Adapting Difficulty Levels in Personalized Robot-Child Tutoring Interactions

Aditi Ramachandran
Yale University
51 Prospect St.
New Haven, CT 06511

Brian Scassellati
Yale University
51 Prospect St.
New Haven, CT 06511

Abstract

Social robots can be used to tutor children in one-on-one interactions. Because students have different learning needs, they consequently require complex, non-scripted teaching behaviors that adapt to the learning needs of each child. As a result of this, robot tutors are more effective given a means of adaptively customizing the pace and content of a student's curriculum. In this paper we propose a reinforcement learning-based approach that affords such capabilities to a tutoring robot, with the goals of fostering measurable learning gains and sustained engagement. We outline an architecture in which the robot uses reinforcement learning to adapt the difficulty of its exercises. Further, we describe a proposed study capable of evaluating the effectiveness of our Intelligent Tutoring System.

Introduction

The 2-Sigma problem refers to the phenomenon that students tutored one-on-one perform two standard deviations better than students learning via conventional classroom instruction on average (Bloom 1984). During one-on-one tutoring, the tutor can conform to the individual learning style and preferences of the student. Because the resources required to provide this tutoring to every student is unrealistic, we seek other methods of instruction that will have comparable effects to one-on-one human tutoring. Intelligent Tutoring Systems (ITSs) are computer systems that can perform student modeling, provide customized instruction to learners, and employ a variety of pedagogical strategies (Graesser, Conley, and Olney 2012).

Additionally, research involving robotic agents as tutors indicate that the physical presence of a robot tutor can increase cognitive learning gains (Leyzberg et al. 2010). This motivates the need to investigate robot tutoring systems as an effective method of instruction. Existing ITS studies show that there are many aspects of a tutoring interaction that can be personalized, particularly within finding pedagogical strategies tailored to students. One strategy is called curriculum sequencing, which deals with the ordering that the questions are presented in to maximize learning of the student (Giannandrea and Sansoni 2013).

We aim to use a social robot to provide personalized curriculum sequencing in a tutoring interaction with children in order to foster learning for the child, as well as maintain the child's engagement during the interaction.

User Modeling

To understand how social robots may be effectively used as tutoring agents, we examine existing work in the ITS community. Here we list some of the main modeling techniques in ITSs.

Cognitive Tutors: One of the earliest ITS frameworks to become popular is the family of Cognitive Tutors (CT). This approach uses the ACT-R cognitive architecture to model student knowledge (Graesser, Conley, and Olney 2012). This type of architecture constantly monitors a student to collect information about the student's behavior. It employs a process termed *model tracing* to compare a student profile to a cognitive model to assess whether the student requires help. While this detailed framework addresses a variety of aspects of a tutoring interaction, creating a model incorporating knowledge components in addition to a detailed model of the learner's behavior can be extremely tedious and require domain experts.

Bayesian Modeling: Many ITSs construct Bayesian Networks (BNs) for student modeling as it is a widely used technique for reasoning under uncertainty in a learning environment. Some ITSs use this technique to model the state of the learner, including their behavior and mental state (Conati et al. 1997). Alternatively, some systems construct a BN to observe student behavior and assess which rules the learner has successfully used in a given knowledge domain (Martin and VanLehn 1995). BNs can also be used to construct how the learner navigated through a given problem, and can identify concepts that the learner needs practice with. While BNs are a viable technique to use, they can often become extremely complex and difficult to evaluate. There is much research within this field detailing how the Bayesian networks for student modeling can be constructed in addition to how to choose prior and conditional probabilities for the model (Gonzalez, Burguillo, and Llamas 2006).

Reinforcement Learning: There has been work done in using Reinforcement Learning (RL) to model student learning patterns during a tutoring interaction. RL as part of an ITS is typically used for the pedagogical model within the system, which deals with how the knowledge is presented to the learner. Some RL tutoring systems focus on presenting questions such that they maximize a student's learning (Malpani, Ravindran, and Murthy 2011). RL has also been used to decide which teaching tactic to use at a given point within a tutoring interaction (Chi, VanLehn, and Litman 2010). One advantage to using RL systems in tutoring is that it may not require tedious encoding of pedagogical rules, as the model should learn the teaching actions to take based on the student's performance. Because of the large number of training examples often required for these systems, simulated students are sometimes used to generate training data.

Metrics for Intelligent Tutoring Systems

Evaluating the effectiveness of a tutoring agent is not trivial, as each the agent interacts with many different learners, and each individual interaction can be complex. We list a few evaluation metrics commonly used in existing tutoring systems.

