An Extended Learner Modeling Method to Assess Students' Learning Behaviors

Yi Dong
Institute for Software Integrated Systems
Vanderbilt University
Nashville, U.S.
yi.dong@vanderbilt.edu

Gautam Biswas
Institute for Software Integrated Systems
Vanderbilt University
Nashville, U.S.
gautam.biswas@vanderbilt.edu

ABSTRACT

This paper discusses a novel approach for developing more refined and accurate learner models from student data collected from Open Ended Learning Environments (OELEs). OELEs provide students choice in how they go about constructing solutions to problems, and students exhibit a variety of learning behaviors in such environments. Building accurate models from limited amount of student data is difficult; to address this we develop a methodology that uses Monte Carlo Tree Search methods to boost the initial set of student action sequences in such a way that we can learn more accurate models of students' learning behaviors. We use a HMM representation to model students' learning behaviors and demonstrate the effectiveness of our approach by running a case study on data collected from 98 students, who worked with the Betty's Brain system for four days. The results have interesting implications for learner modeling and its applications to adaptive scaffolding of students' learning behaviors and strategies as they learn from OELEs.

1. INTRODUCTION

In recent work on computer-based STEM learning environments, there has been a focus on developing OELEs, which provide students with a learning goal, usually in the form of a complex problem or a modeling task, and a set of tools that support the problem-solving/modeling task [1]. To succeed, these students need to make choices on how to structure the solution process, explore alternative solution paths, develop awareness of their own knowledge and problem-solving skills, and develop strategies that support more effective learning and problem solving [2].

Given the complexities students face in working with OELEs, it is imperative that effective scaffolding be provided to help them progress in their learning and problem solving tasks and achieve their learning goals. However, an important component of effective scaffolding is learner modeling that can accurately capture students' cognitive and metacognitive processes. In this work, we take on the challenge of using data-driven techniques to construct accurate models of learner behaviors and performance by analyzing the learners' activity data from OELEs.

Typically, data-driven methods require large volumes of rich data to support accurate and robust learner modeling. However, collecting such data from OELEs, especially in K-12 settings can be a difficult, time consuming process. To alleviate this problem, we propose a novel set of techniques that combine the use of Hidden Markov Modeling (HMM) [7], Monte Carlo Tree Search (MCTS) [3], and a reinforcement learning methodology [4] to generate artificial student activity data that simulates students behavior corresponding to learning activities captured in the log data. The original student data combined with the artificially generated data is then used to derive more accurate and complete models of students' behaviors and strategies used for learning.

In section 2, we briefly review the Betty's Brain OELE that we use for this work, and describe the overall learner modeling approach as well as the two more important techniques that we employ, i.e., HMMs and MCTS. Section 3 provides experimental results and evaluations of our learner modeling method by comparing analysis results of original data with data generated post-reinforcement learning. Section 4 presents the discussion and conclusions.

2. BACKGROUND

We implement the learner modeling methods starting from data collected from student work in the Betty's Brain OELE. Betty's Brain is a learning by teaching environment, where students utilize tools for *information acquisition*, *solution construction* and *solution assessment* to teach a virtual character named Betty by constructing a causal map [5]. The primary student actions in the Betty's Brain environment can be categorized as:

Information Acquisition (***IA***): It relates to actions, such as reading to learn new information (***read***) and searching for specific knowledge ***search***. Taking and viewing notes is also considered to be useful for information acquisition (***notes***).

Solution Construction (***SC***): In Betty's Brain, SC actions are causal map editing actions (***mapedit***), which include addition and deletion of concepts and adding,deleting or changing links in the causal map.

Solution Assessment (***SA***): It consists of asking Betty to take a quiz(***quiz***); answer questions (***query***); and to explain how she derived her answers using qualitative reasoning methods (***expl***). Besides, students can mark correctness of links that have been added to assist their solution assessment.

Students' performance is based on a map score that is computed by comparing their causal models with a pre-specified expert model. In our study, the expert model had 15 links, which implies that the students could achieve a max map score of 15. At any time, the students' map score is computed by number of correct links minus number of incorrect links in their constructed (partial) maps. Next, we describe the learner modeling approach applied to Betty's Brain.

2.1 General Approach

Figure 1 illustrates the general approach that we have developed for our learner modeling method. As a first step, we apply a HMM clustering method [6] that divides the student' behaviors into groups of similar behaviors. We then iteratively generate a more accurate HMM model for each group by running a MCTS algorithm that combined with a reinforcement learning approach to produces a number of additional student behavior sequences that provides more coverage of the students' learning behaviors. These additional sequences when combined with the original student data is used to learn a new HMM model that we believe is a more complete description of the students' learning behaviors.

