BKT-POMDP: Fast Action Selection for User Skill Modelling over Tasks with Multiple Skills

Nicole Salomons*, **Emir Akdere** and **Brian Scassellati**
Yale University
{nicole.salomons, emir.akdere, brian.scassellati}@yale.edu

Abstract

Creating an accurate model of a user's skills is necessary for intelligent tutoring systems. Without an accurate model, sample problems or tasks must be selected haphazardly by the tutor. Once an accurate model has been trained, the tutor can selectively focus on training essential or deficient skills. Prior work offers mechanisms for optimizing the training of a single skill or for multiple skills when individual tasks involve testing only a single skill at a time, but not for multiple skills when individual tasks can contain evidence for multiple skills. In this paper, we present a system that estimates user skill models for multiple skills by selecting tasks which maximize the information gain across the entire skill model. We compare our system's policy against several baselines and an optimal policy in both simulated and real tasks. Our system outperforms baselines and performs almost on par with the optimal policy.

1 Introduction

In the past decade, there has been a significant increase in the deployment of Intelligent Tutoring Systems (ITS) [Desmarais and Baker, 2012]. These systems create models of a student's knowledge states, that is, their expertise across a set of skills. When an ITS system has an accurate model of the student's skills, it can selectively choose problems or tasks (we will use task and problem interchangeably throughout the paper) to focus teaching where needed.

There has been prior work on selecting which task to present a user to maximize their learning [Schodde *et al.*, 2017; David *et al.*, 2016]. However, these systems consider that each task assigned to a student maps one-to-one with a modelled skill, an assumption that frequently does not hold. Consider a simple math task: $(3 * 9)/(1 + 3)$. To successfully complete it, the user would need knowledge of addition, multiplication, and division. However, prior research usually tests one skill at a time when accounting for several skills. Testing skills individually takes longer than if multiple skills are tested concurrently. Furthermore, there are domains where it is not possible to separate skills and test them individually. For example, swimming might consist of a skill for arm movement, leg movement and taking breaths, but these are challenging to test completely independently. Prior work on tasks containing multiple skills [Xu and Mostow, 2011; González-Brenes *et al.*, 2014] did not include action selections policies to select what the best task is to present to the user, and usually present tasks to the user at random.

Selecting the correct action when multiple skills are present is a hard problem for several reasons. One (in)correct observation alone is not sufficient to determine mastery as there is the chance that the participant has slipped or guessed during the task. Action selection when the true state (in this case which skills are mastered) is unknown is usually solved using a Partially Observable Markov Decision Process (POMDP) [Astrom, 1965]. However, the number of states is exponential in the number of skills and a POMDP is exponential in the number of states, making it computationally intractable.

In this paper we present a modified version of POMDPs, allowing action selection on tasks with multiple skills to be done online. Our system selects the best action, which is the one that minimizes the uncertainty of the user's knowledge state, that is, the task that will give the system the most information gain of the user's skill capabilities. The model is also extended to not only test skills but to allow for users to learn throughout the interaction, enabling teaching to occur.

During each time-step, a task is selected for the user. After observing the user complete the task, the system updates each skill's probability of mastery. Using the updated skill levels, the system proceeds to select a new task. In our system, the likelihood of skills being mastered or not is updated using Bayesian Knowledge Tracing (BKT) [Corbett and Anderson, 1994]. Actions are selected using a modified POMDP. We call our system Bayesian Knowledge Tracing - Partially Observable Markov Decision Process (BKT-POMDP).

To validate BKT-POMDP, we compare it against three other action selection policies: a Random policy, a Handcrafted policy, and an Optimal policy. We perform three sets of experiments. The first was done in simulation, where we randomly generate tasks, skills, and users. The second was a human-subjects experiment where participants complete an electronic circuit building task. In the third experiment, BKT-POMDP accounts for learning throughout a simulated inter-

*Contact Author

action. In all three experiments, BKT-POMDP learned the user's state faster and more accurately than the Random policy and the Hand-crafted policy. It performed comparably to the Optimal policy in terms of accuracy and speed. Therefore, we show that BKT-POMDP is a suitable action selection mechanism to create a model of a user's capabilities across multiple skills.