- *Learning Gains*: This metric refers to how much a student learns during the course of an interaction. This is typically measured by the difference in pretest and post-test scores to see how much improvement is made (Graesser et al. 2005). While learning gains are arguably the most important desired effect of an ITS, it is difficult to control for this amongst many participants. Ways to measure learning gains are often domain specific.
- *Time to Complete Problems*: ITS systems often attempt to minimize the average time required for a student to complete problems correctly (Beck, Woolf, and Beal 2000). While this is straightforward to measure, a low time to complete problems does not always indicate a successful tutoring interaction. Students who can complete problems quickly may be finding the problems unengaging if they have already mastered the content.
- *Time Spent Off Task*: Some tutoring systems capture the amount of time the learner spends off-task from the main learning task. This deals with user engagement, and assesses how effective the system is in maintaining engagement. This is difficult to measure, as a user's idle behavior may correspond to other things as well, such as boredom or frustration (Sabourin et al. 2011).

Proposed Model

We aim to understand the effects of using a personalized social robot in a tutoring interaction with children. While RL has been explored in several computer-based tutoring systems, little work has been done using RL in robot tutoring interactions. This provides the motivation to study how a social robot will attempt to personalize curriculum sequencing, or a teaching policy dictating the ordering of questions presented to the learner based on difficulty level. This is an important piece of a tutoring scenario that lends itself well to personalization.

We also wish to augment our RL model by further personalizing the question ordering based on the child's engagement level in real time. The tradeoff between between maximing learning gains and sustaining engagement is not trivial; because we are using an embodied robot, we want to make use of physical sensors to detect engagement online during the tutoring interaction. Maximizing both learning gains and engagement level allows us to explore the tradeoff between challenge and motivation in a robot tutoring system for children, as opposed to just having a set of questions with strictly increasing difficulty ratings.

Model Learning Technique

We propose RL as the method for which the system can learn a desired presentation order of the questions. We envision an application scenario in which children are learning how to solve arithmetic problems.

Let N represent a set containing k questions, of varying levels of difficulty. Let d be the number of difficulty levels we represent in our set of k questions. Then we can define $D_1, ..., D_d$ sets where $D_a = \{i \in N | i \text{ has difficulty level } a\}$. Then $\forall i \in N, \exists a$ s.t. $i \in D_a, 1 \leq a \leq d$. We define $Q = \{q_1, ..., q_d\}$ as our set of states. When the system presents question $i \in D_j$ to the learner, the system is in state q_j where j is the difficulty level of question i. We construct transition matrix T containing the state transition probabilities. These are initialized by estimating probabilities relating the d difficulty levels. We represent a policy π as a given order used to present k questions. An optimal policy π is learned through reward function $R = t_{post} - t_{pre}$ where t_{pre} and t_{post} are the learner's scores on a pretest and post-test.

This reward will be used to find a optimal policy during the training phase. In a later phase, when children are interacting with the social robot, intermittent reward signals that are determined from real-time engagement detection will be applied online to the RL algorithm to further optimize the policy to maximize both overall learning gains and sustained engagement over the course of the interaction.

Collecting Training Data

In order to collect training data, we will implement this learning algorithm in a computer-based system and ask participants (children) to complete a tutoring exercise. We will use a pretest and post-test to evaluate the participant's relative learning gains, and use this evaluation as the reward for the policy they are given. Because students often have differing baseline levels of knowledge, we use the pretest score to bin each participant into a given category. This category dictates the initial state that the system will start in. Therefore, policies will be learned by collecting this data, each dependent on the user's baseline knowledge, reflected in the initial state chosen by the system. After training, we will have a transition matrix, which given some initial state s_0, can be used by the system to decide which q_j to start in.

By collecting this data through a computer-based interaction, we have the potential to have a larger number of children use the system. This is crucial as we will need a large number of training examples for our model to learn. We define q_c as the most commonly chosen initial state for all the participants in the training phase.

Figure 1: DragonBot robot

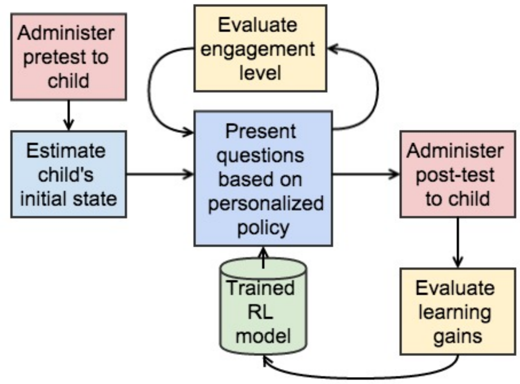

Figure 2: General procedure for proposed study: group receiving personalized question order

Estimating Initial State

Each participant is categorized into a given group based on baseline knowledge level at the start of the interaction. The group selected for the child dictates the initial state that the system will start in when presenting questions. Estimating this initial state is a crucial part of the interaction, as our RL approach attempts to learn optimal policies specific to the various groups. This represents the idea that children with different knowledge levels will require a different curriculum to maximize learning and sustain engagement through the interaction.

