Presenting Adaptive Robotic Tutoring as a Contextual Bandit Problem

Anthony Fumagalli
Advisers: Brian Scassellati and Aditi Ramachandran
Yale University

CPSC 490, Spring 2015

Abstract

Studies consistently show that tutoring services can dramatically improve learning gains in children. Furthermore, the physical presence of social robots as tutors has also been shown to have a positive effect on children’s cognitive abilities. Social robot tutors would be made significantly more effective if they were able to adapt to the specific needs of individual students much in the way that human tutors do. Adaptive systems have already been studied extensively and many are currently used in classrooms to customize learning according to a child's particular goals, pace, and responses to specific exercises. We would like to bring adaptive learning to robotics by programming a social robot tutor to adapt in real time to multiple layers of feedback including student history both within a particular tutoring interaction and over all time. We suggest that presenting this concept as a contextual bandit problem is an appropriate way to increase the efficacy of robot-child interactions, as well as to promote greater engagement and cognitive learning gains in participating children. Since the contextual bandit problem frames our adaptation as a response to a set of data, or feature vector, it is naturally suited to our goals of adapting within a tutoring interaction in real time. Finally, we suggest an initial study on children's help-seeking behaviors that will aid us in evaluating and refining our system. We believe that continued study of robotic tutoring and of our model could prove to be instrumental in solving the issue of inadequate access to tutoring resources.

Introduction

One-on-one tutoring has been found to be a particularly effective way to improve a student’s performance. In a well-known 1984 study published in the journal “Educational Researcher,” psychologist Benjamin Bloom reported that, on average, students who receive one-on-one tutoring perform two standard deviations better than those who do not. This phenomenon has become known as Bloom’s 2 Sigma Problem, referring to the high cost of tutoring and lack of access many students have to tutoring resources [2].
Bloom’s study had many large implications on educational research. For one, it suggests that the labeling of students as low achievers is less relevant since significantly improved learning gains can be observed by introducing high quality tutorial instruction. Additionally, and most importantly for this project, many researchers have proposed technological methods as a means of addressing the lack of resources problem. Finally, this evidence on the efficacy of tutoring suggests an important role of sociality in learning. Robotic tutors are thus well equipped to address these observations. This report explains the methodology used to develop such a robotic tutor.

Background

Adaptive learning systems have become popular among educators in recent years. Such systems allow for learning to be customized to meet the needs of individual students. Currently, almost all adaptive systems are based around on-screen interactions, usually through a computer or tablet.

In 2012, Lezyberg et al. published a paper reporting that robot tutors that are physically present may induce greater learning gains than videos of the same robots providing the same information [6]. Given the known effects of sociality on tutoring that we learned from Bloom and the subsequent results of this study, we have important motivation for continued research on social robot tutors. Moreover, knowing the benefits of adaptation, it is thus natural that we explore the effects of personalized robot-student tutoring interactions [9]. Using this evidence as motivation, we set out to build an adaptive social robot tutor, which we hope can yield strong cognitive learning gains in children.

To create an adaptive robot tutor, we programmed a Nao robot. The Nao is a humanoid robot built by the French company Aldebaran Robotics. The Nao has been used in numerous child studies, and is an ideal choice for a robot tutor. Its human-like appearance makes it easily relatable for children. Furthermore, it is equipped with cameras that could let us detect affect, LEDs that allow it to express emotion, and motorized joints that bring it to life. Additional features are noted in Figure 1.
The Contextual Bandit Problem

We propose presenting adaptive tutoring as a contextual bandit problem. In the general multi-armed bandit problem, a gambler is presented with several slot machines and must choose which machines to play, how much to play them, and in what order they must be played to maximize the sum of rewards. All machines output a reward at random with a distribution specific to that machine.

More formally, the multi-armed bandit problem is presented as a set of real distributions $B = \{R_1, \ldots, R_K\}$. Each distribution is then associated with the reward for pulling one of $K \in \mathbb{Z}$ levers. With $\mu_1, \ldots, \mu_K$ representing the mean values of these distributions, we iteratively pull one lever at a time, observe the attained reward, and aim to maximize the sum of these rewards. We say the horizon $H$ represents the rounds yet to be played and let $T$ be the rounds already completed. We then define the regret to be

$$\rho = T\mu^* - \sum_{t=1}^{T} \hat{r}_t$$

where $\mu^* = \max_k \{\mu_k\}$, the maximum reward, and $\hat{r}_t$ is the reward received at time $t$. We thus want to find, for every $T$, the reward that minimizes regret [1].
To design our adaptive system, we consider a variation of the multi-armed bandit problem known as the contextual bandit. In this case, the gambler is informed not only by the outcomes of his choices but also by a $d$-dimensional feature, or context, vector. After several iterations, the gambler uses information on past rewards to refine the feature vector and thus better predict the next best decision.