2 Background

In this section, we provide background on user skill modelling and action selection in tutoring systems.

2.1 User Skill Modelling

Prior work in assessing knowledge levels of users has mainly been conducted in Intelligent Tutoring Systems [Anderson *et al.*, 1985]. The predominant method used for determining whether a skill has been mastered or not is Bayesian Knowledge Tracing [Corbett and Anderson, 1994] (BKT). In BKT, the student is given a succession of problems for a single skill. The system observes the student's answers and updates their probability of mastery for each skill. An alternative to BKT is Learning Factors Analysis (LFA) [Cen *et al.*, 2006], which learns a cognitive model of how students solve problems. It learns the difficulty and the learning rate of each skill using student data. However, LFA does not track individual student progress by estimating their knowledge state in annotating correct and incorrect answers. In response to the limitations of LFA, Performance Factors Analysis (PFA) [Pavlik Jr *et al.*, 2009] was proposed to both estimate the student's knowledge and to create a more complex model of skills.

Several models have extended BKT, LFA, and PFA to allow multiple interdependent skills in each problem [Xu and Mostow, 2011; González-Brenes *et al.*, 2014; Pardos *et al.*, 2008]. There are also models which assume that all skills need to be applied correctly to achieve the correct answer in a problem [Cen *et al.*, 2008; Gong *et al.*, 2010]. Similar to our system, they can model problems that contain more than one skill present. However, they do not select which problem or task to give to a user to maximize the system's certainty of the user's model. They only create a model of how skills are interlinked in the problems.

In this paper we are using BKT to model the probability of observations rather than LFA or PFA, as we do not have a corpus of previous data on the problems we will be testing. Furthermore, we use BKT instead of a conjunctive model as we assume independence of skills in each problem. However, BKT can easily be replaced by a different algorithm when a more intricate model is required.

2.2 Action Selection During Tutoring

ITS systems have focused on what problems to give students or how to assist students to maximize learning. There has been research on which skill to teach a student so that their knowledge across all skills is maximized [Schodde *et al.*, 2017], which sequence of problems to present to students depending on skill difficulty [David *et al.*, 2016], and what type of help to give the students by observing the student's motivation and knowledge state [Ramachandran *et al.*, 2019]. Other studies created a system that decided what type of help to give a student by creating personalized models of each student [Clement *et al.*, 2013; Lan and Baraniuk, 2016]. Creating individualized models for each student leads to higher learning gains [Yudelson *et al.*, 2013].

Prior research has focused either on action selection or on allowing a problem to have multiple skills. In our work, we construct a system that can both handle problems with multiple skills and select actions that maximize the certainty our model has of the user's skill state.

3 BKT-POMDP Task Selection

We describe here a system which selects optimal actions when creating a model of user capabilities across multiple skills. We make several assumptions in our model: 1) The user skill state is constant, and they will not learn during the interaction. Although we later present an extension to the model that allows for learning. 2) Each skill's importance is equal; however, this can be changed easily if an application requires. 3) One task is given to the user at each time-step, and the task can contain one or multiple skills. 4) Lastly, skills are independent of each other; that is, one skill's mastery is independent of another's skill mastery.

This system draws inspiration from Partially Observable Markov Decision Processes (POMDPs) [Kaelbling *et al.*, 1998] and belief state MDPs [McAllester and Singh, 2013] in that the system does not have full knowledge of the state S, and uses observations o to create an estimate b of what the state is. To learn the model, it selects actions a that maximize the expected information gain reward r of the new belief b' compared to the prior belief b.