Metrics to Evaluate Model

In our training phase, we use the scores of the pretest and post-test to estimate learning gains. These scores serve as the reward signal. In our study involving a social robot with children, we want to use methods of detecting the child's engagement as another metric for evaluation. Because we are using an embodied robot, we can use sensors such as cameras, microphones, and a Kinect to characterize the user's engagement throughout the interaction. These types of measures are important for evaluation of the tutoring interaction. Learning gains will also be measured, but we aim to create a robot tutor that both fosters learning and provides a challenging yet stimulating learning environment for the child. We have prototyped a simple classifier that detects confusion based on head tilt and posture changes found in Kinect data. While the development of the classifier is still in progress, initial tests of the system indicate that it is feasible to detect levels of confusion online while a person interacts with a robot using only a Kinect. We will further develop this system to more precisely capture varying levels of engagement in the tutoring interaction, which will involve detecting confusion, boredom, and distraction.

Proposed Study Utilizing Personalized Model

We propose a study involving a robot tutor using our trained RL system to decide the ordering of basic questions presented to a child in a one-on-one tutoring interaction. We plan to use a DragonBot (see Figure 1), a dragon-like squash-and-stretch robot with five degrees of freedom that is well-suited to interact with children (Setapen 2012).

We will randomize children into two groups. One group will engage in a tutoring interaction in which we estimate the child's initial state and the robot presents the question ordering based on our learned transition matrix. This ordering is more personalized to the child's specific knowledge level. Additionally, children in this group will receive a question order that dynamically changes based on their engagement level. When presenting a question to the student, the model will look at the learned transition matrix, but will also take into account the engagement level assessed in real-time. If the child is detected as distracted or confused, the question ordering will deviate from the learned model, and will adapt to simultaneously maximize learning gains and engagement level throughout the interaction.

The second group represents a control group in which each child is given questions in the order obtained from starting in q_c, the initial state most commonly selected during training. This mimicks the idea that in traditional classroom instruction, teachers tailor their teaching methods to the majority of the class, rather than individual students. Initial state q_c is analogous to the baseline knowlege level of the average student in a classroom setting.

The general procedure for the experiment group given the personalized policy learned from the RL model is shown in Figure 2. We will evaluate our system by looking at both learning gains and engagement level throughout the interaction. We hypothesize that both measures will be higher on average for children in the first experimental group compared to children in the control group. The problem of determining question order based on difficulty level can be generalized to other learning domains outside of our tutoring scenario. This experiment will allow us to explore how beneficial it may be to have a social robot demonstrate personalized behavior in a robot-child tutoring interaction. Additonally, it will allow us to gain insight on the tradeoff between maximizing learning gains and sustaining engagement throughout the tutoring interaction.

References

Beck, J.; Woolf, B. P.; and Beal, C. R. 2000. Advisor: A machine learning architecture for intelligent tutor construction. *AAAI/IAAI* 2000:552–557.

Bloom, B. S. 1984. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher* 13(6):4–16.

Chi, M.; VanLehn, K.; and Litman, D. 2010. Do micro-level tutorial decisions matter: Applying reinforcement learning to induce pedagogical tutorial tactics. In *Intelligent Tutoring Systems*, 224–234. Springer.

Conati, C.; Gertner, A. S.; VanLehn, K.; and Druzdzel, M. J. 1997. On-line student modeling for coached problem solving using bayesian networks. *Courses And Lectures-International Centre For Mechanical Sciences* 231–242.

Giannandrea, L., and Sansoni, M. 2013. A literature review on intelligent tutoring systems and on student profiling. *Learning & Teaching with Media & Technology* 287.

Gonzalez, C.; Burguillo, J.; and Llamas, M. 2006. A qualitative comparison of techniques for student modeling in intelligent tutoring systems. *2013 IEEE Frontiers in Education Conference (FIE)* 0:13–18.

Graesser, A. C.; Chipman, P.; Haynes, B. C.; and Olney, A. 2005. Autotutor: An intelligent tutoring system with mixed-initiative dialogue. *Education, IEEE Transactions on* 48(4):612–618.

Graesser, A. C.; Conley, M. W.; and Olney, A. 2012. Intelligent tutoring systems. *APA handbook of educational psychology. Washington, DC: American Psychological Association*.

Leyzberg, D.; Spaulding, S.; Toneva, M.; and Scassellati, B. 2010. The physical presence of a robot tutor increases cognitive learning gains. In *34th Annual Conference of the Cognitive Science Society (Cog. Sci.12)*.

Malpani, A.; Ravindran, B.; and Murthy, H. 2011. Personalized intelligent tutoring system using reinforcement learning. In *FLAIRS Conference*.

Martin, J., and VanLehn, K. 1995. Student assessment using bayesian nets. *International Journal of Human-Computer Studies* 42(6):575–591.

Sabourin, J.; Rowe, J. P.; Mott, B. W.; and Lester, J. C. 2011. When off-task is on-task: The affective role of off-task behavior in narrative-centered learning environments. In *Artificial Intelligence in Education*, 534–536. Springer.

Setapen, A. 2012. Creating Robotic Characters for Long-Term Interaction. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA.