Reinforcement Learning Approach for Adaptive E-learning 
Systems using Learning Styles

Balasubramanian Velusamy, S. Margret Anouneia and George Abraham 
School of Computer Science and Engineering, VIT University, Vellore-632014, India

Abstract: This study aims to provide a mathematical modelling for an adaptive e-Learning environment using SARSA algorithm, by relating it to the concept of Reinforcement Learning. An adaptive e-learning system based on learning styles is very much alike an intelligent agent. The system needs to assess the various interactions from the user and provide them with the best possible content, so as to enhance the learning experience of the user. Successive interactions with the system by the same user should result in he/she being provided with the best content in the same manner as the previous time, if not for a better option. This is possible as the system will have learned from its previous interaction.

Key words: Semantic, e-learning, SARSA, ontology, learning styles

INTRODUCTION

The boom in the Internet sector and its percolation into all domains of human life as changed the dimensions of the education system with the introduction of virtual learning or e-learning. There has and will always be a steady rise in the demand of e-learning systems catering to different needs in the various fields and levels of education (Brusilovsky and Peylo, 2003; Nath et al., 2012). An e-learning system with the facility of personalization or adapting to the users’ needs is the ideal system that is required, as people with different skill sets use the system. Some people may be fast learners while some may be slow, some may need to practice more problems while others may need just example. These preferences are in general called the learning styles of an individual. Creation of a learner model enables in the capturing of all the preferences and needs of an individual. This learner model can be extracted from the personality factors of an individual like their learning styles, their behavioural factors like their browsing history or general patterns in browsing and knowledge factors like the user’s prior knowledge (Ozpolat and Akbar, 2009). Out of all the three factors mentioned above, creation of a learner model from the learning styles of an individual yields the best results if proper personalization can be provided based on it (Dag and Gecer, 2009).

The main challenge is the detection of the learning styles. Researchers have described various learning styles models like Myers and Myers (1980), Kolb (1984), Honey and Mumford (1992), Dunn and Dunn (1978) and Felder and Silverman (1988). Research has proved that the Felder-Silverman Learning Style Model (FSLSM) is the most suited for the engineering students’ environment as it also considers the psychological aspects of a person (Felder and Spurlin, 2005). The Index of Learning Styles (Felder and Soloman, 2012) is a questionnaire-based approach for detection of learning styles based on the FSLSM. The problem with questionnaire-based approach is that it suffers from the “inaccurate self-conceptions of students” (Dung and Florea, 2012; Graf et al., 2008) at a specific time. Moreover these questionnaires are incapable of tracking the changes in a learner’s learning style, i.e., the dynamicity of the learning style.

As a result of these problems, various researches have been conducted to come out with alternate automated solutions for learning style detection. These works can be broadly classified into two groups: data-driven approach and literature-based approach. Some of the noticeable works in the data-driven approach are by using Bayesian Networks (Schiaffino et al., 2008; Garcia et al., 2007), NBTree classifiers (Ozpolat and Akbar, 2009) and Genetic Algorithms (Chang et al., 2009). Literature-based approach is a relatively new method with some of the noticeable works being done by Graf et al. (2008), Dung and Florea (2012) and Simsek et al. (2010).

Personalization of the e-Learning system means to provide a system that adapts according to the learners’ learning process.

Corresponding Authors: Balasubramanian Velusamy, Parallel and Distributed Computing Division, School of Computing Science and Engineering, VIT University, Vellore-632014, India Tel: 0416-2202811

2306
ADAPTIVE WEB-BASED EDUCATION

The concept of an adaptive system was initially stressed by Brusilovsky and Peylo (2003). They talk about improving the system of web-based education by providing an Adaptive and Intelligent Web-Based Educational System (AIWBES) as an alternative to the traditional systems. AIWBES adapts to the learners’ needs, knowledge and behaviour like a human teacher would do. An adaptive system modifies its solutions to a problem based on various factors, for instance the learners’ previous experience with the system whereas an intelligent system provides the same solution irrespective of the different needs of the learners. AIWBES is a mixture of adaptive hypermedia technologies and intelligent tutoring technologies. It also contains adaptive information filtering, intelligent monitoring and intelligent collaborative learning. Adaptive hypermedia mainly consists of adaptive presentation and adaptive navigation support while intelligent tutoring mainly consists of curriculum sequencing, problem solving support and intelligent solution analysis.

