A Markov Decision Process Model of Tutorial Intervention in Task-Oriented Dialogue

Christopher M. Mitchell, Kristy Elizabeth Boyer, and James C. Lester

Department of Computer Science, North Carolina State University,
Raleigh, North Carolina, USA
{cmmitch2, keboyer, lester}@ncsu.edu

Abstract. Designing dialogue systems that engage in rich tutorial dialogue has long been a goal of the intelligent tutoring systems community. A key challenge for these systems is determining when to intervene during student problem solving. Although intervention strategies have historically been hand-authored, utilizing machine learning to automatically acquire corpus-based intervention policies that maximize student learning holds great promise. To this end, this paper presents a Markov Decision Process (MDP) framework to learn an intervention policy capturing the most effective tutor turn-taking behaviors in a task-oriented learning environment with textual dialogue. The model and its learned policy highlight important design considerations, including maintaining tutor engagement during student problem solving and avoiding multiple consecutive interventions.

Keywords: Tutorial Dialogue, Markov Decision Processes, Reinforcement Learning

1 Introduction

The effectiveness of tutorial dialogue has been widely established [1, 2]. In recent years, reinforcement learning (RL) has proven useful in the analysis and creation of tutorial dialogue system behaviors in structured interactions [3, 4]. Extending this prior work, this paper presents a novel application of RL to a corpus of textual tutorial dialogue. In particular, the focus here is automatically learning intervention strategies from a fixed corpus of human-human task-oriented tutorial dialogue with unrestricted turn-taking. The presented approach and policy results can inform the development of tutorial dialogue systems whose policies are acquired automatically based on fixed corpora.

The corpus analyzed in this paper consists of 66 text-based tutorial dialogues between first-year university students and experienced tutors as the students worked to solve introductory computer science problems. Each student-tutor pair collaborated using the JavaTutor remote interface [5], which supports textual communication between the tutor and student as well as giving the tutor a real-time synchronized view of the student's workspace. Over the course of a 40-minute session, each student endeavored to build a working program using the Java programming language. In

order to measure the effectiveness of each session, students completed a pre-test and post-test. Students scored significantly higher on the post-test than the pre-test ($p < .001$). We computed normalized learning gain, which can range from -1 to 1. In the present study normalized learning gains ranged from -0.29 to 1 (mean = 0.42; median = 0.45; st. dev. = 0.32).

2 Building the Markov Decision Process and Policy Learning

From the tutors' perspective, the decision to intervene was made based on the state of the interaction as observed through the two information channels in the interface: the textual dialogue pane and the synchronized view of the student's workspace. In order to use a MDP framework to derive an effective intervention policy, we describe a representation of the interaction state as a collection of features from these information channels.

A Markov Decision Process is a model of a system in which a policy can be learned to maximize reward [6]. It consists of a set of states S, a set of actions A representing possible actions by an agent, a set of transition probabilities indicating how likely it is for the model to transition to each state $s' \in S$ from each state $s \in S$ when the agent performs each action $a \in A$ in state s, and a reward function R that maps real values onto transitions and/or states, thus signifying their utility.

The goal of this analysis is to model tutor interventions during the task-completion process, so the possible actions for a tutor were to intervene (by composing and sending a message) or not to intervene. Hence, the set of actions is defined as $A = \{TutorMove, NoMove\}$. We chose three features to represent the state of the dialogue, with each feature taking on one of three possible values. These features, described in Figure 1, combine as a triple to form the states of the MDP as (Current Student Action, Task Trajectory, Last Action). In addition, the model includes 3 more states: an *Initial* state, in which the model always begins, and two final states: one with reward +100 for students achieving higher-than-median normalized learning gain and one with reward -100 for the remaining students, following the conventions established in prior research into reinforcement learning for tutorial dialogue [3, 4].

Current Student Action	Task Trajectory	Last Action
Task: Working on the task	***Closer***: Moving closer to the final correct solution	***TutorDial***: Tutor message
StudentDial*:* Writing a message to the tutor	***Farther***: Moving away from correct solution	***StudentDial***: Student message
NoAction: No current student action	***NoChange***: Same distance from correct solution	***Task*****:** Student worked on the task

Fig. 1. The features used to define the states of the Markov Decision Process

Using these formalizations, one state was assigned to each of the log entries collected during the sessions and transition probabilities were computed between them when a

tutor made an intervention (*TutorMove*) and when a tutor did not make an intervention (*NoMove*). An excerpt from the corpus with these assigned states is shown in Figure 2.

