability). The item characteristic curve is the basic building block of item response theory; all the other constructs of the theory depend upon this curve (Baker, 2001).

Several nice features of IRT include the examinee group invariance of item parameters and item invariance of an examinee’s ability estimate (Wang, 2006). Under item response theory, the standard mathematical model for the item characteristic curve is the cumulative form of the lo-
logistic function. It was first used as a model for the item characteristic curve in the late 1950s and, because of its simplicity, has become the preferred model (Baker, 2001).

Based on the number of parameters in logistic function there are three common models for ICC; one parameter logistic model (1PL) or Rasch model, two parameter logistic model (2PL) and three parameter (3PL) (Baker, 2001; Wang, 2006). In the 1PL model, each item is characterized by only one parameter, the item difficulty $b_i$, in a logistic formation as shown

$$P_i(\theta) = \frac{1}{1 + \exp(-D(\theta - b_i))},$$  \hspace{1cm} (1)

where $D$ is a constant and equals to 1.7 and $\theta$ is ability scale. In the 2PL model, another parameter, called discrimination degree $a_i$, is added into the item characteristic function, as shown

$$P_i(\theta) = \frac{1}{1 + \exp(-a_iD(\theta - b_i))}.$$  \hspace{1cm} (2)

The last 3PL model adds a guess degree $c_i$ to the 2PL model, as shown in Eq. (3), modeling the potential guess behavior of examinees (Wang, 2006).

$$P_i(\theta) = c_i + \frac{1}{1 - c_i} \cdot \frac{1}{1 + \exp(-a_iD(\theta - b_i))}. \hspace{1cm} (3)$$

Several assumptions must be met before reasonable and precise interpretations based on IRT can be made. The first is the assumption of unidimensionality, which assumes there is only one factor affecting the test performance. The second assumption is the local independence of items, which assumes test items are independent to each other. This assumption enables an estimation method called maximum likelihood estimator (MLE) to effectively estimate item parameters and examinee's ability (Wang, 2006).

$$L(\theta | u_1, u_2, \ldots, u_n) = \prod_{i=1}^{n} P_i(\theta)^{u_i}(1 - P_i(\theta))^{1-u_i},$$ \hspace{1cm} (4)

where $Q_i(\theta) = 1 - P_i(\theta)$. $P_i(\theta)$ denotes the probability that learner can answer the $i$th item correctly, $Q_i(\theta)$ represents the probability that learner cannot answer the $i$th item correctly, and $u_i$ is 1 for correct answer to item $i$ and 0 for incorrect answer to item $i$ (Wainer, 1990). Since $P_i(\theta)$ and $Q_i(\theta)$ are functions of learner ability $\theta$ and item parameters, the likelihood function is also a function of these parameters. Learner ability $\theta$ can be estimated by computing the maximum value of likelihood function. Restated, learner ability equals the $\theta$ value with maximum value of likelihood function (Chen et al., 2005).

Item information function (IIF) in IRT plays an important role in constructing tests for examinees and evaluation of items in a test. Any item in a test provides some information about the ability of the examinee, but the amount of this information depends on how closely the difficulty of the item matches the ability of the person. The amount of information, based upon a single item, can be computed at any ability level and is denoted by $I_i(\theta)$, where $i$ is the number of the items. Because only a single item is involved, the amount of information at any point on the ability scale is going to be rather small (Baker, 2001). If the amount of item information is plotted against ability, the result is a graph of the item information function such as shown in Fig. 2.

As shown in this figure it can be seen that an item measures ability with greatest precision at the ability level corresponding to the item's difficulty parameter. The amount of item information decreases as the ability level departs from the item difficulty and approaches zero at the extremes of the ability scale (Baker, 2001).

Item information function is defined

$$I_i(\theta) = \frac{P_i'(\theta)}{P_i(\theta)Q_i(\theta)}, \hspace{1cm} (5)$$

where $P_i(\theta)$ is the first derivative of $P_i(\theta)$ and $Q_i(\theta) = 1 - P_i(\theta)$. A test is a set of items; therefore, the test information at a given ability level is simply the sum of the item information at that level. Consequently, the test information function (TIF) is defined as:

$$I(\theta) = \sum_{i=1}^{N} I_i(\theta), \hspace{1cm} (6)$$

where $I_i(\theta)$ is the amount of information for item $i$ at ability level $\theta$ and $N$ is the number of items in the test. The general level of the test information function will be much higher than that for a single item information function. Thus, a test measures ability more precisely than does a single item. An important feature of the definition of test information given in Eq. (6) is that the more items in the test, the greater the amount of information. Thus, in general, longer tests will measure an examinee’s ability with greater precision than will shorter tests (Baker, 2001).