AUTOMATIC LEARNING STYLE RECOGNITION

Due to the various disadvantages of questionnaire-based learning style detection, the process has to be automated so that it can incorporate various aspects of the learner while modelling the learner. The process of automatic detection of learning styles consists of two phases: Identifying the relevant behaviour for each learning style and Inferring the learning style from the behaviour (Graf, 2007), as shown in Fig. 1.

The first step of identifying the relevant behaviour for each learning style consists of the following phases:

1. Selecting the relevant features and patterns of behaviour, classifying the occurrence of the behaviour and defining the patterns for each dimension of the learning style (Graf, 2007), as shown in Fig. 2.

All this is performed by studying the various literatures of the respective learning model and other supportive research works that have already been done. The second step of inferring the learning style from the respective behaviour is where the approaches differ. But the initial step is of preparing the input data which is common. This input data is prepared from the extracted information and is formulated in the form of matrices that corresponds to each learning style. Then the calculation methodology can be data-driven or literature-based approach (Graf, 2007), as shown in Fig. 3.

Fig. 1: Idea of automatic detection of learning style preference

Fig. 2: Identifying the relevant behaviour for each learning style

2307
In computer science, reinforcement learning (Sutton and Barto, 1998) is an area of machine learning concerned with what actions an agent, i.e. an intelligent program, should take in an environment so as to maximize the cumulative reward. The agent learns by sensing various parameters from the environment, exploring various possibilities for a better reward and when performing the same task again, exploiting the already learned best path.

The e-learning system that is the topic of concern in this study is similar to an intelligent agent. It senses the various user interactions and has to decide on the best possible responses to the user so as to enhance the learning experience of the user. If the same user uses the system again, then the previously learned options that were best suited to the person should be given again, if not for a better option.

Reinforcement learning scenarios are described by states, actions and rewards. There exist two main reinforcement learning algorithms - Q-Learning Algorithm (Watkins, 1989) (Watkins and Dayan, 1992) and SARSA Algorithm (Rummery and Niranjan, 1994).

**Q-learning algorithm**: Q-Learning (Watkins, 1989) is a form of model-free reinforcement learning (Watkins and Dayan, 1992). The problem domain consists of an agent its various states S and a set of actions per state A. The agent can move from one state to another by performing some action a \in A. The transition, i.e. the next state gives a reward to the agent. The goal of the agent is to maximize the total reward. This is achieved by optimizing the actions for each state. Hence, there exists a function Q that calculates the quality of each state-action combination. Initially Q returns a fixed value that the designer has set. Then during each step when the agent is rewarded, new values are calculated and updated (Eden et al., 2013).
Table 1: Summary of the literature survey

