

Mining Student Behavior Models in Learning-by-Teaching Environments

Hogyeong Jeong and Gautam Biswas

{hogyeong.jeong, gautam.biswas}@vanderbilt.edu

Department of Electrical Engineering and Computer Science, Vanderbilt University

Abstract. This paper discusses our approach to building models and analyzing student behaviors in different versions of our learning by teaching environment where students learn by teaching a computer agent named Betty using a visual concept map representation. We have run studies in fifth grade classrooms to compare the different versions of the system. Students' interactions on the system, captured in log files represent their performance in generating the causal concept map structures and their activities in using the different tools provided by the system. We discuss methods for analyzing student behaviors and linking them to student performance. At the core of this approach is a hidden Markov model methodology that builds students' behavior models from data collected in the log files. We discuss our modeling algorithm and the interpretation of the models.

1 Introduction

We have been developing computer environments for middle school students that use the learning by teaching paradigm to help middle school students learn metacognitive strategies while learning in science domains [1]. To teach, one must gain a good understanding of the domain knowledge and then structure that knowledge in a form that they can present to others [8]. Preparing to teach is a self-directed and open-ended activity where one explores, integrates, and structures knowledge first for oneself, and then for others. In addition to preparatory activities, teachers answer questions, provide explanations during teaching, and receive feedback from their students.

In our teachable agent (TA) environment, students teach a computer agent using well-structured visual representations. Using their agent's performance (which is a function of how well it was taught) as a motivation, students work to remediate the agent's knowledge, and, in this process, they learn better on their own [1]. An important property of our TA environments is that students can monitor how their agents answer questions and solve problems, and they may correct them when they notice discrepancies between their own knowledge and the agent's. Moreover, students' errors come more from a conflict between students' background knowledge and the procedures involved than the difficulty of using such a system [2]. Therefore, these environments (provided that they are well-designed) could help students become more knowledgeable of and responsible for their own cognition and reasoning. As a result, students are likely to develop problem solving and monitoring skills that go beyond the learning of domain specific content; rather these environments provide the much larger framework that guide students on *how to learn* and *how to prepare for future learning* [10].

We have run a number of experimental studies to understand the cognitive and metacognitive processes that help students learn with understanding in the TA environments. The

studies have contrasted situations where students either learn by teaching or are taught by a mentor agent. In one condition, students receive metacognitive prompts as they learn by teaching. This paper investigates the use of data mining methods to systematically build and analyze models of student behaviors as they work in the TA environments. Traditional student modeling methods have focused on capturing students' knowledge of the domain. In contrast, our emphasis on preparation for future learning has directed us to focus on strategies and behaviors that students employ during the learning process. This paper approaches student modeling by analyzing student behaviors from their activity sequences, and linking these behaviors to learning strategies.

1.1 Learning by teaching: The Betty's Brain system

The Betty's Brain system is illustrated in Fig. 1. The teaching process is implemented as

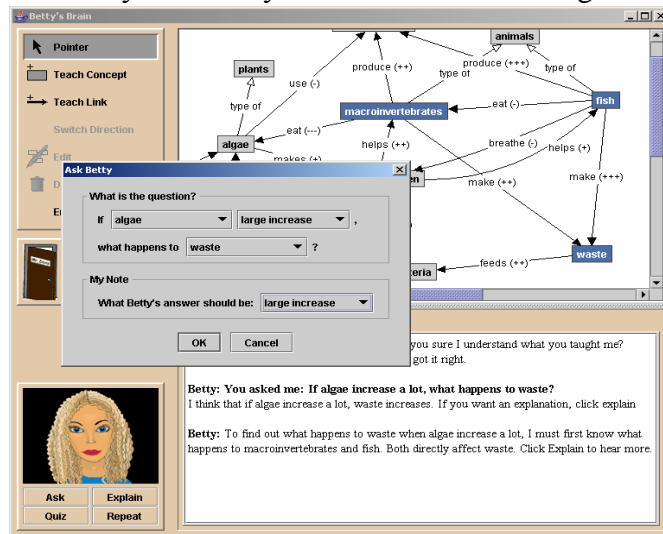


Figure 1: Betty's Brain System with Query Window

three primary activities: (i) *teach* Betty concepts and links using a concept map representation [6]; (ii) *query* Betty to find out how she answers questions using what she has been taught; and (iii) *quiz* Betty on a set of predefined questions generated by the mentor agent to see how she performs.

Betty uses qualitative reasoning methods to reason through chains of links to answer questions, and, if asked, explains her reasoning using text and animation schemes. She also provides feedback that reflects the students' teaching behaviors.

The goal is to get the students to adopt metacognitive strategies in their learning tasks [11]. Students reflect on Betty's answers and her explanations, and revise their own knowledge as they make changes to the concept maps to teach Betty better.

1.2 Experimental Design

Our participants were 56 students in two 5th grade science classrooms taught by the same teacher. Data for three students was dropped from the main study, and three more were dropped from the transfer study due to excessive absences. Students were assigned to one of three conditions using stratified random assignment based on standardized test scores. The students first created concept maps on river ecosystem concepts and causal relations in the main phase (seven 45-minute sessions) of the study. Depending on their assigned group, students utilized one of three versions of the system: a) a learning by teaching (LBT) version in which students taught Betty, b) a self-regulated learning by teaching (SRL) version in which students taught Betty and received metacognitive prompts from Betty, and c) an intelligent coaching system (ICS) version in which students created a map for themselves with guidance from the mentor agent. The ICS system served as our control condition [3]. After an eight-week delay, students participated in the transfer

phase (five 45-minute sessions) in which they learned about a new domain, the land-based nitrogen cycle. During this phase, all students used a stripped-down version of the LBT system in which all of the feedback provided by Betty or the mentor was removed.

1.3 Results

We scored the students' final main and transfer concept maps to identify correct inclusions of concepts and links based on the resources that were provided to the students. Table 1 shows the average concept map scores by condition for the two phases of the study. Students who taught developed more complete and interconnected concept maps than students who created maps for themselves. The differences in map scores are statistically significant (SRL > LBT, ICS; LBT > ICS; $p < 0.05$) in the main phase, and the difference between SRL and ICS persisted during the transfer phase.

Table 1. Concept map score: main and transfer phases

Condition	Mean (SD) Map Scores	
	Main Phase	Transfer Phase
ICS (n = 17)	22.83 (5.3)	22.65 (13.7)
LBT (n = 15)	25.65 (6.5) ^c	31.81 (12.0)
SRL (n = 18)	31.58 (6.6) ^{a,b}	32.56 (9.9) ^a

^a SRL > ICS, $p < .05$; ^b SRL > LBT, $p < .05$; ^c LBT > ICS, $p < .05$.

We also performed preliminary analyses of students' behavior sequences to shed light on their learning processes [1]. We found that students who generated better concept maps used balanced learning strategies: they read resources, built and edited their concept maps, asked queries to probe the correctness of their maps, and used quiz questions in equal measure to learn about the domain. On the other hand, students who generated low scoring concept maps adopted a myopic learning strategy that overly focused on getting their quiz answers correct. They relied mostly on the quiz questions, and did not seem to read resources or probe their maps by asking queries very often.

1.4 Model Selection

To investigate this relation between learning performance and the use of strategies by group, it became important to go beyond frequency counts or proportions of individual activities, and examine how these activities came together as larger behavior patterns and strategies. To this end, we found the hidden Markov models (HMMs) to be the most appropriate, as they allow us to identify some of the students' general behavior patterns from sequences of their interactions with the system..

2 A Data Mining Approach to Analyzing Student Behaviors

Our approach to analyzing student behaviors in the main and transfer phases of the study involves four steps that appear in most data mining applications [12]: (i) devise a logging system that records student interactions with the system; (ii) perform data cleaning by parsing the generated log files and splicing the information into desired activity sequence data that will form the input to the HMM generating algorithm; (iii) construct the HMM

models; and (iv) interpret generated models as student learning behaviors and compare models across conditions. We describe each of these steps in greater detail below.

2.1 Generating the Raw Data: The Logging System

The raw data corresponds to actions performed by students on the system and the responses provided by Betty and the mentor agent. The agents and the student interact by communicating through the system environment agent. For example, if the student asks Betty a question, the request is broadcast to the environment agent, who routes it to other agents. The left side of Fig. 2 illustrates the message passing that takes place in the system. The signal first broadcast to the environment agent is routed to the teachable agent, who makes a request to the environment agent to use the current concept map to answer the question. This raw data is then processed in real-time and stored in log files as shown in the right side of Fig. 2. At the end of a session, the logs are sent to a server and consolidated into a session by session sequence for each student into a database.