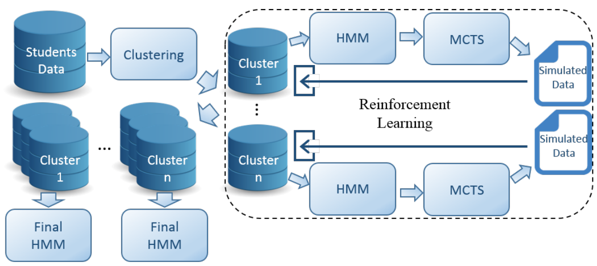


Figure 1: Architecture of the Overall Approach

2.2 HMM applied to Learner Modeling

A HMM is defined as a tuple, i.e., $\lambda = \{\mathbf{A}, \mathbf{B}, \pi\}$, where $\mathbf{A}$ and $\mathbf{B}$ represent state transition probability distribution and emission probability distribution matrices, respectively, while π is the initial state probability distribution [7]. Figure 2 presents the state diagram of a simple HMM example trained on two action sequences S_1 and S_2 with only 4 action types. Although not explicitly shown in the action sequences, the hidden states h_1 and h_2 can be interpreted as IA state (searching for and reading resources) and SC state (editing concept entities and causal links) respectively.

Based on the different probability distribution for each observation (action), the hidden states can be labeled by the primary actions associated with that state. The transitions between states capture changes in student activities over time, as also frequent patterns of activities, e.g., frequent occurrence of information acquisition followed by solution construction patterns.

2.3 Reinforcement Learning using MCTS

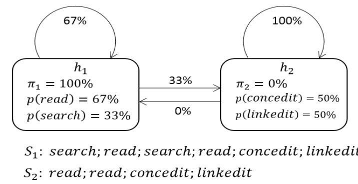


Figure 2: Simple HMM example.

To learn accurate and robust HMMs, it is important that the data set cover the range of behaviors a student exhibits in sufficiently large numbers.. However, given that we have limited student activity data on the system, we suffer from the data impoverishment problem. To address this problem, we propose a novel reinforcement learning method using Monte Carlo Tree Search (MCTS) and combine it with an initially derived HMM model to generate artificial data that matches students' learning behaviors. For generating action sequences that simulate actual students' behavior, we build the MCTS tree and traverse it to iteratively pick the next best node (with highest number of simulations) as the new action and add it to the tail of the sequence. In the reinforcement learning process as illustrated in Figure 1, we repeatedly generate simulated action sequences that maximize a specified reward function, and add them to the previously generated data. The reinforced data set is used to construct a refined version of the HMMs.

MCTS performs an iterative search with each iteration consists of 4 steps, i.e., ***Selection***, ***Expansion***, ***Simulation*** and ***Backpropagation*** [3]. In most MCTS implementations, the Upper Confidence bounds applied to Trees (UCT) algorithm is applied as the reward function for ***Selection***:

$$UCT = \frac{w_i}{n_i} + c\sqrt{\frac{\ln t}{n_i}} \tag{1}$$

where n_i is the number of simulations performed after adding the ith action; c is the exploration parameter with a typically chosen empirical value of $\sqrt{2}$; t is the total number of simulation runs for the parent node, which is equal to the sum of all the n_i; w_i is the sum of wins (1's) for all simulations after adding the ith action.

We adopt a similar reward function and compute the w_i value for generating action sequences that form a *Reinforced scaffolding model*. In this model, the normalized simulation results in the range of lowest-to-highest performance measure are summed up to compute w_i. For example, an action sequence has $w_i = 1$ when it achieves the max map score (i.e., 15) in Betty's Brain. This allows MCTS to better utilize coherence relations [8] to generate action sequences with more effective SC actions. The resulting HMM will favor the use of more coherent actions and be able to capture evolvement of learning behaviors/strategies that lead to better learning performance. Such behavioral and strategic evolvements can provide the basis for adaptive scaffolding.