<table>
<thead>
<tr>
<th>Study</th>
<th>Approach</th>
<th>Technology</th>
<th>Key points</th>
<th>Assessment methods</th>
<th>Precision/accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bergasa-Suso et al. (2005)</td>
<td>Data-driven</td>
<td>Browser-based system with rules</td>
<td>Processing dimension</td>
<td>67 students-ILS (Training) 7 students-ILS (Lessons)</td>
<td>71% - Processing</td>
</tr>
<tr>
<td>Garcia et al. (2007)</td>
<td>Data-driven</td>
<td>Bayesian networks</td>
<td>Detection only</td>
<td>27 systems engineering students-AL-SAVER</td>
<td>58% - Processing 7% - Perception 63% - Understanding</td>
</tr>
<tr>
<td>Schiaffino et al. (2008)</td>
<td>Data-driven</td>
<td>Bayesian networks</td>
<td>Detection-suggestions</td>
<td>42 Systems engineering students-AL-SAVER with each teacher</td>
<td>83% feedback received was positive 77.33% - Input 79.23% - Processing 76.67% - Perception 73.33% - Understanding</td>
</tr>
<tr>
<td>Graf et al. (2008) Based on Graf (2007)</td>
<td>Literature-based Simple rules on matching hints</td>
<td>LMS independent Better results that data-driven approach</td>
<td>127 students-Info Sys. Comp. Sci. Austria Univ.-Object oriented modeling-moodle LMS</td>
<td>10 graduate student (Training) 30 graduate student (Testing)-PosTech</td>
<td>53% - Input 70% - Processing 73% - Perception and understanding</td>
</tr>
<tr>
<td>Ozpolat and Akbar (2009)</td>
<td>Data-driven</td>
<td>NBTREE</td>
<td>Detection-suggestion</td>
<td>Uses only data objects selected by the user LMS independent</td>
<td>IRIS dataset by UCI 117 students-SCORM-compatible Java-based</td>
</tr>
<tr>
<td>Chang et al. (2009)</td>
<td>Data-driven Enhanced k-NN clustering with GA</td>
<td>k-NN pre-contrast and post-comparison reduced No of behavioural features</td>
<td>LMS-windows XP</td>
<td>67 students-ILS (Training) 7 students same research task-lessons</td>
<td>82% - Input 81% - Processing 69% - Understanding</td>
</tr>
<tr>
<td>Sanders and Bergasa-Suso (2010)</td>
<td>Data-driven Browser-based system with rules for reasoning Simple rules on matching hints</td>
<td>More dimensions Improved rules Unknown category Processing dimension 6 features considered</td>
<td>27 students-Comp. Educ.- Derivatives-Moodle LMS</td>
<td>Study conducted on 25 and 15 learners participating in a bookmarking site such as <a href="http://www.tagme.com">www.tagme.com</a> By varying the Learning rate No. of hidden neurons Values of epochs</td>
<td>For No. of learners = 25, Recognition = 72% For No. of learners = 15, Recognition = 86.66% For learning rate = 0.01, No. of hidden neurons = 40 and Epochs = 150, Recognition rate = 93.33% for 15 learners</td>
</tr>
<tr>
<td>Darwesh et al. (2011)</td>
<td>Data-driven Analyzing web page content interactions using social bookmarking</td>
<td>No special effort from student side for collecting data LMS independent</td>
<td>98 undergraduate students fundamentals of computer networks course</td>
<td>EFC has more accuracy than grouping based on behaviour in log files of LMS</td>
<td></td>
</tr>
<tr>
<td>Ghorbani and Montazer (2011)</td>
<td>Data-driven Evolutionary Fuzzy Clustering (EFC) method using genetic algorithm</td>
<td>People with similar learning styles in a cluster High computational and memory usage costs, so use particle swarm optimization technique Better classification for “Unknown”</td>
<td>Comp. Sci. Eng.-Anna Univ.-C-language</td>
<td>-NA-</td>
<td></td>
</tr>
<tr>
<td>Deborah et al. (2012)</td>
<td>Data-driven Fuzzy logic</td>
<td>Bell-shaped membership function</td>
<td>44 UG students-Comp. Sci.-Politechnics Univ.-Bucharest-AI course-Web-based LMS POLCA</td>
<td>70.15% - Input 72.73% - Processing 70.15% - Perception 65.91% - Understanding</td>
<td></td>
</tr>
<tr>
<td>Dung and Florea (2012)</td>
<td>Literature-based Simple rules on matching hints</td>
<td>LMS independent Parameters No. of visits and time spent</td>
<td>40 M.Sc. students on 4 different courses, done in three phases 105 students-C-DAC Hyderabad and D Labs-R and D+courses like Embedded systems, system S/W and Adv. business computing</td>
<td>Good accuracy for student user profile data sets less than 150</td>
<td></td>
</tr>
<tr>
<td>Saberi and Montazer (2012)</td>
<td>Data-driven ILS + LMS logs + Bayesian networks</td>
<td>Improved accuracy Decreased uncertainty</td>
<td>40 M.Sc. students on 4 different courses, done in three phases 105 students-C-DAC Hyderabad and D Labs-R and D+courses like Embedded systems, system S/W and Adv. business computing</td>
<td>-NA-</td>
<td></td>
</tr>
<tr>
<td>Jyothi et al. (2012)</td>
<td>Data-driven Recommender system based on ILS and clustering</td>
<td>Best for data sets less than 150 users Prior knowledge of learner needed LS captured by ILS</td>
<td>105 students-C-DAC Hyderabad and D Labs-R and D+courses like Embedded systems, system S/W and Adv. business computing</td>
<td>Good accuracy for student user profile data sets less than 150</td>
<td></td>
</tr>
</tbody>
</table>