One of the best known applications of the contextual bandit problem was proposed by Li et al. as a way to personalize news recommendations. Their 2010 paper compared algorithms for solving the contextual bandit problem, hoping to find a way to better adapt web content on Yahoo! In particular, their paper compared versions of the $\epsilon$-greedy and LinUCB algorithms. $\epsilon$-greedy simply works by picking the arm with highest predicted payoff with probability $1 - \epsilon$ and picking a random arm with probability $\epsilon$. LinUCB, on the other hand, is more sophisticated and involved, and is shown in Figure 2 [7].

![Algorithm 1 LinUCB with disjoint linear models.](image)

**Figure 2:** The LinUCB algorithm  
Source: “A Contextual-Bandit Approach to Personalized News Recommendation”

While the version of the Nao tutor programmed for this project uses an algorithm that deviates significantly from the contextual bandit problem, it is still useful to consider our goals from this perspective. In particular, a contextual bandit solver could be very helpful when our tutor is tasked with adapting to multiple layers of feedback such as long-term and short-term student history, assumptions based on the student answer, and perceived level of engagement.
Implementation and Discussion

My part of this project was the implementation of an adaptive robot controller for the Nao tutor. The final product deviates from algorithms like LinUCB, but is still very much inspired by the contextual bandit. I began by writing a Python script that gives commands to the robot. The script also collects data on the tutoring interaction and adapts commands based on user feedback and behavior.

My Python script uses a feature vector that consists of: percentage of questions answered correctly, total number of questions asked, number of consecutive incorrect answers, and whether the user has asked for help. The script tracks this data for all participants across several question types (in the demo, the types are simple addition and multiplication). Decisions are then made based on this information. For example, if a student responds to a question incorrectly but has a high percentage in that question category, the Nao encourages the student to try again. Conversely, if the student shows continued struggles with a question type, the Nao offers to provide a hint. Moreover, the script is able to track this data across multiple tutoring sessions so that we can assess student progress over time. This feature is key to developing an effective robot tutor.

There are two versions of the script included in this project. One is capable of managing the entire interaction, from generating questions to evaluating answers. The other requires a TCP server to present questions and send answer data to the robot controller. The TCP connection provides important functionality by allowing us to present questions on an external source such as a tablet or computer screen. This helps us to increase the interactivity of a tutoring session.

The robot controller included with this project is meant to serve as a demo of possible capabilities, and therefore is much too static in its approach to power a truly effective tutor. However, this script provides an important framework for the storage and access of data, as well as establishing the TCP connection for more engaged interaction. The script is also easily adaptable to a large number of question types and can be used broadly with any appropriate TCP server.

Limitations

The current version of the Nao robot is still rudimentary in its approach to adaptation and would need to be improved in order to function as an effective tutor. In particular, the Nao is not currently successful at solving the contextual bandit problem. This means that Nao does not truly learn from its decisions yet. For example, the current version asks the user if they would like a hint if an incorrect answer were given for a question in a category the user has historically answered less than 70% correct. Regardless of the outcome of making this decision, the Nao will always ask this question when presented this scenario. If Nao were implementing a contextual bandit solver, the 70% cutoff would adjust according to
whether the outcome of presenting a hint was positive or negative. If the hint led to better overall student performance, the Nao may maintain the cutoff level and continue to give hints in this situation. If the hint decreased student performance, then Nao might start pulling from a different hints database or suggest taking a break as an alternative action.

We would also like Nao to use more layers of feedback than just student performance history. One possibility is to utilize the Nao’s built-in camera to detect the affect of the student. This information could then be used to make even more informed decisions within the tutoring interaction. As an example, if a student appeared disengaged then Nao might suggest playing a short non-academic game as a break. Or if the student appeared bored but was generally answering questions correctly, then Nao might switch to a different topic or increase the difficulty level of the questions. Human tutors are exceptionally skilled at detecting affect, so we should expect robot tutors to do the same although accomplishing this has proved extremely challenging for now.