This section presents our system called Bayesian Knowledge Training - Partially Observable Markov Decision Process (BKT-POMDP). Similar to the POMDP, our model is composed of the following:

- **S** - The true skill state of the user. A state is represented as a binary vector, with each element i in the vector representing whether skill i is mastered (1) or not (0).
- **b** - The skill belief vector. This is the current estimate the system has of S. Each element in the vector represents the estimated probability of skill i being mastered.
- **A** - The set of actions that can be taken. Each action is a task that can be presented to the user that contains multiple skills. Each action is a vector, with 1s for the skills being tested, and 0s for those that are not.
- **O** : $P(o|b, a)$ - Observation probabilities. The probability of an observation given the current belief distribution and the action chosen. The observation probabilities are based on Bayesian Knowledge Tracing.
- **T** : $b' = P(b|o)$ - The transition function updates the belief, given the current belief and the observation. In BKT-POMDP, the transition will be updated using the Bayesian Knowledge Tracing formulation.
- **R** - The reward function. In traditional POMDPs, the reward is a function of either the current state or of the current state plus action. However, our reward is a function of the current belief and the previous belief. Our

reward function maximizes the information gain of the user's state at each time-step.

- Ω - The set of possible observations. An observation will be a vector of 0s, 1s, and 2s, where 0 represents the wrong answer for that particular skill, 1 represents the right answer, and 2 represents a skill not being tested during that time-step.

3.1 Skill Belief Vector

Even though the number of possible states is exponential in the number of skills tested, it can be represented as a belief vector with a belief value for each skill. For example, if the belief for skill i is currently 0.95, that means that it is very likely that the user has mastered that skill. If the value is 0.3, it is more likely that they do not know that skill, but the system is not certain of this. The skill belief vector is initialized to 0.5 for all skills, representing complete uncertainty at the start of the interaction. In our formulation of the POMDP, all computations can be done on the belief vector rather than over all the possible states. This makes BKT-POMDP much faster to solve than traditional POMDP, as POMDP computes over all the possible skill states ($2^{|S|}$ different possible states), and our calculations are done on just the belief vector.

3.2 BKT-POMDP Action Selection

The optimal value function of the POMDP will select the action (the task), which has the highest expected reduction in uncertainty of the user's skill state (Equation 1). It iterates over all possible actions (in this case, the possible combinations of skills to test) and selects the one which it expects to have the highest Q value. The Q value (Equation 2) is the expected reward when taking a specific action. Upon selecting an action, it will consider all the possible observations and calculate the resulting belief from that observation $b' = T(b, o)$. It will calculate the likelihood of the observation multiplied with the observation's reward and add the discounted expected reward of the next optimal action. The reward is calculated by the expected increase of certainty of the user's skill state after taking an action.

$$V^*(b) = \max_{a \in A}(Q^*(b, a)) \tag{1}$$

$$Q^*(b, a) = \sum_{o \in O} [P(o|b, a) \cdot R(b, b')] \tag{2}$$

3.3 Belief Update

Each of the tested skills in the belief vector is updated independently using the BKT framework [Yudelson *et al.*, 2013]. In BKT, the probability of knowing a skill is dependant on whether the observation was incorrect ($o_i = 0$) or correct ($o_i = 1$), and also on the probability of guessing ($P(G_i)$) or slipping ($P(Sl_i)$) for that skill. When the skill is not being tested ($o_i = 2$), that particular skill's belief value remains the same. Equation 3 shows the belief update.

$$b'_i = \begin{cases} \dfrac{b_i \cdot p(Sl_i)}{b_i \cdot p(Sl_i) + (1 - b_i) \cdot (1 - p(G_i))}, & \text{if } o_i = 0 \\ \dfrac{b_i \cdot (1 - p(Sl_i))}{b_i \cdot (1 - p(Sl_i)) + (1 - b_i) \cdot p(G_i)}, & \text{if } o_i = 1 \\ b_i, & \text{if } o_i = 2 \end{cases} \tag{3}$$

3.4 Reward Function

In the traditional POMDP model, the reward usually is related to the specific state. Conversely, in the BKT-POMDP, the reward relates to how much the certainty of the skill state has increased compared to the previous time step. That is, the more certain the system is of the user's skill compared to the previous time step, the higher the reward will be. We use Kullback-Leibler divergence (KLD) [Kullback and Leibler, 1951] to calculate the information gain of the new belief compared to the previous belief (Equation 4). KLD is first calculated for both the old belief and the new belief compared to the belief vector of complete uncertainty (U), where $U = [0.5, 0.5, ...0.5]$. The reward is how much information is gained with the new belief compared to the old belief (Equation 4).