The following represents the Q-learning algorithm (Watkins, 1989):

\[
Q(S_t, a_t) = Q(S_t, a_t) + \alpha [R(S_t, a_t) + \gamma \max_Q(Q(S_{t+1}, a_{t+1}) - Q(S_t, a_t))]
\]

\[t = \text{Current interaction} \]
\[t+1 = \text{Next interaction} \]
\[Q(S_t, a_t) = \text{The Q-values of the current interaction} \]
\[R(S_t, a_t) = \text{Reward obtained for performing action } a_t \text{ in } S_t \]
\[\alpha = \text{Learning rate (0} \leq \alpha \leq 1\text{)} \]
\[\gamma = \text{Discount factor that decides the importance of future rewards (0} \leq \gamma < 1\text{)} \]

Where:
- \(Q(S_t, a_t)\) = Updating the old value
Q-Learning algorithm is also called as an “Off-Policy” or “Policy Independent” algorithm as it does not depend on any policy. Policy is the decision process used to select an action given a certain state. Q-Learning works on the greedy mechanism by selecting the maximum of the Q-values of all the state-action pairs that are possible from the current state. The agent learns through experience (unsupervised) by exploration. Each exploration is called an episode (Sutton and Barto, 1998). In an episode, an agent moves from initial to goal state after which the next episode starts.

In the Q-learning algorithm, if the discount factor (γ-value) is set to 0, i.e. γ = 0 then:

\[ Q(S_n, a) = Q(S_n, a) + \alpha [R(S_n, a) + Q(S_{n+1}, a)] \]  

(2)

This means that the updating can happen then and there itself without the consideration of a new state.

Now, if the learning rate (α value) is set to 0, i.e. α = 0 then:

\[ Q(S_n, a) = Q(S_n, a) \]  

(3)

This means that no learning takes place and the value remains as it is.

Else, if the learning rate (α value) is set to 1, i.e. α = 1 then:

\[ Q(S_n, a) = Q(S_n, a) + R(S_n, a) - Q(S_n, a) - R(S_n, a) \]  

(4)

This means that the agent will consider only the most recent information, i.e., the reward only. This is too simplistic.

If α = 0.5 (for example), the old and the new Q-values meet halfway, given the reward.

Hence, when the discount factor γ is set to 0, the agent will consider only the current reward. It has only short term greedy goals.

In the Q-learning algorithm, if the discount factor (γ value) is set to 1, i.e. γ = 1 and also α = 1 then:

\[ Q(S_n, a) = Q(S_n, a) + R(S_n, a) + \max [Q(S_{n+1}, a_{n+1}) - Q(S_n, a)] \]

(5)

\[ \max [Q(S_{n+1}, a_{n+1}) - Q(S_n, a)] \]

This means that the updated Q-value for a state is equal to the reward plus the maximum of the possible Q-values from the state. As α tends to 1, or more, the more it optimizes for long-term goals. It is ideally set in between 0 and 1 (Eden et al., 2013).