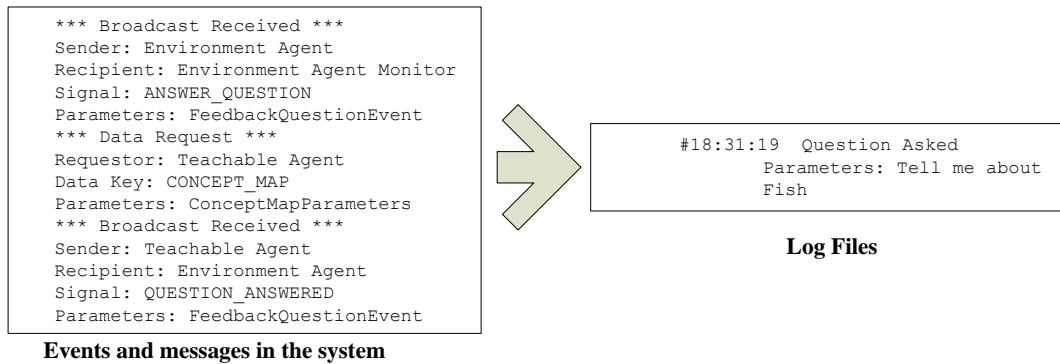


Figure 2: Events and messages being turned to log files

2.2 Parsing the Log Files

In this study, we derive students' behavior patterns by analyzing the sequence of their interactions with the system over multiple sessions. To simplify the interpretation task we lumped analogous student actions into one aggregate activity. For example, all of the map

Table 2: Student Activities and Related Actions

Activity	Student Actions
Edit Map (EM)	adding, modifying, or deleting concepts and links
Ask Query (AQ)	asking Betty queries
Request Quiz (RQ)	asking Betty to take the quiz
Resource Access (RA)	accessing the resources
Request Explanation (RE)	asking Betty for an explanation to her query answer
Continue Explanation (CE)	asking Betty to provide a more detailed explanation
Quiz Denied (QD)	asking Betty to take the quiz without adequate preparation
Quiz Taken (QT)	asking Betty to take the quiz with adequate preparation

creation and modification activities, i.e., adding concepts and links, deleting concepts and links, and editing nodes and links, were combined into one aggregate activity called Edit Map (EM). All student activities were expressed as the eight activities summarized in Table 2.

```
S5213, 1, RA, EM, EM, EM, EM, EM, EM, AQ, AQ, AQ, EM, EM, AQ, EM, EM
S5254, 1, RA, RA, RA, RA, EM, EM, EM, AQ, AQ, EM, EM, EM, EM, EM, EM
S6236, 1, RA, EM, EM, EM, AQ, EM, EM, EM, EM, EM, EM, EM, EM, EM, EM, RQ, QT, EM, EM, AQ, AQ
```

Figure 3: Parsed Data

2.3 Constructing the HMMs

The first step in interpreting this behavior data was to generate hidden Markov models. A hidden Markov model consists of hidden states that are not directly visible, and are governed by three sets of parameters: initial probability vector π , transition probability matrix, A , and output probability matrix, B [7]. By representing concise models of student activity patterns, a hidden Markov model has the potential of providing us with a global aggregated view of how students approach the learning task [3].

The algorithm that constructs HMMs given a set of observation sequences derives an optimal set of these parameters (π , A , B) that maximizes the likelihood of the input sequences. Further, simpler models are easier to interpret (Occam's razor principle), so we apply an algorithm developed by Li and Biswas [5] that uses the Bayesian information criterion (BIC) to trade off simplicity of the model against information provided by the model. BIC, defined as $\log(L) - d/2\log(N)$, uses the log likelihood of the model ($\log(L)$), model size (d), and the number of observations (N) to find the model that strikes a balance between high likelihood and low complexity [5].