$$D_{KL}(b \parallel U) = \textstyle\sum_i b_i \ln\left(\frac{b_i}{0.5}\right) + (1 - b_i) \ln\left(\frac{(1 - b_i)}{0.5}\right)$$
$$R(b, b') = D_{KL}(b' \parallel U) - D_{KL}(b \parallel U) \tag{4}$$

3.5 Observation Function

The probability of a specific observation is the product of all of the individual skill observations that were tested during that round ($a_i = 1$) given the current belief state (Equation 5). The probability of observing the incorrect answer ($o_i = 0$) is the probability that the user possessed the skill but slipped plus the probability that they did not possess the skill and did not guess correctly. The probability of observing the correct answer ($o_i = 1$), is the likelihood that they possessed the skill and did not slip plus the probability they did not possess the skill but guessed correctly. When the skill was not tested ($a_i = 0$), it did not influence the observation's probability. The observation's update function can be seen in Equation 5.

$$P(o|b, a) = \prod_i (p(o_i|b_i, a_i)) \tag{5}$$

$$p(o_i|b_i, a_i) = \begin{cases} 1, & \text{if } a_i = 0 \\ b_i \cdot p(Sl_i) + (1 - b_i) \cdot (1 - p(G_i)), & \text{elif } o_i = 0 \\ b_i \cdot (1 - p(Sl_i)) + (1 - b_i) \cdot p(G_i), & \text{elif } o_i = 1 \end{cases}$$

4 Metrics

In this section, we present the baselines and the measures used for evaluating BKT-POMDP.

4.1 Baselines

We compare our policy (BKT-POMDP) against three different policies: two baselines (a Random policy and a Hand-crafted policy) and the Optimal policy. We assume there is no repetition of tasks, although the policies could easily be modified to allow it.

- **Random** - A task is selected randomly and presented to the user. The Random policy is the most commonly used action selection mechanism in tutoring systems.
- **Hand-Crafted** - It selects the task with the skills least recently tested. It does so by assigning each skill a counter that is initialized at 0. During each time-step, all non-tested skills' counters are increased by one. If the skill is tested, the counter is reset to zero. This policy will use the unweighted sum of these counters to choose its next action.
- **BKT-POMDP** - Our policy creates a model of the user's skills and chooses tasks that it expects will result in the highest information gain of the user's skill state. This is the policy presented in Section 3.
- **Optimal** - This policy selects the optimal action at each time-step. In Experiments 1 and 2, where the goal is skill estimation, it will choose the action that brings the estimate of the user's model b as close to the real model of the user S. In Experiment 3, where the goal is to maximize learning, it will choose the action with the highest expected increase of skills mastered. This policy can select optimal actions as we assume it has full access to S from he start. This assumption does not hold in real scenarios and therefore this policy serves only to illustrate what the optimal policy would be.

4.2 Measures

We used the following measures to validate BKT-POMDP. They were calculated each round after the user completed the selected task, and the model's belief was updated.

Distance to True State - How close the current belief b is from the correct skill state S for each user. It is calculated by the difference between b and S. This metric is used in the first two experiments where the goal is correctly estimating the user's true state.

$$Dist(b, S) = \sum_i (|b_i - S_i|) \tag{6}$$

User Mastery - The number of skills that are mastered. This metric is used in the third experiment, where the goal is teaching all the skills to the user.

$$Mast(b) = \sum_i (S_i) \tag{7}$$

5 Experiment 1 - Skill Estimation in Simulation

We ran a total of 100 rounds of simulations, where different simulated skills, tasks and users were generated. In each round a different user was generated, and they completed 40 different tasks for each of the four policies.