Also to be noted is that for values of α that are close to 1, the same Eq. 5 can be used.

**Sarsa algorithm**: SARSA (Rummery and Niranjan, 1994) algorithm is an improvement on the Q-learning algorithm where the name comes from the fact that the rule that updates the Q-value depends on the current state aₙ₊₁ the action the agent chooses aₙ the reward rₙ the next state that the agent will be in after taking the action aₙ₊₁ and the action that the agent will take in the new state aₙ₊₁. The name SARSA stands for the same-state, action, reward, state (next state), action (next action).

The following represents the SARSA algorithm (Rummery and Niranjan, 1994; Eden et al., 2013; Van Hasselt, 2013):

\[ Q(S_n, a) = Q(S_n, a) + \alpha [R(S_n, a) + \gamma Q(S_{n+1}, a_{n+1}) - Q(S_n, a)] \]  

(6)

SARSA algorithm is also called as an “On-Policy” or “Policy Dependent” algorithm as it depends on the decision process used to select an action given a certain state. It does not have a greedy approach like the Q-learning algorithm.

The policy selection process is not always selecting the action that results in the maximum Q-value as this will lead to a phenomenon of “local maxima”. Instead it is determined on a factor epsilon which determines the extent to which the actions are randomized.

There are three types of action selection policies:

- **ε-greedy**: Action with the highest estimated reward is chosen independent of the Q-value estimates
- **ε-soft**: Best action is selected with a probability of 1 - ε and rest of the time the actions are selected uniformly
- **Softmax**: The problem with both the above methods is the uniform selection that may result in the worst possible action being selected as second best

A solution to this is the softmax policy where a rank or a weight is assigned to each action according to its action-value estimate and actions are selected based on these weights, as a result of which the worst actions are unlikely chosen.

Assuming that the ε-soft policy is being followed Eq. 6 can be re-written as:

\[ Q(S_n, a) = Q(S_n, a) + \alpha [R(S_n, a) + k \gamma Q(S_{n+1}, a_{n+1}) - Q(S_n, a)] \]  

(7)

where, is k-The probability of selecting the action.

**GENERIC MODELLING SOLUTION**

Consider the state-diagram, as shown in Fig. 4, with the action and their respective rewards mentioned on the
Fig. 4: Generic state-action-reward diagram

By randomly selecting an action, select a2:

\[ Q(S,a_2) = Q(S,a_2) + \alpha[R(S,a_2) + \gamma \max(Q(S_12), \text{all actions}) - Q(S,a_2) - \gamma \max(Q(S_12,a_5), Q(S_12,a_6), Q(S_12,a_7)) - Q(S,a_2)] \]

**Solving by SARSA algorithm:** Initially Q-values for all are set to 0.

Observations from Table 2 and 3 show that there are 3 possible actions that can be taken from S, i.e., a1, a2 and a3.

The first action a2 is selected randomly. On performing this action, the system will reach state S12. From here, the system can select any of the three actions-a5, a6 or a7. This selection is done with a probability 1-ε for the best action and uniform probability for the remaining actions. The best action is assumed to be the one that results in highest reward. This probability is substituted as the value of k in the equation.

Therefore:

\[ Q(S, a_2) = Q(S, a_2) + \alpha[R(S, a_2) + k \gamma[S_12, \text{action}]) - Q(S, a_2)] \]

In both the solutions, after the first episode is over, the system, i.e., the agent starts over again. In the consecutive runs, the agent is aware of the best path with the help of the Q-values. This helps the agent to work efficiently.

**MODELLING THE E-LEARNING SYSTEM**

For modelling the e-learning system, SARSA algorithm is best suited as it depends on the action that is performed rather than randomly selecting the action that gives the best result.