Event	Tutor action and state transition
1. *Student is declaring a String variable named "aStringVariable".*	*NoMove* ↓ (Task, NoChange, Task)
2. *Tutor starts typing a message*	*TutorMove*
3. *1.5 seconds elapse, task action is complete.*	↓
4. **Tutor message:** That works, but let's give the variable a more descriptive name	(NoAction, Closer, TutorDial)
5. *Tutor starts typing a message*	*TutorMove*
6. *Student starts typing a message*	↓
7. **Student message:** ok	
8. **Tutor message:** Usually, the variable's name tells us what data it has stored	(NoAction, Closer, TutorDial)

Fig. 2. An excerpt from the corpus with state, action, and transition labels

In order to learn a tutorial intervention policy, we used a policy iteration algorithm [6] on the MDP. Some noteworthy patterns emerge in the intervention policy learned from the corpus. For example, in seven of the eight states where the student is actively engaged in task actions, i.e., matching the pattern (*Task*, *, *), the policy recommends that the tutor make a dialogue move. On its surface this policy may seem counterintuitive, since the student may be making task progress and there is a risk of interruption by the tutor. However, the policy suggests that sessions in which the tutor remained engaged in the problem-solving process by making dialogue moves as the student was working were more likely to produce high normalized learning gains.

Among the states in which no action is currently being taken by the student and the last action was a tutor message, i.e., matching the pattern (*NoAction*, *, *TutorDial*), we find that the policy recommends that a tutor not make another consecutive dialogue move, regardless of how well the student is progressing on the task. It is possible that consecutive tutor dialogue moves would present more information than a student could effectively process, thus leading to high cognitive load or disengagement for the student and, in turn, lower learning gains. While this could be interpreted as a recommendation for the tutor to be less talkative, the just-mentioned recommendation regarding continual tutor engagement during task completion would seem to contradict this interpretation. Instead, it is more likely that an effective tutor will compose messages such that they engage the student in dialogue or provide succinct guidance for the student to make progress on the task without additional intervention. Further investigation of the consequences of these recommendations will be addressed in future work.

3 Discussion and Conclusion

The model presented here demonstrates a novel approach to automatically determining an intervention policy for tutorial dialogue with unrestricted turn-taking from a fixed corpus using a reinforcement learning-based approach. The resulting policy provides insight into the effectiveness of tutor intervention decisions with respect to the success of a tutorial dialogue. We note the gap between the recommended action in the learned policy and the actual actions taken by tutors in the corpus: tutors follow the recommended *(Task, *, *)* policy only 11% of the time, while following the recommended (*NoAction*, *, *TutorDial*) policy slightly more than 43% of the time. Avoiding policies prevalent in sessions with lower learning gain is one of the key advantages of using reinforcement learning.

Further exploration of the state space via simulation and utilizing a more expressive representation of state are highly promising directions for future work. Other directions for future work include undertaking a more fine-grained analysis of the timing of interventions, which could inform the development of more natural interactions, as well as allowing for more nuanced intervention strategies. Additionally, these models should be enhanced with a more expressive representation of both dialogue and task. It is hoped that these lines of investigation will yield highly effective machine-learned policies for tutorial dialogue systems.

Acknowledgements

This work is supported in part by the National Science Foundation through Grants DRL-1007962 and CNS-1042468. Any opinions, findings, conclusions, or recommendations expressed in this report are those of the participants, and do not necessarily represent the official views, opinions, or policy of the National Science Foundation.

References

1. Bloom, B.: The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. Educational Researcher. 13, 4–16 (1984).
2. VanLehn, K., Graesser, A.C., Jackson, G.T., Jordan, P., Olney, A., Rosé, C.P.: When Are Tutorial Dialogues More Effective Than Reading? Cognitive Science. 30, 3–62 (2007).
3. Chi, M., VanLehn, K., Litman, D.J.: Do Micro-Level Tutorial Decisions Matter: Applying Reinforcement Learning To Induce Pedagogical Tutorial Tactics. Proceedings of the International Conference on Intelligent Tutoring Systems. pp. 224–234 (2010).
4. Tetreault, J.R., Litman, D.J.: A Reinforcement Learning Approach to Evaluating State Representations in Spoken Dialogue Systems. Speech Communication. 50(8), 683-696 (2008).
5. Grafsgaard, J.F., Fulton, R.M., Boyer, K.E., Weibe, E.N., Lester J.L.: Multimodal Analysis of the Implicit Affective Channel in Computer-Mediated Textual Communication. Proceedings of the International Conference on Multimodal Interaction. pp. 145-152 (2012).
6. Sutton, R., Barto, A.: Reinforcement Learning. MIT Press, Cambridge, MA (1998).