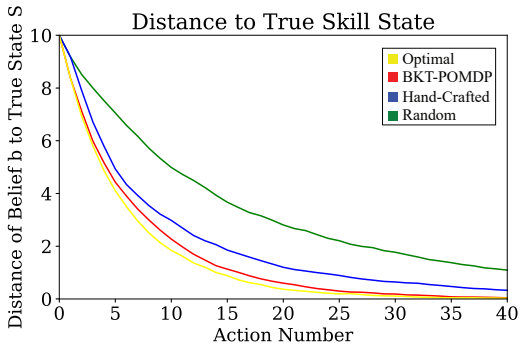

Figure 1: Experiment 1 - The average distance of belief b to the true state S for each of the four policies. Overall the Optimal and BKT-POMDP policies chose the best actions learning the user skill states the quickest. The third best policy was the Hand-crafted policy, and the Random policy performed the worst.

Skills - 20 different skills were created each round. Each skill had associated with it a probability of guessing it correctly and also a probability of slipping while doing it. The probability of guessing and slipping was randomly chosen from a uniform distribution between 0 and 0.3.

Tasks - 200 different tasks were created. Each one had randomly assigned to it between 1 and 5 skills. During each time-step, a task was selected until a total of 40 different tasks were chosen for that round.

User - During each round, a simulated user was generated. For each skill, they were randomly assigned as mastered or not with equal probability. Each user was associated with an observation for each task they would complete. The observation was created using the probability distribution of guessing or slipping depending on whether they were assigned as having mastered that skill.

5.1 Results

We measured the accuracy of the belief state compared to true state using Equation 6. All four action selection mechanisms learned the user's skills accurately over time. However, the Random policy took significantly longer to approach the true user state. The Hand-crafted policy performed better than the Random policy. BKT-POMDP performed almost as well as the Optimal policy. These results can be seen in Figure 1.

During three different points (after 10 tasks, after 20 tasks, and after 30 tasks) we compare whether the accuracy of the policies were statistically significant from each other using an ANOVA with Bonferroni Corrections. In all three cases, the Optimal and the BKT-POMDP policies performed statistically significantly better than the Hand-crafted and the Random policies, and the Hand-crafted solution performed statistically significantly better than the Random policy. The Optimal and the BKT-POMDP policies did not significantly differ from each other.

6 Experiment 2 - Skill Estimation with Participants

We compare BKT-POMDP on a real task with participants completing electronic circuit tasks [Elenco, 2021], using

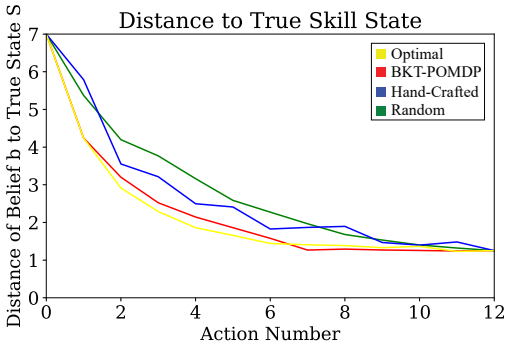

Figure 2: Experiment 2 - The average distance of belief b to the correct state S for each of the 12 task actions. The Optimal policy performed the best, closely followed by the BKT-POMDP policy. The Hand-crafted policy was the third best policy and the Random policy was last.

enlarged electronic pieces including wires, resistors, and switches. The pieces can be snapped together on a board to form circuits. We chose circuits because they require the user to be proficient in a variety of skills, many of the skills are order independent, and there are several possible assemblies.

Skills - There were six different pieces being tested: a switch, a resistor, an LED, a music circuit, a speaker, and a photo-resistor. There were three different types of skills necessary for accurately completing the tasks: placing the correct piece on the board, placing the piece in the correct location, and placing the piece in the correct orientation. Placement of pieces was dependent on choosing the correct piece. The orientation of pieces was dependent on the participant having chosen the correct piece and placing it in the correct location. Therefore if the participant did not choose the correct piece, then we also defined that they were incorrect in the placement of the piece and the orientation of the piece (this slightly breaks our independence assumption, but does not change the computational cost of the algorithm). Only the LED and the music circuit were directional. Therefore there were a total of 14 skills (six pieces chosen + six pieces placed + two pieces orientation) being tested. We consider a participant to have mastery of a skill if they apply it correctly at least 70% of the time (most skills were tested on average five times, so this allowed at least one slip or guess). The guess and slip probabilities for each particular skill were determined by the number of times participants did not have mastery and guessed correctly and when they did have mastery and slipped in our experiment. The average probability of guessing was 0.28, and the average probability of slipping was 0.10.

Tasks - There were 12 different tasks for the user to complete. Each task required a combination of different skills. A board was given to the participant with wires and a battery piece (without batteries inside) that were already placed. The participant was then asked to complete a task. For example, there was a task where the user was asked to create a circuit with a light that could be turned off and on. Therefore they needed to choose the correct pieces: an LED, a resistor, and a switch; place each in the correct location; and place the LED with the correct orientation. In addition to the six different pieces that were being tested, we gave the user four additional distractor pieces (making guessing correctly less likely).

Users - 23 participants completed the 12 circuit tasks, of which 14 were male and 9 were female. The study was approved by the university's Institutional Review Board and participants signed a consent form agreeing to participate. They were not provided with any information on how electronic circuits worked, other than the piece's name and the ports on the pieces. We also assumed that no learning happened throughout the experiment, as no help or feedback was provided. The participants' expertise on circuits was varied, with some participants having mastery of none of the skills, and some having full mastery. All 23 participants' data was used for the four policies by simulating which task the system would have chosen during each time-step for each participant.

6.1 Results

We annotated for every participant whether they had mastery over each skill (they were considered to have mastery if they got the skill right over 70% of the time). For every participant, observations were created by annotating whether they demonstrated the skills successfully in each task. On average, participants were able to choose the right pieces 77.39% ($SD = 14.59\%$) of the time. Participants placed the piece in one of the correct locations 38.75% ($SD = 33.91\%$) of the time. And participants placed the directional pieces in the correct orientation 35.36% ($SD = 32.43\%$) of the time.

During the last few rounds all four policies had high certainty on the user's skills. Therefore we compare rounds 3, 5 and 7 for statistical significance using ANOVAs and Bonferroni Corrections, measuring the distance of the belief compared to the true state (Equation 6). After taking three actions, the Optimal policy performed significantly better than the Hand-crafted and Random policies. BKT-POMDP performed significantly better than the Random policy. The other comparisons were not significant. After five rounds, the Optimal policy performed significantly better than the Random policy. The other comparisons were not significant. There were no significant differences after seven rounds.

7 Experiment 3 - Learning in Simulation

There are many situations, especially in ITS, where we do not only want to create a model of skills, but also select the tasks which will teach the most. We extend BKT-POMDP by changing the reward function and the belief update function to account for learning. These modification allow BKT-POMDP to select the task with the skills that it estimates will bring all the skills' mastery's closest to 1.

7.1 Reward Function for Teaching

The reward function is replaced with Equation 8. It rewards increases in the belief of the skills. Therefore, it rewards the user having higher mastery over the skills. At the start of the interaction the skill belief vector is set to low probability of mastery (0.05) for all the skills.

$$R(b, b') = \sum_i \left[(b'[i] - b[i])\right] if (b'[i] > b[i]) \qquad (8)$$

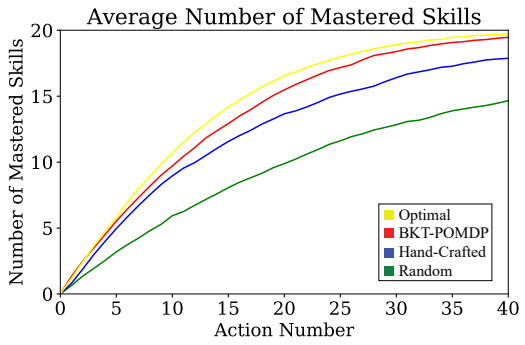

Figure 3: Experiment 3 - The graph shows the number of mastered skills. The BKT-POMDP and the Optimal policies selected tasks that brought the user skill closer to mastery of all skills quicker than the Hand-crafted and the Random policies.