Item response theory usually is applied in the computerized adaptive test (CAT) domain to select the most appropriate items for examinees based on individual ability. The CAT not only can efficiently shorten the testing time and the number of testing items but also can allow finer diagnosis at a higher level of resolution. Presently, the concept of CAT has been successfully applied to replace traditional measurement instruments (which are typically fixed-length, fixed-content and paper–pencil tests) in several real-world applications, such as GMAT, GRE, and TOEFL (Chen et al., 2006).
In this paper, we will use IRT-3PL model to test construction, ability estimation and appropriate post-test selection for learners. In the following section we will describe test construction process.

4. Test construction process

We have three types of tests in our system; pre-test, post-test and review tests. In this section construction process of these three types will be described. All of these tests have 10 items. For post-test construction, as shown in Fig. 3, at the first step the teacher analyzes the learning contents based on learning objectives and designs suitable multiple choice items from each learning object (LO). Every item has its unique code and is stored in testing items database. Then these items were presented to students to answer them. Having collected their responses, according to IRT their testing data were analyzed by the BILOG program to obtain the appropriate item parameters under 3PL model. Therefore, each item has its own $a$, $b$ and $c$ parameters and we will have a calibrated item bank, which can be used for item selection in CAT and test construction. Afterwards we designed a set of adaptive tests for our system. At this step, the teacher constructs a few appropriate tests at each ability level. For example let $\theta = 0.5$ and the items of 3rd session have been ranked based on their item information function. Then the teacher selects 10 items from top to down and constructs the desired test. The selection of items is based on their information and their content. These tests are constructed for each ability scale. Then they were stored with their item codes, item parameters and the session number in the test database. This process was repeated for each session. These tests are adaptive and will be used as post-test in the system.
To construct review tests, the learning contents were analyzed and the learning concepts were extracted. Then appropriate tests were designed. They are fixed and static in the learning process and can be considered as the teacher’s expectation from learners. They are used to diagnose the learner’s learning problems. And at last we need pre-test at the entrance of each session. Theses tests are static too, and are stored in our test database. Finally, we have a test database which may be used for assessment of the learners in our system.

5. System architecture

Integrating multiple intelligent agents into distance learning environments may help bring the benefits of the supportive classroom closer to distance learners, therefore, intelligent agents are becoming more and more hot in ITS study (Zhou, Wu, & Zhang, 2005). Also, the agent metaphor provides a way to operate and simulate the ‘human’ aspect of instruction in a more natural and valid way than other controlled computer-based methods (Xu & Wang, 2006). Therefore, the proposed architecture is a multi-agent system which will be used for personalization of the learning environment. As shown in Fig. 4, the proposed architecture is a three-layer architecture. The middle layer contains four agents: activity agent, test agent, planning agent and remediation agent.

- **Activity agent**: This agent records e-learning activities, online learners’ learning activities (such as mouse action) learning duration on a particular task, documents load/unload, etc. and stores them in the learner’s profile.

- **Planning agent**: This agent plans the learning process. At start of each session, this agent asks the learner whether he/she is familiar with the session. If the answer is ‘no’ the agent permits the learner to enter the session, but if the answer is ‘yes’ then this agent requests the test agent to present him/her a pre-test. Based on the learner’s responses on this test, the agent can decide which part of the next session does not need to be presented to the learner. At the end of the session this agent requests the test agent to present him/her the appropriate post-test. After some sessions e.g. three sessions this agent asks from test agent to present the review test to the learner and then presents their responses to the remediation agent.

- **Test agent**: This agent based on the requests of planning agent, presents appropriate test type to the learner. So this agent selects the suitable test from test database and presents it to the learner. Also, in the case of post-test, this agent extracts learner’s ability from his/her profile and presents the most appropriate post-test to the learner based on his/her ability, then estimates learner’s ability in this post-test and updates it in the profile.

- **Remediation agent**: This agent analyzes the results of review tests, and diagnoses learner’s learning problems, like a human instructor, and then recommends the appropriate learning materials to the learner.

Therefore, these agents cooperate with each other to help and assist learners in the learning process, like a human instructor, and increase the quality and effectiveness of learning.