Finding the optimal HMM parameters from data is an optimization problem. Two common iterative convergence optimization schemes are the Baum-Welch [7] and the segmental K-Means [4] algorithms. In this paper, we use the segmental K-Means algorithm in conjunction with BIC for iterative segmentation and optimization steps to achieve the optimal model parameters, which include (π , A , B) and the number of states in the model, d . The segmentation step uses the Viterbi algorithm for sequential decoding, while the optimization step finds a new set of model parameters as dictated by the K-Means method [4]. A chief advantage of the K-Means algorithm is its faster execution time gained by setting a restricted optimization objective. In the future, the faster speed may allow on-line computation of the behavior models to provide guided feedback as the student works on the system.

A concern, however, for the segmental K-Means algorithm is the likelihood of its convergence to local maxima. In this work, we ran the algorithm one hundred times with random initializations (by sampling the initial parameter values from uniform distributions), all of which converged to the same configuration. We also compared the BIC values generated by the Baum-Welch and the segmental K-Means algorithms, and found that the K-Means algorithm produced consistently better results. While these empirical results do not conclusively dispel the fact that our solutions may converge to local maxima, they nevertheless show the algorithm to be quite robust.

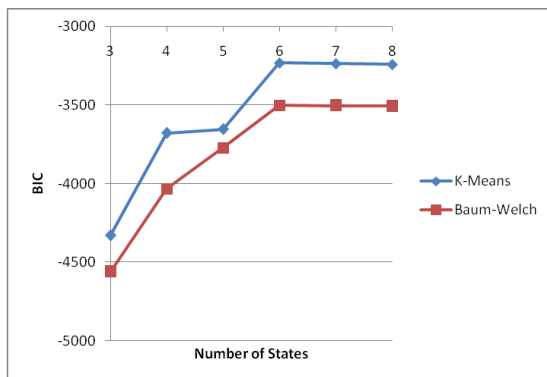


Figure 4: K-Means and Baum-Welch-generated BIC values for the ICS Model

states. For example, the model predicts that students in the ICS condition RQ(75%)CE(25%) state requested the quiz 75% of the time, and asked for a continued explanation 25% of the time. The transition probability associated with a link between two states indicates the likelihood of the student transitioning from the current state to the indicated state. For example, the HMM model states student in the ICS condition in state RQ(75%)CE(25%) of the main phase would demonstrate a 6% likelihood of transitioning to state EM, a 19% likelihood of remaining in the same state, and 71% likelihood of transitioning to state QT. Likelihoods less than 2% were not represented in the figure, explaining why these numbers do not sum to 100%. HMMs are so named because their states are hidden. That is, they are not directly observed in the input sequences, but provide an aggregated description of the students' interactions with the system. Sequences of states may be interpreted as the students' learning behavior patterns. We investigate further by interpreting these models in terms of cognitive learning behaviors of the students.

In looking at these HMMs, we find several interpretable patterns that present themselves through high intrastate transition probabilities and low interstate transition probabilities. For example, we see that the transition probabilities from states with Request Explanation (RE) to Continue Explanation (CE) are strong ($\geq 49\%$). Also, we see that these states are quite isolated, as the transition probabilities into these two states from other states are typically small (only 7% in the transfer SRL model). We combine these observations with knowledge of patterns relevant to interactive metacognition to find three patterns: *basic map building, map probing, and map tracing* [3].

Basic map building is a pattern characterized by editing the map, submitting the map for a quiz, and occasionally referring to the reading resources. The pattern reflects a basic and important metacognitive strategy. Students work on their maps, check the map by taking a quiz to see if there are flaws, and occasionally refer to the readings.

Map probing is defined by students asking questions of their map to check for specific relations between two concepts (e.g., if fish increase, what happens to algae?). This pattern exhibits a more proactive, conceptually driven strategy, because students are targeting specific relations rather than relying on the quiz to identify errors. Students also need to formulate their own questions to do so.

The parsed activity sequences were used to derive two sets of three hidden Markov models for the three conditions using the above algorithm.