7.2 Belief Update

The belief update still follows Equation 3, however it includes the learning update from BKT [Yudelson *et al.*, 2013]. Each time the participant practices a skill, they have a chance of learning it represented by $P(L_i)$.

$$P(b_i') = P(b_i|o_i) + (1 - P(b_i|o_i)) \cdot P(L_i) \tag{9}$$

Rounds

100 rounds of simulation were run during which 40 different tasks were chosen using the different policies. During each round the following were generated:

Skills - 20 different skills were generated. The probability of guessing and slipping was randomly chosen from a uniform distribution between 0 and 0.3. Additionally the probability of learning ($P(L_i)$) was generated from a uniform distribution between 0.15 and 0.3.

Tasks - 200 tasks were generated, each with between one and five skills.

User - In each round a user was generated. All skills for the user were set as not mastered. After each task, each non-mastered skill in the chosen task was updated by having a $P(L_i)$ chance of the user having learned it.

7.3 Results

We tested which of the policies selected tasks that increased the knowledge of the participants the fastest. We did this by measure the number of skills with mastery using Equation 7. The results are shown in Figure 3, which shows that the Optimal condition selected the best tasks to teach closely followed by BKT-POMDP. The Hand-crafted condition performed third, and Random performed the worst.

During three different points (after 10 tasks, after 20 tasks, and after 30 tasks) we compare whether the conditions were statistically significant from each other using an ANOVA with Bonferroni Corrections. All six pairwise comparisons were statistically significant from each other, except the BKT-POMDP and the Hand-crafted policies after 10 rounds. This indicates that the Optimal policy performed the best, followed by BKT-POMDP, Hand-Crafted and Random.

8 Discussion

We discuss the results of the BKT-POMDP system and how it can be applied in different scenarios.

8.1 BKT-POMDP Task Selection

In the first set of experiments, BKT-POMDP and the Optimal condition converged on the user's true state after 40 tasks. Random and Hand-crafted were approaching convergence and would do so with more assigned tasks. This means that the policies were able to correctly learn the model of the user's skills. However, BKT-POMDP did so much faster than the other baselines, and almost performed as well as the Optimal policy. As the Optimal policy is not possible to use in real scenarios (as it requires a perfect model of the user), BKT-POMDP is a good policy to model a user's skills.

In the circuit experiment, BKT-POMDP and the Optimal policy also outperformed the other baselines. This experiment shows that BKT-POMDP translates well to real world applications. Unfortunately none of the models completely converged in 12 rounds, due to the low number of rounds and the high guess rate for some of the skills.

In the third experiment, we show that BKT-POMDP can easily be modified to allow for different goals. We show that modifying the reward function accounts for user learning. Instead of maximizing student skill, it now maximizes the expected amount of learning the user will have over all skills. In the experiments, BKT-POMDP outperforms the other baselines and performs on par with the Optimal model.

8.2 Applications of BKT-POMDP

BKT-POMDP is a flexible system which can be used for several different applications, and where individual parts can be changed to suit each application. In ITS, the main goal is to teach skills the student does not have mastery over. BKT-POMDP can quickly and accurately create a model of a student's capabilities, so that the ITS can focus on teaching. Modifying the reward function allowed BKT-POMDP to not only create a model of user skills but also account for learning. It selected the tasks that would teach the user the most, and bring the user closer to having full mastery of all skills. Therefore it can be used in intelligent tutoring systems to select which task to teach when there are multiple skills present.

BKT-POMDP could also be used in manufacturing settings. The system could quickly model which skills an employee has, and assign tasks that are within the employee's expertise while also avoiding tasks which they would not be able to do as well. Additionally, when multiple people are present, the system can assign tasks to each person according to their expertise across all tasks, or create teams whose members have an equal balance of skills. In manufacturing settings, some skills are more important than others as they appear in many tasks. The user model over these skills could be prioritized by giving different weights to each skill in the reward function instead of having equal value and, therefore, quickly learning higher weight skills first.

Acknowledgements

This work was supported by awards NSF1813651, NSF1928448, and NSF2033413.

References

[Anderson *et al.*, 1985] John R Anderson, C Franklin Boyle, and Brian J Reiser. Intelligent tutoring systems. *Science*, 228(4698):456–462, 1985.