We use the HMM to constrain the ***Expansion*** and ***Simu-***

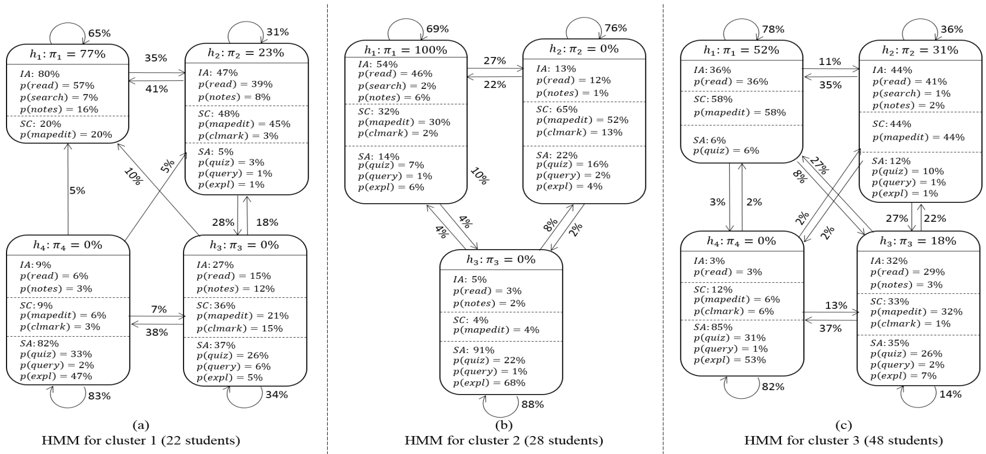


Figure 3: HMMs for the three clusters

Table 1: Comparison of the Three Clusters

	IA state	SA state	Balanced IA&SC state	Balanced SC&SA state	Search & Note Actions rate	Better strategic state transitions	$\overline{S_g}$	$\overline{S_m}$
Cluster 1	h_1	h_4	h_2	h_3	High	Yes	6.22	7.5
Cluster 2	h_1	h_3	-	-	Low	No	2.85	-2.25
Cluster 3	-	h_4	h_2	h_3	Low	Yes	5.61	3.79

lation steps to prevent expanding unvisited nodes and associated actions that are are not likely to occur in a given state. With these simulation and expansion policies, we can always generate action sequences that fit the HMM within a specified variance range. Figure 4 shows a simple example of generating artificial action sequence by applying MCTS.

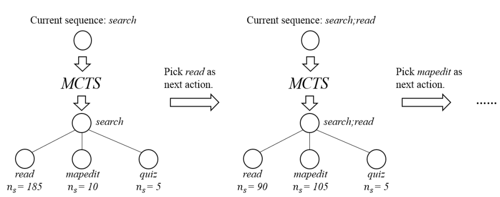


Figure 4: Simple example of applying MCTS for generating action sequence. n_s is the number of simulations performed during MCTS.

3. EXPERIMENTS AND ANALYSIS

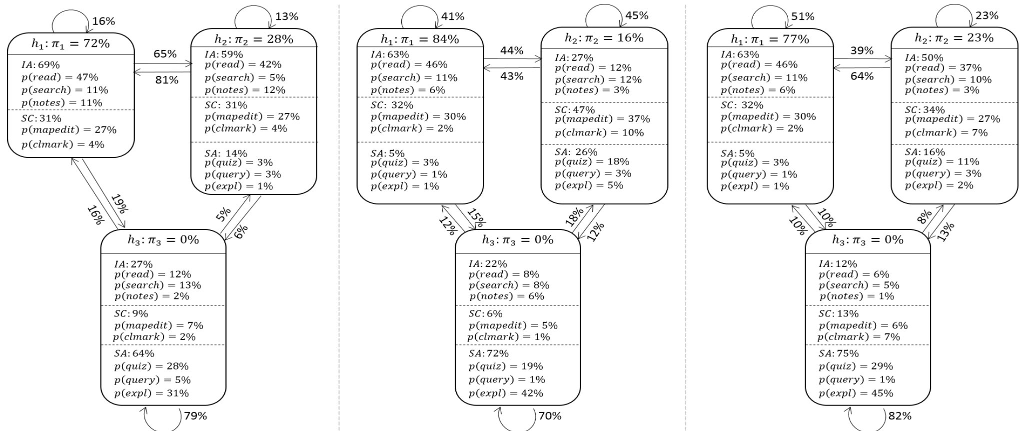
We use data from a Betty' Brain study run with 98 6th grade middle school students in a science classroom for our experiments. A HMM clustering algorithm [6] is applied to discover groups of action sequences with high within-cluster homogeneities. This algorithm produced 3 clusters with the highest Partition Mutual Information value. HMMs for the three clusters are represented by the state diagrams shown in Figure 3, where h_i represents the ith hidden state with corresponding initial probability π_i. State transition probabilities are marked on the transition links while emission probability of an action a in a state diagram is given by $p(a)$. For measuring students performance in the different clusters, we denote the average pre- and post-test score gain as $\overline{S_g}$ and denote the average final causal map score of the group as $\overline{S_m}$. We combine this information to interpret and compare students' behaviors in the three different groups as shown in Table 1.

