Research Journal of Applied Sciences, Engineering and Technology 7(16): 3298-3303, 2014
DOI:10.19026/rjaset.7.674
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: September 14, 2013 Accepted: October 01, 2013 Published: April 25, 2014

Research Article

Intelligent Adaptive E-learning Model for Learning Management System

S. Bhaskaran and P. Swaminathan
School of Computing, SASTRA University, India

Abstract: In this study we have proposed an Intelligent Adaptive e-Learning Model that incorporates the ability to intelligently classify learners. There is a need for learning to continue, whether learners are on- or off-line. This study emphasize on developing an agent-based personalized adaptive learning model. This model is deployed as a service using agent technology and not just as an application as is the case with all other available LMS. We tested Intelligent Adaptive e-Learning Model prototype that implements an adaptive presentation of course content under conditions of intermittent Internet connections on postgraduate students studying a networking course. The study found out that it is possible for learners to study under both off-line and on-line modes through adaptive learning and the Intelligent Adaptive E-Learning system successfully classified learners and the accuracy was 85%.

Keywords: Adaptive learning system, intelligent training systems, learning management system, software agent technology

INTRODUCTION

Due to the stupendous raise in Internet users and network technology e-learning has become a simple and popular choice among the Internet users. Watering and Rijt (2006) Learning does not require any specific time limits and is also not bounded to any geographical place. In the e-learning environment, it is necessary to personalize the design of learning materials that matches with the interest of individual learners. In traditional learning environment there is a fixed content and learning sequence commonly available for all the learners. This type of arrangement is not flexible and hence a customized learning content is needed (Tay and Lim, 2011). E-Learning systems are developed to work under constant internet connections (web applications that must be accessed online in order to function). In these instances, the learners can only learn when online. These concepts are found in nearly all e-Learning systems which has been deployed by many learning institutions and even in organizations that train their staff and clients using Learning Management Systems (LMS). For instance, nearly all university e-Learning portals are only functional online, meaning that learners must be connected to the internet in order to be able to carry out learning activities. The same case applies to most online training systems for organizations. A framework is designed in the form of an Application Programming Interface (API) which can be integrated into any LMS and can be used to classify learners dynamically into various categories as defined by information in the learner model. The intention of this research is to develop an intelligent learning system that can support personalized learning. To classify learners K-nearest-neighbor algorithm is used in this study.

The Key Objective of this research is intended to meet the following objectives:

- To develop an adaptive learning model to support learning under conditions of intermittent internet connections
- To classify learners correctly using the KNN classification algorithm which considers learners' features as values

LITERATURE REVIEW

E-learning allows learners to study without the limitations of time and space which is beneficial to some extent. The ideal system should classify learners and should also provide necessary number of learning materials that is tailored for the individual user's requirement. The 'one size fits all' (Bai and Chen, 2008a) philosophy results in too much information for users and lacks personalization. However, there can be some improvements in LMS. We have identified that Learning Management Systems in this category do not satisfy the constraints to develop and manage contents to meet the demand of learning institutions. A survey conducted by Verdú *et al*. (2008) investigated the critical factors that can possibly affect learner's satisfaction in e-learning. In this study we have considered to focus on course flexibility and e-learning course quality which are primary to develop the Intelligent Adaptive E-learning Model (IAEM). Adaptive learning is an important research topic in

Corresponding Author: S. Bhaskaran, School of Computing, SASTRA University, India

learning systems. Adaptive systems are also called as Intelligent Educational Systems. Adaptive educational systems are the only alternative for traditional educational system. The traditional systems are teacher centric whereas adaptive systems are student centric. Adaptive learning offers various advantages to students. They help in personalized learning. The learning process can be effective and student satisfaction can also be measured. Adaptive systems apply Artificial Intelligence techniques to improve e-learning systems. Saleh *et al*. (2010) proposed an adaptive e-learning system. This system was a user-centric approach that helped to improve its usability and acceptance by users. E-learner requirements, including user skills, learning styles, learning strategy and other user profile information, were introduced into the system. In this system, the user learning activities are identified and are used to update the user profile. The e-learning system was developed to adjust according to a dynamic learner profile.