3 Analysis of the HMMs

The behavior sequence models for the ICS, LBT, and SRL groups in the main and transfer study created using our HMM algorithm are shown in Fig. 5. Each model is made up of a set of states, the activity patterns (the output probability) associated with each state, and the transition probabilities between

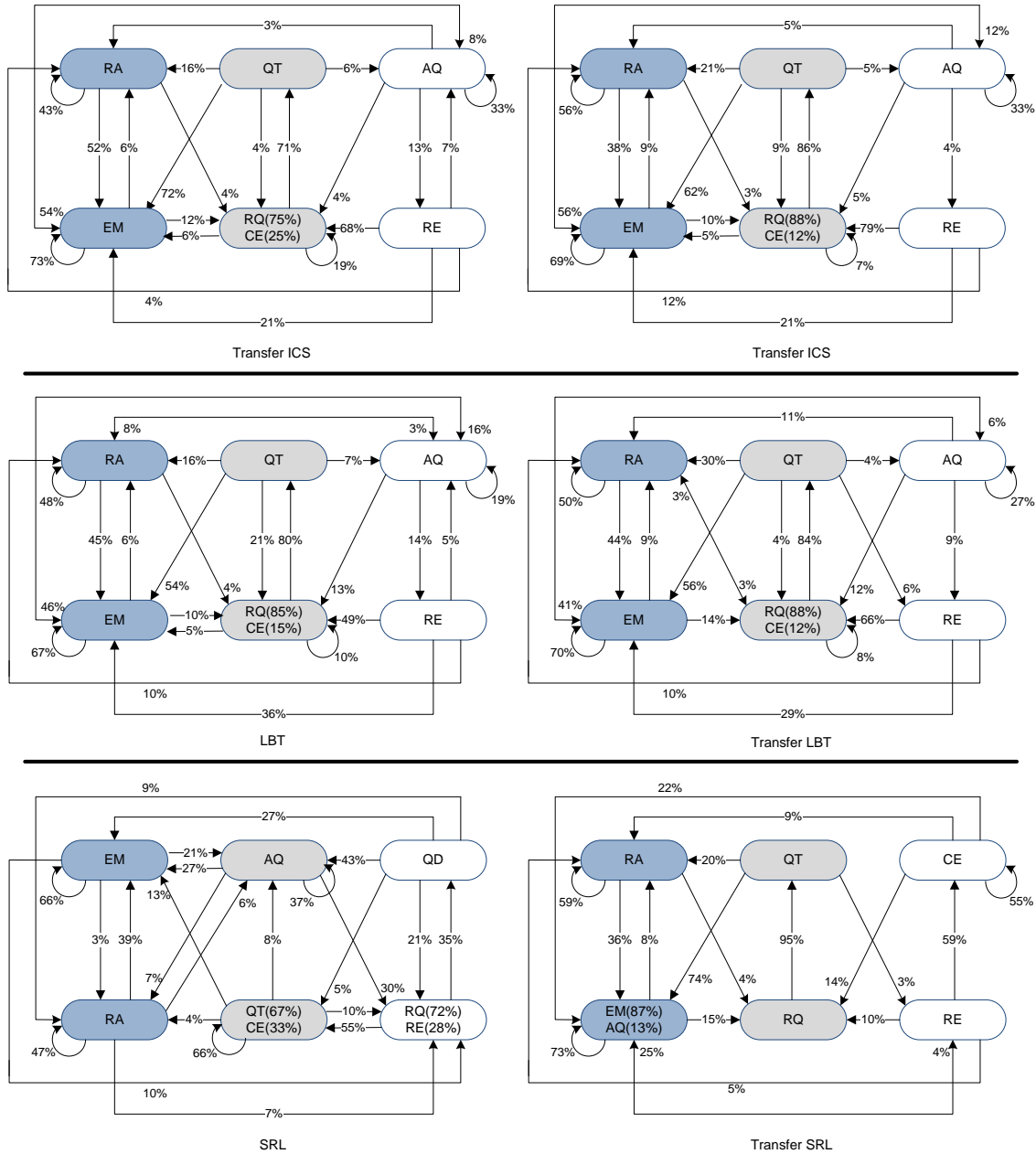


Figure 5: HMM Models for the three conditions in the main and transfer phase

Map tracing pattern reflects students asking Betty or the mentor (depending on the system) to explain the reasoning step by step. When Betty or the mentor initially answers a question, they state that a change in one entity causes a change in another entity and highlight the paths they followed to reach their answer. To follow the details of the inference chain, students had to ask Betty or the mentor agent to explain their reasoning. The agents did so by hierarchically decomposing the chain of inference; for each explanation request, they showed how a particular path within the larger chain contributed to the final answer. Receiving more details about the reasoning process is particularly useful when maps become complex, and there are multiple paths between two concepts.