As we can see from Table 1, all three clusters have a SA state (primarily focusing on SA actions). However, Cluster 3 doesn't have an IA state, while Cluster 2 doesn't have states that balances efforts between IA & SC, and SC & SA. These balanced efforts are aimed to use acquired information or solution assessment results to support subsequent SC actions. Besides, only Cluster 1 maintains a good proportion of Search & Note actions which are considered to be more active as for acquiring information. Students in Cluster 1 and 3 did better in strategic state transitions, while for Cluster 2, self transitions dominated in all states. The performance measures of students in Cluster 1, i.e., $\overline{S_g}$ and $\overline{S_m}$, are the best among all three clusters.

3.1 Reinforced Scaffolding Model Analysis

The reinforced scaffolding model as described in section 2.3 is aimed to capture useful behavioral and strategic evolutions. To validate it, we analyze the generated reinforced HMMs along with artificial action sequences that equal the sample size of original data set. The reinforced HMMs are shown in Figure 5.

Compared to the original HMMs (Figure 3), the HMMs for the three clusters gradually converge to a isomorphic 3-state HMM structure. The differences between original and refined HMMs can be summarized as (1) the HMMs tend to redistribute the efforts made between IA & SC, as well as SC & SA, e.g., the proportion of IA in h_1 is decreased for



(a) Reinforced HMM for cluster 1 | (b) Reinforced HMM for cluster 2 | (c) Reinforced HMM for cluster 3

Figure 5: Reinforced Scaffolding HMMs for the three clusters

cluster 1 but it is increased for the other two clusters. Given the probability of IA supporting SC, $P_{ia-sc} = 0.43 \approx 3:7$ according to statistics, the reinforced HMMs tend to have all SC actions to be supported by at least one IA action by converging emission probability of IA and SC towards a ratio of 70% : 30%. This is because the SC actions being supported by IA actions have higher probability to be effective (the ratio for unsupported:supported *mapedit*s to be correct is 0.41 : 0.53); and (2) the usage frequency for actions, such as *search*, increase significantly, especially for clusters 2 and 3. An explanation for this phenomena is that in the few cases that *search* appeared in the original data set, it is very likely followed by a *read* that supports a subsequent *mapedit*. The original HMM captures this pattern by having a hidden state h_s with relatively high emission probability for *search*, *read* and *mapedit*. When it expands to a node with *search* action during MCTS, the posterior probability for the hidden state to remain in h_s is high and, therefore, further expansion can form this specific pattern and result in a higher chance of correct *mapedit*. Since the reward function is designed to optimize the causal map score, the reinforcement learning is likely to follow this pattern more frequently when generating artificial action sequences.

4. DISCUSSION AND CONCLUSIONS

In this paper, we proposed a novel reinforcement learning method for learner modeling, which integrated Hidden Markov Model and Monte Carlo Tree Search within a Reinforcement learning framework to generate more accurate learner models for groups of students. We applied the HMM clustering algorithm to divide students into groups based on their behaviors. Analysis and interpretation on these groups are presented to explain the clustering results.

We then used data of student activities collected from a study with the Betty's Brain OELE and generated reinforced data sets along with the *Reinforced scaffolding model*. The experiments showed promising results according to our interpretation, where we were able to generate and interpret reinforced HMMs by analyzing evolvements of learning behaviors that can lead to better performance in building causal maps.

In future work, we will develop scaffolding methods to support students' learning new, more productive behaviors and strategies as they work on the system. And it will be of interest to study how our reinforcement learning method works with longitudinal studies on students and collect data across longer periods of time to generate dynamic coherence models. Besides, we will collect data from other learning environments, or even data from other domains to see how well our modeling methods perform.

5. REFERENCES

[1] G. Biswas et al. From design to implementation to practice a learning by teaching system: Betty's Brain. *International Journal of Artificial Intelligence in Education*, 26(1):350–364, 2016.

[2] J. E. Brophy. *Motivating students to learn*. Routledge, 2013.

[3] C. B. Browne et al. A survey of Monte Carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43, 2012.

[4] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[5] K. Leelawong and G. Biswas. Designing learning by teaching agents: The Betty's Brain system. *IJ Artificial Intelligence in Education*, 18(3):181–208, 2008.

[6] C. Li and G. Biswas. A Bayesian approach to temporal data clustering using Hidden Markov Models. In *ICML*, pages 543–550, 2000.

[7] L. R. Rabiner. Readings in speech recognition. chapter A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, pages 267–296. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

[8] J. R. Segedy, J. S. Kinnebrew, and G. Biswas. Using coherence analysis to characterize self-regulated learning behaviours in open-ended learning environments. *test*, 2(1):13–48, 2015.