In developing an Intelligent Learning Management Systems, Component technologies and Artificial Intelligence are used to guide and provide e-learning. An agent recommends activities to a learner based on his access history. The recommendation should be an on-line activity including doing an exercise, providing messages on conferencing systems, running an on-line simulation, or web resources. This agent is claimed to improve course material navigation and assist the on-line learning process (Bai and Chen, 2008b). By observing user typing events, behaviors on studying lessons on web browser, tasks and examples, errors made by users and debugging events on the editor, the agent learns to understand user behavior (Li and Chen, 2009).

Adaptive learning: The software agent must be autonomous, intelligent, should adapt to the environment and should also learn. These type of special properties make them adaptable and helps to distinguish it from other programs. Autonomy is an important feature of software agent. It indicates that the software agent has the capability to perform its tasks without any direct control of user or at least with minimum intervention of user. Software agents are classified according to a common set of characteristics. There may be some cases were different classes of software agents often overlap, implying that a software agent might belong to more than one class at a time. Software Agents may, or may not, have any combination of the following characteristics: a user interface, intelligence, adaptation flexibility and collaborative properties (Saleh *et al*., 2010). User interfaces are very important and should be designed carefully so that it is easily used by the users. Not all agents interact with users. There are many collaborative agents that interact with many agents that reside in both local and remote hosts. So User Interface is not a mandatory characteristics. Wooldridge (2009) terms intelligence should include three important properties, namely:

- Reactive
- Pro-activeness
- Collaboration

Reactivity is the property to assess and study the environment inputs and should respond to stimulus. Pro-activeness is the property that initiates necessary actions that is required in its environment in order to achieve its design goals; and Collaboration property is the social ability that interacts with the other agents. These three properties are very important for Intelligent Agents. There are numerous ways available to define the intelligent nature of software agents. For the purposes of this study we use (Walker *et al*., 2009) definition in which the collaborative nature of a software agent refers to the agent's ability to share information or barter for specialized services so as to cause a deliberate synergism amongst agents. It is expected that most agents should have a strong collaborative nature without necessarily implying other intelligence properties. Adaptability is also an important characteristic that can also be regarded as an important property, although it is not counted as being a prerequisite for identifying an agent as intelligent. Adaptability refers to an agent's ability to customize itself on the basis of previous experiences. An agent is said to be adaptive if it can assess the environment and can classify them successfully. An agent is considered to be flexible when it can choose dynamically which actions to invoke, in what sequence, in response to the state of its external environment. A stationary agent can be seen as a piece of autonomous (or semi-autonomous) software that resides permanently on a particular host. An example of such an agent is one that performs tasks on its host machine such as accepting mobile agents, allocating resources, performing specific computing tasks, enforcing security policies and so forth. A mobile agent is a software agent that has the ability to transport itself from one host to another in a network. The ability to travel allows a mobile agent to move to a host that contains an object with which the agent wants to interact and then to take advantage of the computing resources of the object's host in order to interact with that object. An example of a mobile agent is provided by a flight-booking system where a logged request is transferred to a mobile agent that then traverses the web seeking suitable flight-information quotations as well as itineraries. We considered only stationary agents in this research. Agents reside on host devices and only interact with others through the implemented functions.

Analysis: The following is a detailed description of the learning process as designed in this research:

- **Registration:** In this details of a new learner are gathered and Login Credentials were created for subsequent logins and use of the system. An existing learner could also login and continue with the learning process.

- **Initial question classification:** In this stage, the courses were designed with immense care so that it covers all portions of the course, beginning with the basic level through to the expert level. At the Basic level the introductory concepts of the course was set. For the expert level the most advanced concepts of the course was set. Questions was also carefully designed in such a way that those presented at the beginning level got tested with the basics of the course whilst the questions presented at the expert level tested the complex concepts of the course. Each question was given a weight. The weights also reflected the level of the course being tested by the question, hence weights increased from first question to the last question. If a learner failed in the beginner questions and subsequent ones he would be classified as a basic learner. Depending on how the learner performed in each section, together with other learning attributes, the learner was classified into an appropriate class level.
- **Determine new class level:** Course level classification is done to determine new class level for the learner and the relevant information
- Repeat Step 2 and Step 3 until expert level is achieved.
- **Course evaluation:** Upon fulfilling all the requirements for the expert level, both soft and hard copies of the evaluation questionnaires were provided and the learners' assessment of the system was captured.