To build reduced versions of HMMs that incorporate these patterns, we first built aggregate states that represented the patterns of its individual behaviors. For instance, Edit Map, Request Quiz, Quiz Taken, Quiz Denied, and Resource Access were combined into the basic map building state; Ask Query was treated as map probing state; and Request and Continue Explanation were combined into a map tracing state. The transitions were then constructed in accord to a proportional weighing of the individual behavior’s stationary probabilities.

3.1 Initial Analysis

Our preliminary analysis consisted of examining the prevalence of each behavior in the resultant stationary probabilities. In a sense, this first level of analysis is equivalent to the frequency count analysis that we have performed in some of our other studies [3], and similar results in both studies will help to validate our methods and results.

Table 3: Grouped Stationary Probabilities

State	ICS	LBT	SRL	Transfer ICS	Transfer LBT	Transfer SRL
Map Building	0.72	0.66	0.42	0.73	0.73	0.68
Map Probing	0.24	0.30	0.47	0.25	0.25	0.27
Map Tracing	0.04	0.04	0.11	0.02	0.02	0.05

In the main phase, we see that the differences in stationary probabilities among the groups are quite pronounced. For example, we see that there exists a significant drop-off in the probabilities of the students’ edit map behaviors between successive conditions. Meanwhile, we see proportional increases in activities belonging to higher-level patterns, such as requesting explanation, and continuing explanation. The students’ use of continue explanation is especially pronounced.

In the transfer phase when the students operate in a common environment, we see that the differences become smaller. In all three groups, we see a great spike in the number of resource accesses relative to the main phase. At the same time, we observe a decrease in the occurrence of some of the higher-level patterns. This may be due to the students learning about a new domain with a simpler expert map that contains fewer interdependences between concepts, and being given few sessions than in the main phase (five versus seven). This also could be due to the students internalizing the reasoning mechanism, therefore, a reduced need to perform such activities in the transfer phase [3].

3.2 Analysis of the HMM Patterns

Our next level of analysis consisted of examining the interactions among states in our models. These interactions inform us as to how the students typically use the system in regards to the behavior patterns that we have identified.

In the main phase, we find that the students in the ICS group tend to stay mainly in the basic map building state, while the SRL students tend to stay more in the higher-level states once they get there. For example, the self-loop vector for the map tracing state is

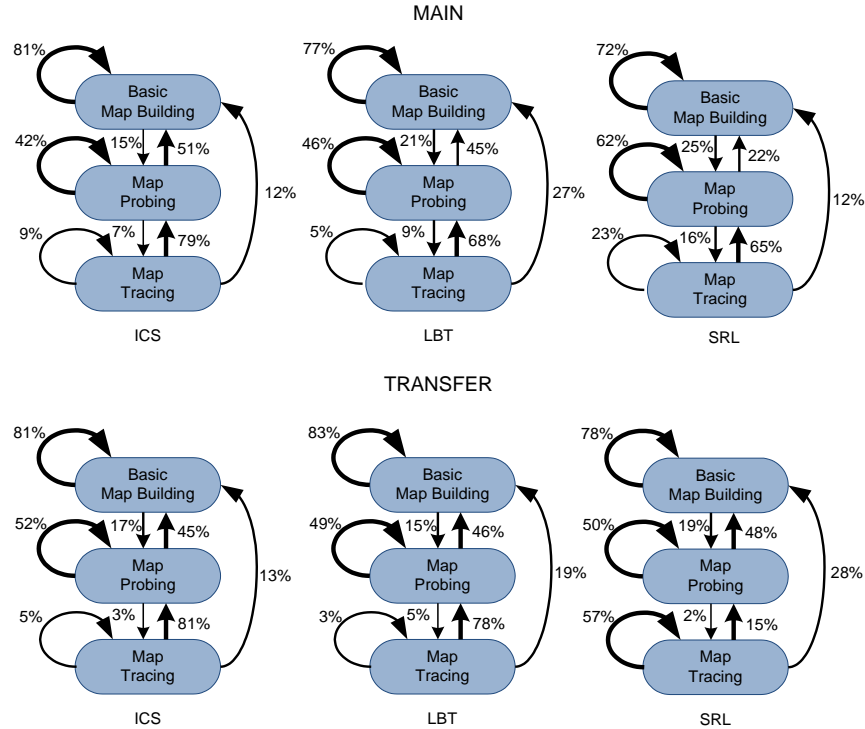


Figure 5: HMM Patterns showing the ICS, LBT and SRL maps in the main and transfer study

much larger in the SRL group (23%) than in the ICS or the LBT groups (9% and 5% percent, respectively). Also, while the differences between ICS and LBT seem to be small, the ICS students seem spend most of their effort in basic map building, while the LBT students do spend a little more time in map probing, a higher level metacognitive activity.