The lower layer is the repository layer and contains learner profile database which stores user profile, learning objects database which store learning materials as learning objects and test database which has been designed in the previous section. The system operates as follows:

At first, the planning agent asks the learner whether he/she is familiar with the first session. If the answer is positive this agent asks test agent to present him/her a pre-test. Having analyzed the learner’s responses, this agent presents the appropriate first session LOs. Otherwise, this agent presents all first session LOs. At the end of this session test agent presents the appropriate post-test with medium difficulty based on IRT. This is because there is no information about the learner ability at the first session. At the end of each session test agent presents the appropriate post-test from the test database that matches learner’s ability. At the end of the third session (for example), test agent presents the review test to the learner to diagnose his/her learning problems. When the learner finishes, the remediation agent analyzes the responses and after diagnosing learner’s learning problems, recommends the appropriate LOs. So, the learner will be guided to a remediation session to improve his/her learning problems. Then test agent presents a post-test and estimates learner’s ability and updates it in his/her profile. The test agent receives learner’s responses in each post-test and estimates his/her ability in each test by applying maximum likelihood estimator and stores it in his/her profile. In the next session post-test this agent selects and presents most appropriate post-tests with respect to learner’s ability in his/her profile. A post-test with a maximum information function value under learner with ability \( \theta \) is presented to the learner. Restated information of all test items in the learner’s ability are calculated and summed. It must be noted that a test with a maximum information function value under learner’s ability \( \theta \) has the highest recommendation priority.

6. System design and development

After designing the review tests, they were presented to some students to collect their responses. Then their tests and responses were presented to the instructor so he could diagnose their learning problems and recommend them appropriate learning materials. Learning materials are in the form of LOs. The maximum number of recommended LOs were five. Due to the large number of responding states to a test \((2^{10})\) states for each test), an artificial neural network was used for recommending remaining states. The items of the test and the students’ responses have been considered as the inputs of the network and the recommendations as the output of the one. The network may be trained with these data; then the trained network will recommend appropriate learning materials instead of human instructor in the learning environment.

In order to experiment the remediation agent, “Essentials of information technology management” course was chosen. This course was divided into several LOs and a few codes were allocated for all LOs. The teacher designed the items from each LO and these items had their unique code, too. Then five new tests were designed from these items. They were presented to the students and their responses were collected. The number of responses was 200 per each test. So there were 1000 responses totally. These items and their responses were presented to the instructor. Table 1 shows a sample items and responses.

As shown, this table has 20 columns, first 10 columns from left (I1 to I10) are item codes and latter 10 columns (R1 to R10) are corresponding responses which coded 1 for correct response and 0 for incorrect response. These data were presented to the instructor, and then he analyzed each item response pair and having diagnosed learner’s learning problems, recommended the suitable LOs to the students. The recommended LOs for the item response pairs in Table 1 have been shown in Table 2.

Thus Table 1 considers the input data to the neural network and Table 2 the output ones. But in order to use these data to train the network, a preprocessing part should be used. The details of the neural network development process will be described in the subsequent sub-sections.
6.1. The artificial neural network

A back-propagation network was used to learn from the data. These networks are the most widely used type of networks and are considered the workhorse of ANNs (Basheer & Hajmeer, 2000). A back-propagation network (see Fig. 5) is a fully connected, layered, feed-forward neural network. Activation of the network flows in one direction only: from the input layer through the hidden layer, then on to the output layer. Each unit in a layer is connected in the forward direction to every unit in the next layer. A back-propagation network may contain multiple hidden layers. Knowledge of the network is encoded in the (synaptic) weights between units. The activation levels of the units in the output layer determine the output of the whole network (Hamdi, 2007). This network can learn the mapping from one data space to another using examples, and also has a high generalization capability. The used network has twenty input nodes and five output neurons. Ten of these inputs are item codes and the others are responses, and the output neurons are recommended LOs.

6.2. Data normalization and partitioning

Normalization of data within a uniform range (e.g., 0–1) is essential to prevent larger numbers from overriding smaller ones, and to prevent premature saturation of hidden nodes, which impedes the learning process. There is no one standard procedure

Table 1
Items responses data.

<table>
<thead>
<tr>
<th>I1</th>
<th>I2</th>
<th>I3</th>
<th>I4</th>
<th>I5</th>
<th>I6</th>
<th>I7</th>
<th>I8</th>
<th>I9</th>
<th>I10</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
<th>R10</th>
</tr>
</thead>
<tbody>
<tr>
<td>121</td>
<td>122</td>
<td>123</td>
<td>166</td>
<td>167</td>
<td>193</td>
<td>194</td>
<td>195</td>
<td>244</td>
<td>245</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>62</td>
<td>111</td>
<td>112</td>
<td>166</td>
<td>167</td>
<td>201</td>
<td>202</td>
<td>224</td>
<td>225</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>166</td>
<td>167</td>
<td>171</td>
<td>172</td>
<td>196</td>
<td>197</td>
<td>204</td>
<td>205</td>
<td>234</td>
<td>235</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>76</td>
<td>77</td>
<td>78</td>
<td>136</td>
<td>137</td>
<td>183</td>
<td>184</td>
<td>185</td>
<td>246</td>
<td>247</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>121</td>
<td>122</td>
<td>123</td>
<td>166</td>
<td>167</td>
<td>193</td>
<td>194</td>
<td>195</td>
<td>244</td>
<td>245</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>76</td>
<td>77</td>
<td>78</td>
<td>136</td>
<td>137</td>
<td>183</td>
<td>184</td>
<td>185</td>
<td>246</td>
<td>247</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>166</td>
<td>167</td>
<td>171</td>
<td>172</td>
<td>196</td>
<td>197</td>
<td>204</td>
<td>205</td>
<td>234</td>
<td>235</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>121</td>
<td>122</td>
<td>123</td>
<td>166</td>
<td>167</td>
<td>193</td>
<td>194</td>
<td>195</td>
<td>244</td>
<td>245</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>62</td>
<td>111</td>
<td>112</td>
<td>166</td>
<td>167</td>
<td>201</td>
<td>202</td>
<td>224</td>
<td>225</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Table 2
Recommended LOs.

<table>
<thead>
<tr>
<th>LO1</th>
<th>LO2</th>
<th>LO3</th>
<th>LO4</th>
<th>LO5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>7</td>
<td>29</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>11</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>19</td>
<td>10</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>15</td>
<td>12</td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>19</td>
<td>16</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>13</td>
<td>0</td>
<td>8</td>
<td>0</td>
</tr>
</tbody>
</table>

Fig. 5. A multi-layer back-propagation network.

Fig. 6. Learning curve for a network with 15 neurons in the hidden layer.

Fig. 7. Learning curve for a network with 15 neurons in the first hidden layer and 10 neurons in the second hidden layer.
for normalizing inputs and outputs. One way is to scale input and output variables \((x_i)\) in interval \([\lambda_1, \lambda_2]\) corresponding to the range of the transfer function:

\[
X_n = \lambda_1 + (\lambda_2 - \lambda_1) \left( \frac{x_i - x_{i_{\text{min}}}}{x_{i_{\text{max}}} - x_{i_{\text{min}}}} \right),
\]

where \(X_n\) is the normalized value of \(x_i\), \(x_{i_{\text{min}}}\) and \(x_{i_{\text{max}}}\) are the maximum and minimum values of \(x_i\) in the database, respectively (Bashmeer & Hajmeer, 2000). We normalized the input and output data between 0.1 and 0.9.

In the second step, we have to say that the development of an ANN requires partitioning of the parent database into three sub-sets: training, test, and validation. The training subset should include all the data belonging to the problem domain and is used in the training phase to update the weights of the network. The validation subset is used during the learning process to check the network response for untrained data. The data used in the validation subset should be distinct from those used in the training; however they should lie within the training data boundaries. Based on the performance of the ANN on the validation subset, the architecture may be changed and/or more training cycles applied. The third portion of the data is the test subset which should include examples different from those in the other two subsets. This subset is used after selecting the best network to further examine the network or confirm its accuracy before being implemented in the neural system and/or delivered to the end user (Bashmeer & Hajmeer, 2000). We used 60% of all data for training, 10% for validation and remaining data for testing the network.

### 6.3. Network architecture and training

In most function approximation problems, one hidden layer is sufficient to approximate continuous functions (Bashmeer & Hajmeer, 2000). Generally, two hidden layers may be necessary for learning functions with discontinuities (Masters, 1994). Therefore, we might use one or two hidden layer in the network and then trained it with various neurons in each layer in MATLAB software environment. The sigmoid function was used as activation function in hidden layer but for the output neurons, first the linear activation function and then sigmoid activation function were used. Training algorithm was Levenberg-Marquart (Hagan & Menhaj, 1994).