[Astrom, 1965] Karl J Astrom. Optimal control of markov processes with incomplete state information. *Journal of mathematical analysis and applications*, 10(1):174–205, 1965.

[Cen *et al.*, 2006] Hao Cen, Kenneth Koedinger, and Brian Junker. Learning factors analysis–a general method for cognitive model evaluation and improvement. In *International Conference on Intelligent Tutoring Systems*, pages 164–175. Springer, 2006.

[Cen *et al.*, 2008] Hao Cen, Kenneth Koedinger, and Brian Junker. Comparing two irt models for conjunctive skills. In *International Conference on Intelligent Tutoring Systems*, pages 796–798. Springer, 2008.

[Clement *et al.*, 2013] Benjamin Clement, Didier Roy, Pierre-Yves Oudeyer, and Manuel Lopes. Multi-armed bandits for intelligent tutoring systems. *arXiv preprint arXiv:1310.3174*, 2013.

[Corbett and Anderson, 1994] Albert T Corbett and John R Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.

[David *et al.*, 2016] Yossi Ben David, Avi Segal, and Ya'akov Kobi Gal. Sequencing educational content in classrooms using bayesian knowledge tracing. In *Proceedings of the sixth international conference on Learning Analytics & Knowledge*, pages 354–363. ACM, 2016.

[Desmarais and Baker, 2012] Michel C Desmarais and Ryan S Baker. A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction*, 22(1-2):9–38, 2012.

[Elenco, 2021] Elenco. Snap circuits. https://www.elenco.com/brand/snap-circuits/, 2021. Accessed: 2019-9-10.

[Gong *et al.*, 2010] Yue Gong, Joseph E Beck, and Neil T Heffernan. Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. In *International conference on intelligent tutoring systems*, pages 35–44. Springer, 2010.

[González-Brenes *et al.*, 2014] José González-Brenes, Yun Huang, and Peter Brusilovsky. General features in knowledge tracing to model multiple subskills, temporal item response theory, and expert knowledge. In *The 7th International Conference on Educational Data Mining*, pages 84–91. University of Pittsburgh, 2014.

[Kaelbling *et al.*, 1998] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

[Kullback and Leibler, 1951] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[Lan and Baraniuk, 2016] Andrew S Lan and Richard G Baraniuk. A contextual bandits framework for personalized learning action selection. In *EDM*, pages 424–429, 2016.

[McAllester and Singh, 2013] David A McAllester and Satinder Singh. Approximate planning for factored pomdps using belief state simplification. *arXiv preprint arXiv:1301.6719*, 2013.

[Pardos *et al.*, 2008] Zachary Pardos, Neil Heffernan, Carolina Ruiz, and Joseph Beck. The composition effect: Conjuntive or compensatory? an analysis of multi-skill math questions in its. In *Educational Data Mining 2008*, 2008.

[Pavlik Jr *et al.*, 2009] Philip I Pavlik Jr, Hao Cen, and Kenneth R Koedinger. Performance factors analysis–a new alternative to knowledge tracing. *Online Submission*, 2009.

[Ramachandran *et al.*, 2019] Aditi Ramachandran, Sarah Strohkorb Sebo, and Brian Scassellati. Personalized robot tutoring using the assistive tutor pomdp (at-pomdp). 2019.

[Schodde *et al.*, 2017] Thorten Schodde, Kirsten Bergmann, and Stefan Kopp. Adaptive robot language tutoring based on bayesian knowledge tracing and predictive decision-making. In *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI*, pages 128–136. IEEE, 2017.

[Xu and Mostow, 2011] Yanbo Xu and Jack Mostow. Using logistic regression to trace multiple sub-skills in a dynamic bayes net. In *EDM*, pages 241–246. Citeseer, 2011.

[Yudelson *et al.*, 2013] Michael V Yudelson, Kenneth R Koedinger, and Geoffrey J Gordon. Individualized bayesian knowledge tracing models. In *International conference on artificial intelligence in education*, pages 171–180. Springer, 2013.