In the transfer phase, the difference between the ICS and the LBT group becomes harder to discern. However, like the main phase, the LBT students seem more likely to enter map tracing state than the ICS students (5% as opposed to 3%), but are more likely to leave once they get there. However, unlike in the main phase, the LBT students now seem to be more confined to the basic map building state than the ICS students (83% as opposed to 81%). However, the SRL students still perform more map probing than the other groups.

4 Conclusions and future work

Our data mining approach to building HMM models of student behaviors from log files have been very revealing. They have helped us establish that learning by teaching provides better opportunities for learning even among 5th grade students. Further, metacognitive prompts while learning enable students to develop higher level learning strategies that they retain even when the feedback prompts are removed. In the future, we will further refine our data mining techniques and algorithms to set up a framework for designing adaptive learning environments, where the learning support and feedback provided to students will be guided by the information derived from the student behavior models. We will also work further towards developing a more quantitative way of analyzing and

comparing the models (these may involve using distance metrics and more comprehensive cognitive learning patterns).

Acknowledgements

This work has been supported by a Dept. of Education IES grant #R305H060089 and NSF REESE Award #0633856. The authors wish to acknowledge the help provided by John Wagster and Rod Roscoe in the data collection, data analysis, and data interpretation tasks.

References

- [1] Biswas, G., Leelawong, K., Schwartz, D., Vye, N., The Teachable Agents Group at Vanderbilt. Learning by Teaching: A New Agent Paradigm for Educational Software. *Applied Artificial Intelligence*, 2005, 19, p. 363-392.
- [2] Easterday, M., Aleven, V., and Scheines R. The logic of Babel: Causal reasoning from conflicting sources. *Artificial Intelligence in Education*, Marina del Rey, California, 2007.
- [3] Jeong, H., Gupta, A., Roscoe, R., Wagster, J., Biswas, G., Schwartz, D. Using Hidden Markov Models to Characterize Student Behavior Patterns in Computer-based Learning-by-Teaching Environments. To appear in: *Intelligent Tutoring Systems*, Montréal, Canada, 2008.
- [4] Juang B., Rabiner L. The segmental k-means algorithm for estimating the parameters of hidden Markov Models. *IEEE Trans. Acoustics, Speech and Signal Processing*, 1990, 38(9) p. 1639-1641.
- [5] Li, C. and Biswas, G. A Bayesian Approach for Learning Hidden Markov Models from Data. *Special issue on Markov Chain and Hidden Markov Models, Scientific Programming*, 2002, 10, p. 201-219.
- [6] Novak, J.D. Learning, Creating, and Using Knowledge: Concept Maps as Facilitative Tools in Schools and Corporations. *Lawrence Erlbaum Associations*, 1998. Mahwah, NJ.
- [7] Rabiner L. R. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proc. IEEE*, 1989, 77(2), p. 257-286.
- [8] Roscoe, R. D., Chi, M. Understanding tutor learning: Knowledge-building and knowledge-telling in peer tutors' explanations and questions. *Review of Educational Research*, 2007, 77(4), p. 534-574.
- [9] Roscoe, D., Wagster, J., and Biswas, G., Using Teachable Agent Feedback to Support Effective Learning by Teaching, in review, Cognitive Science Conference, Washington, DC, July 2008.
- [10] Schwartz, D.L. and Martin, T., Inventing to Prepare for Future Learning. *Cognition and Instruction*, 22(2), p. 129-184, 2004.
- [11] Wagster, J., Tan, J., Wu, Y., Biswas, G., Schwartz, D. Do Learning by Teaching Environments with Metacognitive Support Help Students Develop Better Learning Behaviors? *The twenty-ninth Annual Meeting of the Cognitive Science Society, Nashville, Tennessee*, 2007, p. 695-700.
- [12] Whitten, I.H.. and Berry, M., *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed, Morgan Kaufmann., Los Altos, CA, 2005.