Finally, tutoring interactions would likely benefit from more animation from the Nao. Currently, Nao stands up when introducing itself and then sits down when tutoring begins. By sitting down, Nao assumes an unintimidating position that we believe children will respond positively to. However, throughout the interaction, Nao currently remains motionless. To make the tutor seem more relatable, it might therefore be beneficial to program Nao perform more actions, such as nodding its head when a correct answer is given or moving its arms while speaking.

**Proposed Follow-Up Experiment**

In order to better understand what kinds of feedback are important to track in a robot-child tutoring interaction, it is important that we study the kinds of behaviors children show when interacting with our adaptive robot. As an initial investigation, we propose a study on children’s help-seeking behaviors when engaged with our robot. We are interested in help-seeking behaviors because an important skill of human tutors is identifying when a student should be given extra help and when it is best to let a student continue to work on their own.

Interactive learning environments (ILEs) often have an on-demand help feature that students can call when needed. However, this is not always used appropriately as some students may be averse to asking for help or may game the system by always doing so. We ask whether these same behaviors will be present when students interact with a social robot tutor. In particular, we are interested in the types of help-seeking behaviors that students will exhibit in one-on-one tutoring interactions and whether the robot’s timing and availability of support might influence learning gains and engagement. Understanding this could help us build a robotic tutor that can influence the most beneficial help-seeking behaviors in children.
Our proposed experiment will use the Nao robot as a tutor for a connected tablet math game. The tablet creates a TCP server that sends information about the questions and student answers to the Nao, which then processes, stores, and adapts to that data. We will focus on one-on-one interactions with 5th grade children learning basic operations (addition, subtraction, multiplication, and division) on fractions. We focus on fractions specifically because much information can be gleamed from a student answer to these kinds of questions. For example, if a student answers the expression $1/3 + 2/5$ with $3/8$, this shows a different kind of misunderstanding than a student who answers $11/16$. The first suggests a student who is ignorant of the need for common denominators. The second shows an error in achieving common denominators. We can thus potentially use the student answer itself as contextual information for our adaptive system.

Our proposed experiment will be conducted in four sessions, each followed by a post-interview about perceptions of the robot, perceived engagement, and other topics. The first session begins with a pretest and the last ends with a posttest so that we can evaluate knowledge level and learning gains. Furthermore, the first session will be shorter than the other three and will be used to evaluate the student’s tendencies towards help-seeking behaviors. The overall structure is depicted in Figure 3.

![Figure 3: Proposed experiment structure](source: Created by Aditi Ramachandran)

Participating students will be put into one of two groups for the final three sessions. The first group will be working with an adaptive robot that will use contextual information to rate-limit hints and provide assistance only when appropriate. More specifically, the adaptive robot will not provide a hint more than once in a set amount of time and will automatically provide help if the student’s performance is below a defined threshold and they take a long time to answer a question. The
second group will work with a control robot that will provide hints immediately upon request.

We expect that our findings will be consistent with previous research on robotic tutor systems and that those in the adaptive group will show the greatest learning gains at the end of the four sessions. Yale PhD student Aditi Ramachandran will conduct this experiment using the adaptive Nao tutor in a future semester.

Conclusion

The proven efficacy of tutoring, as well as the benefits observed when physical robot tutors are present, provide important motivation for studying social robot tutoring and adaptive learning. Presenting robotic tutoring as a contextual bandit problem could be an effective way of building such an adaptive learning system. This has already been done to personalize news recommendations and we believe the same can be done in education. Additionally, we are interested in using such a system to explore children’s help-seeking behaviors. This could allow us to refine adaptive learning programs even further. The source code included in this project only begins to scratch the surface of what is possible with a contextual bandit based adaptive system. Further refinements can help create a more effective robot tutor.

Acknowledgements

I’d like to thank:
Professor Brian Scassellati and Aditi Ramachandran for the opportunity to work with them on this project in Yale’s Social Robotics Lab,
Claudia Merson for introducing me to the lab and for continuing to inspire my passion for education, and
Ijeamaka Anyene for her constant support and editing skills that helped to make this report possible.
References