METHODOLOGY

System architecture: In this research, agent technology was used to build Intelligent Adaptive E-learning Model (IAEM). We considered only stationary agents in this research. Agents reside on host devices and only interact with others through the implemented functions.

The System architecture consists of three types of Agents:

1. Learners Agent
2. Classifier Agent
3. Collaborative Agent

1. **Learners agent:** The main function of the learner agent was to facilitate the learning process both on-line and off-line. This agent provided interaction between the learner and the system and also provided the following facilities like learner register or login, access his or her profile details, get learning materials, read the notes, answer section quizzes and view all changes up to date. This module was connected to the data store and displayed information from the data store to both the learner and instructor. There were two versions of the learner model namely:
   - The client model (off-line model)
   - The server model (client-server or server-centric model)

   The client model was installed on the client machine (local machine) which was used whilst the learning was off-line. The server model was installed on the server machine (remote) and was used for online learning. It also facilitated profile updates by updating the off-line model whenever the internet connection was re-established.
2. **Classifier agent:** In this module four input attributes are only considered namely:

a. Score
b. Quiz time
c. Reading time
d. Weight of questions

The details of these four attributes are as follows:

a. Score: All learning institutions use the scores as a major factor in grading their students. The higher the score a student attains, the higher the grade the student is awarded. We have classified our own scores as given below.

b. Quiz time: This refers to the time the learner uses in taking a test. Normal practice is that exams have a specific time allocated for their completion (Fig. 1). When the allocated time is over, the learner sitting for the examination stops or, in the case of an online examination, is timed out. In this research, the system did not time out a learner if the set time was exceeded. Instead, the more time spent after the set time, the less the performance for the learner and the lower the class assigned. On the other hand, if the learner took less than the set time to sit for the examination, a higher the level of classification was assigned. A combination of both performance and time or other resources spent in achieving the learning is called learning efficiency and is a measure of a learner's expertise.

c. Reading time: This is the time taken by the learner to read the learning material for the level or section. Normally, there is no limit in time for reading notes in preparation for an examination. A learner can take as long as possible to read the notes. In this research, a reasonable time threshold was set for reading the notes to enable both slow readers and fast readers to complete a topic. It was adjusted during testing of the course to make it appropriate.

d. Weight of questions: The questions were weighted in an increasing manner from the first question to the last question. In addition, the questions were designed such that basic questions come first and complex ones come toward the end. Considering the design of the questions, it was prudent that basic questions were assigned less weight compared with complex questions. If a

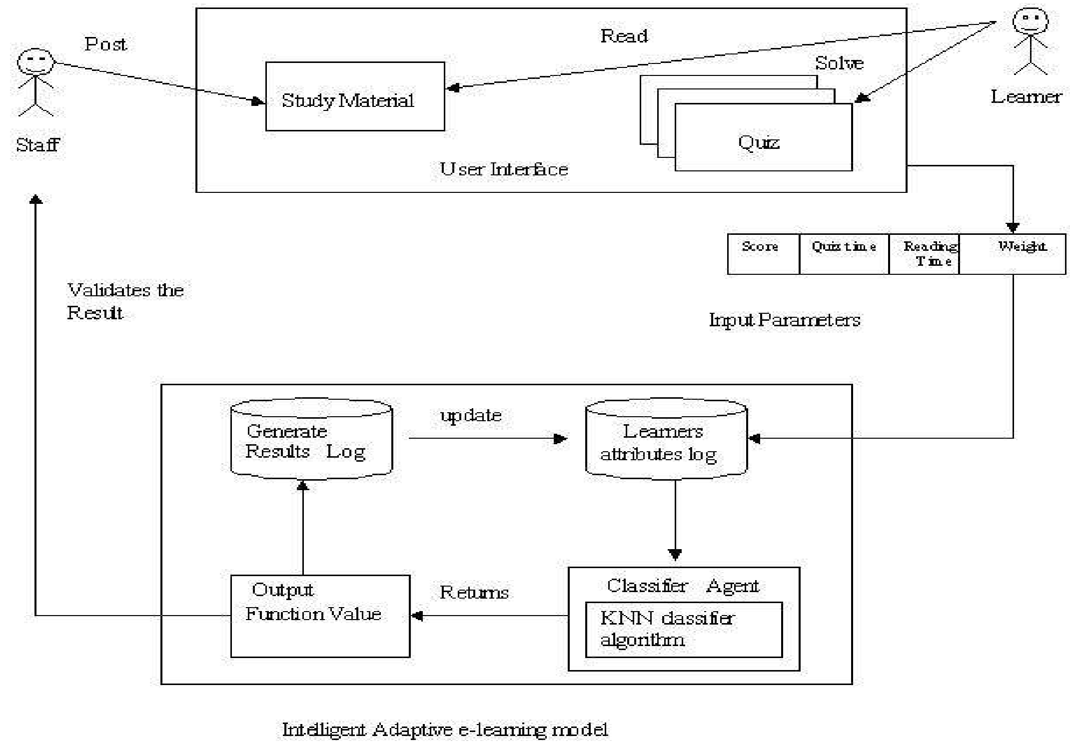

Fig. 1: Architecture diagram for intelligent adaptive e-learning model

Table 1: Score input and course level

Score	Output course level
0-40	Beginner
41-60	Intermediate
61-80	Advanced
81-100	Expert

learner failed to answer basic questions correctly, it was assumed that he was still a beginner and was assigned to the beginner level class.

Output function (course level): The course level was based on the experience of the learner. It was assumed in general that learners can be easily grouped into four levels as given in Table 1 with the novices in the course being studied being referred to as 'Beginner'. The second category, we thought, should have a bit more experience, hence the label 'Intermediate'. 'Advanced' learners were the ones who had vast knowledge of the subject matter and 'Expert' learners were those who had the ability to apply the knowledge from the study. There were no particular criteria considered in coming up with these learner classes.

The Classifier Agent uses K-Nearest-Neighbor (KNN) algorithm to classify new learners. The parameter K is an integer parameter representing the number of nearest neighbors to a new learner and whose most common class becomes the new learner's class. The default value of K can be fixed to an odd number such as 3. However, a low value of K restricts the classification of the new learner to classes of only a few neighbors. The best choice of K depends on the data and, in general, larger values of K reduce the effect of noise on classification but make boundaries between the classes less distinct. A good K is chosen using heuristic techniques such as cross-validation. The parameter K should also be an odd integer number so that the majority vote is always attained. Even numbers for K can result in a tying vote that can hamper correct classification. For this research, K was 9. This figure was arrived at after considering that learners would be increasing with time and also to avoid restricting classification to a few training examples.

Training data set: The training data had two sections: the input attributes and the output function values (i.e., the associated class) which was represented as a vector in the form <a1, a2, a3,… n> are the input attributes and <Oa> is represented as Output function. In this research, the training example input attributes are defined as follows: <score, quiztime, readtime, weight> and <course level> as output function. Given the training data, when a new learner joins with feature vector values such as <45, 43 10, 9><?>, the KNN algorithm takes the new instance and compares it with the training data. The distances between the new instance attributes and the training data attributes are calculated. It is important to note that after classification of the new instance is carried out, the instance becomes part of the training data. The classifier agent receives data from the environment and after applying the KNN algorithm, classifies the learner and updates his or her profile dynamically. This agent is autonomous as it does not require any supervision and makes decisions depending on the prevailing information. This agent trains the model so that, based on the experience the model has with existing training data, it can classify new instances correctly.

3. **The collaborative agent:** This agent collaborates with the classifier agent and learner module so that after the classifier agent had made changes with regard to the learner status, it made sure that learner's profile matched both locally and remotely. The agent tried to establish a connection to the URL of the online application by using the public Internet Protocol address. Depending on whether the application was accessible or not, internet connection establishment was confirmed or failed. If connection establishment was confirmed then the remote version was used; if not, the local version was used. The connection of the model to the local and remote databases was checked. After establishing the connection status, status of the contents was compared. The status was determined by examining which database had more records and/or latest records. If the local copy was the latest, then the remote copy was updated and vice versa. The module also displayed a message to a learner when there was no connection to the remote server and also allowed learner to continue learning with a local copy. The local copy was later synchronized with the remote copy when the connection was reestablished. For synchronization of both databases to take place, the synchronizer agent in the client machine located the domain address for the remote server and then connected to the database in the remote server. All records were compared. The records of the side with more or the latest records were copied to the side with the missing data.

EXPERIMENTAL RESULTS

In Computer Lab 20 students used 6 h for learning a day for 5 days. Two sets of learners were used. One group connected to both the intranet and internet, downloaded the information to the database in the local module and used it off-line to learn. The other group did the learning online only. The two groups were swapped around half-way through the course and the process was repeated (Table 2).

The experiment was carried out with 20 students. Zero was used to indicate a query data that was classified incorrectly whilst 1 was used to show query data that was classified correctly. Of the 20 learners studied, only 3 learners were classified incorrectly. The percentage accuracy was: $17/20 \times 100 = 85\%$

Table 2: Learner prediction input-output values

Score	Read time	Quiz time	Weight	Course level
90	40	15	8	Excellent
75	47	10	9	Advanced
30	65	35	4	Beginner
50	50	20	7	Intermediate

After the experiment was carried out, the learners were given hard-copy questionnaires. The survey collected information related to the first and second objectives to double-check results from the logging of the learner activities. The results show that an overwhelming majority of the students indicated that they were able to learn both on and off-line. A great majority of the students gave a positive feedback and also indicated that it was easy to learn with the system, that they were able to get appropriate notes and that they were classified fairly. The questions were also well designed. This application has been developed using Java language.

CONCLUSION

A number of conclusions were made. The first conclusion is to classify learners appropriately. From the percentages of the learners that were classified correctly, it could be concluded that the model was accurate in classifying learners, with an accuracy of 85%. The second conclusion is from the survey results, 18 out of 20 (90%) learners said that they were classified as per their expectations. The third conclusion is the learning process took place both on-line and off-line.

LIMITATIONS

- The study was conducted over 6 hours a day for 5 days.
- Only the KNN algorithm was used for the experiment. It would have been better to have been able to compare its performance against that of other algorithms.
- During the testing stage of the model, not all stakeholders were involved due to time constraints. It would possibly have been a good idea to hold off and let them participate later so that the model could gain a wider audience acceptance.

RECOMMENDATIONS

The available duration of time for the research was very less. The amount of data available for use during classification was also limited. It would have been ideal to observe changes in learner knowledge levels over an entire semester instead of over just 5 days of intensive work. KNN algorithm usually classifies data using up to 20 attributes. Only three attributes were used in this research. In future we will try to apply many parameters to the model.

REFERENCES

Bai, S.M. and S.M. Chen, 2008a. Evaluating students learning achievement using fuzzy membership functions and fuzzy rules. Expert Syst. Appl., 34(1): 399-410.

Bai, S.M. and S.M. Chen, 2008b. Automatically constructing grade membership functions of fuzzy rules for students' evaluation. Expert Syst. Appl., 35(3): 1408-1414.

Li, T.K. and C.M. Chen, 2009. A new method for students' learning achievement evaluation by automatically generating the weights of attributes with fuzzy reasoning capability. Proceeding of International Conference on Machine Learning Cybernetics, pp: 2834-2839.

Saleh, A.A., H.M. El-Bakry and T.T. Asfour, 2010. Adaptve e-learning tools for numbering systems. Proceedings of the 9th WSEAS International Conference on Telecommunications and Informatcs, pp: 293-298.

Tay, K.M. and C.P. Lim, 2011. A fuzzy inference system-based criterion-referenced assessment model. Expert Syst. Appl., 38(9): 11129-11136.

Verdú, E., L.M. Regueras, M.J. Verdú, J.P. de Castro and M.A. Pérez, 2008. An analysis of the research on adaptive learning: The next generation of e-learning. WSEAS T. Inform. Sci. Appl., 5(6): 859-868.

Walker, E., N. Rummel and K.R. Koedinger, 2009. CTRL: A research framework for providing adaptive collaborative learning support. User Model. User-Adap., 19(5): 387-431.

Watering, G.V.D. and J.V.D. Rijt, 2006. Teachers' and students' perceptions of assessments: A review and a study into the ability and accuracy of estimating the difficulty levels of assessment items. Educ. Res. Rev., 1: 133-147.

Wooldridge, M., 2009. An Introduction to Multi-agent Systems. John Wiley and Sons, Cambridge.