Training process is done to learn from training data by adjusting the weights of the network. Two different criteria used to stop training; one is training error (mean square error or MSE) which has been adjusted to \(10^{-4}\) and the other is maximum epoch which has been adjusted to 1000 epochs. But one problem that would occur in training process is overtraining or overfitting. In this case, the network would memorize the training data and its performance in these data would be superior but when the new data were presented to the network its performance would be low. This

### Table 3

<table>
<thead>
<tr>
<th>No.</th>
<th>First hidden layer neurons</th>
<th>Second hidden layer neurons</th>
<th>Activation function in output neurons</th>
<th>Number of training epochs</th>
<th>Training error</th>
<th>Testing error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>–</td>
<td>Linear</td>
<td>384</td>
<td>0.0058</td>
<td>7.13</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>–</td>
<td>Linear</td>
<td>144</td>
<td>0.0007</td>
<td>1.147</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>–</td>
<td>Linear</td>
<td>411</td>
<td>8.43E-05</td>
<td>0.127</td>
</tr>
<tr>
<td>4</td>
<td>25</td>
<td>–</td>
<td>Linear</td>
<td>285</td>
<td>9.84E-06</td>
<td>0.0193</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>7</td>
<td>Linear</td>
<td>90</td>
<td>0.003</td>
<td>2.65</td>
</tr>
<tr>
<td>6</td>
<td>15</td>
<td>10</td>
<td>Linear</td>
<td>231</td>
<td>9.97E-06</td>
<td>0.0158</td>
</tr>
<tr>
<td>7</td>
<td>15</td>
<td>15</td>
<td>Linear</td>
<td>166</td>
<td>9.90E-06</td>
<td>0.0213</td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>10</td>
<td>Linear</td>
<td>176</td>
<td>1.19E-05</td>
<td>0.0239</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>–</td>
<td>Sigmoid</td>
<td>94</td>
<td>2.70E-03</td>
<td>1.76</td>
</tr>
<tr>
<td>10</td>
<td>15</td>
<td>–</td>
<td>Sigmoid</td>
<td>92</td>
<td>1.35E+00</td>
<td>1.61</td>
</tr>
<tr>
<td>11</td>
<td>20</td>
<td>–</td>
<td>Sigmoid</td>
<td>844</td>
<td>2.60E-05</td>
<td>0.04</td>
</tr>
<tr>
<td>12</td>
<td>25</td>
<td>–</td>
<td>Sigmoid</td>
<td>450</td>
<td>2.48E-05</td>
<td>0.0431</td>
</tr>
<tr>
<td>13</td>
<td>10</td>
<td>7</td>
<td>Sigmoid</td>
<td>396</td>
<td>2.90E-03</td>
<td>3.63</td>
</tr>
<tr>
<td>14</td>
<td>15</td>
<td>10</td>
<td>Sigmoid</td>
<td>152</td>
<td>1.55E-05</td>
<td>0.0265</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>15</td>
<td>Sigmoid</td>
<td>501</td>
<td>3.24E-05</td>
<td>0.0708</td>
</tr>
<tr>
<td>16</td>
<td>20</td>
<td>10</td>
<td>Sigmoid</td>
<td>69</td>
<td>1.51E-04</td>
<td>0.452</td>
</tr>
</tbody>
</table>

![Fig. 8. Performances of 16 networks of different sizes in Table 3.](image-url)
7. System evaluation

The final model was tested with the new test set data. For this purpose 6 different responses were collected for each test and were presented to the network. Then the recommended LOs from the network compared with recommended LOs from a human instructor. For 25 of 30 tests (83.3%), the network’s actual output was exactly the same as the target output, i.e., the network suggested the same LOs as the human instructor does (Tables 4 and 5).

8. Conclusion

This paper proposed a personalized multi-agent e-learning system which can estimate learner’s ability using item response theory and then it can present personalized and adaptive post-test based on that ability. Also, one of the agents of the system can diagnose learner’s learning problems like a human instructor and then it can recommend appropriate learning materials to the learner. This agent was designed using artificial neural network. So the learner would receive adaptive tests and personalized recommendations. Experimental results showed that the proposed system can provide personalized and appropriate course material recommendations with the precision of 83.3%, adaptive tests based on learner’s ability, and therefore, can accelerate learning efficiency and effectiveness. Also this research reported the capability of the neural network approach to learning material recommendation.

Acknowledgments

This paper is provided as a part of research which is financially supported by Iran Telecommunication Research Center (ITRC) under Contract No. T-500-10117. So, the authors would like to thank this institute for its significant help.

References