Assuming that the e-learning system is a Learning Management System (LMS) like Moodle...
A course, suitable for e-learning environment, is uploaded to the system. The course contents can be theory relevant to the topic, examples, programming codes, practice exercises, diagrams and tests to evaluate the student. Users can browse through the system, select the course, read its relevant theory, go through the examples, practice exercise questions, solve tests and participate in forum discussions. The forum is assumed to be one where the student posts a question to which the teacher responds to. Peer discussions are not assumed to be present at the moment. The e-learning system keeps track of the students' log files, details from which we use for later analysis.

Assuming some of the possible states the system can be in, the possible actions that cause the transformation and also some of the possible rewards that the system can offer.


**Actions:** “read”, “readMore”, “solve”, “submitting”, “askDoubt”, “discuss”, “forMoreUnderstanding”, “givingUp”, “Q and A”, “understanding”, “exit”.

**Rewards:**
- Completely understood = +100
- Well understood = +50
- Moderately understood = +30
- Did not understand = -50
- No reward = 0

The rewards have been assumed based on the following assumptions:
- The reader will benefit more if he/she reads the theory first and then solve the exercises
- He/she further benefits if he reads some extra materials
- The learner stands to earn more reward if after understanding and discussion of the result, he/she goes to solve more exercises
- If he/she exists without understanding or discussing the explanation of the results, he/she earns a negative reward
- Exiting after understanding or discussion of the results earns the maximum reward

The following diagram shows a part of the various state transitions and their respective rewards with respect to the above mentioned assumptions.

Table 4 gives a tabular representation. Fig. 5, giving the rewards received on performing a particular action at each stage.

When we solve using the SARSA algorithm, the learning rate $\alpha$ is set to 0.8 (assumption) while the discount factor $\gamma$ is assumed to be 1. The probability of selecting the best action is assumed to be 30% while the remaining actions are selected uniformly. This sets $\epsilon = 0.7$.

After the initial episode, the system knows the best action to be considered for earning the maximum reward. In all further interactions of the user with the system, the

Fig. 5: State-Action-Reward diagram of assumed scenario
Table 4: Reward table of assumed state-action combination

<table>
<thead>
<tr>
<th>Action</th>
<th>Read</th>
<th>Read more</th>
<th>Solve</th>
<th>Submitting</th>
<th>For more understanding</th>
<th>Exiting</th>
<th>Discuss</th>
<th>Understanding</th>
<th>Exit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start</td>
<td>+50</td>
<td>-</td>
<td>+50</td>
<td>-</td>
<td>-</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Read/Theory</td>
<td>-</td>
<td>+50</td>
<td>-</td>
<td>+30</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ReadingExtra</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Material</td>
<td>-</td>
<td>-</td>
<td>+30</td>
<td>-</td>
<td>-</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Solving</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Exercises</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>WaitFor</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Result</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-50</td>
<td>+30</td>
<td>+30</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Discussion</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+50</td>
<td>+100</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Understanding/explanation</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+100</td>
<td>+30</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Idle</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>End</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

CONCLUSION AND FUTURE WORK

Adaptive learning in an e-learning system can be easily compared to an agent with reinforcement learning approach. SARSA algorithm suits the situation much more than Q-learning algorithm mainly because of its selection policy.

Every user has a unique characteristic called its learning style. It is the manner in which they learn or process information. The difference in selection of an action to shift from one state to another so as to earn the maximum is reward is due to their differences in learning styles. Learning in an e-learning system becomes more efficient if the system can be adaptive and personalized based on the user’s learning styles. To provide more precision in the recognition of learning style, a new methodology that incorporates both the approaches by creating an ontological framework for modelling the learner and using fuzzy reasoning engine is proposed. Simple rule-based reasoning can be performed on the ontology to extract the required content. The recognized learning style can be stored within the system in the learner database and can be used for further interactions with the learner so as to provide the learner with his/her relevant content. This makes the learning process similar to reinforcement learning in machines.

REFERENCES


