COGNITIVE SCIENCE
A Multidisciplinary Journal

Cognitive Science 40 (2016) 1290–1332

ISSN: 0364-0213 print / 1551-6709 online
DOI: 10.1111/cogs.12290

Faster Teaching via POMDP Planning

Anna N. Rafferty,[a] Emma Brunskill,[b] Thomas L. Griffiths,[c] Patrick Shafto[d]

[a]*Department of Computer Science, Carleton College*
[b]*Computer Science Department, Carnegie Mellon University*
[c]*Department of Psychology, University of California*
[d]*Department of Mathematics and Computer Science, Rutgers University – Newark*

Received 12 December 2012; received in revised form 28 April 2015; accepted 14 May 2015

Abstract

Human and automated tutors attempt to choose pedagogical activities that will maximize student learning, informed by their estimates of the student's current knowledge. There has been substantial research on tracking and modeling student learning, but significantly less attention on how to plan teaching actions and how the assumed student model impacts the resulting plans. We frame the problem of optimally selecting teaching actions using a decision-theoretic approach and show how to formulate teaching as a partially observable Markov decision process planning problem. This framework makes it possible to explore how different assumptions about student learning and behavior should affect the selection of teaching actions. We consider how to apply this framework to concept learning problems, and we present approximate methods for finding optimal teaching actions, given the large state and action spaces that arise in teaching. Through simulations and behavioral experiments, we explore the consequences of choosing teacher actions under different assumed student models. In two concept-learning tasks, we show that this technique can accelerate learning relative to baseline performance.

Keywords: Automated teaching; Partially observable Markov decision process; Concept learning

1. Introduction

To instruct a student, a teacher selects a series of activities. Ideally, the teacher should customize her choices based on her knowledge of the domain as well as her beliefs about the student's understanding. When selecting an activity, both the immediate benefit to the learner as well as the activity's potential long-term benefits are relevant. For example,

Correspondence should be addressed to Anna N. Rafferty, Department of Computer Science, Carleton College, Northfield, MN 55057. E-mail: arafferty@carleton.edu

giving a student a quiz might not greatly improve the student's understanding (although this is complicated by the testing effect; for example, Carpenter, Pashler, Wixted, & Vul, 2008), but it could still be beneficial by providing data to inform later pedagogical choices. Similarly, it would be beneficial for the teacher to track the student's likely knowledge over time, based on what activities she has already studied, and provide her with information that supplements this understanding.

Automated tutors should be able to follow these same steps and perhaps even track and respond to information from the learner's behavior more closely than a human teacher. While human teachers may only be guided by coarse information about their students' knowledge (see VanLehn, 2011, for literature review), finer grained information could prove helpful to an automated tutor. The components of the tutor should be modular, such that we can consider different assumptions about learning without changing the entire architecture of the tutor and that we can easily adapt the tutor to different domains. Such a tutor could result in more scalable systems that can adapt to particular characteristics of a domain without relying on heuristics or requiring experts to define pedagogical strategies for individual domains.

Parts of this problem have been approached in previous work. For instance, there has been substantial interest in the cognitive science, education, and intelligent tutoring systems communities in modeling and tracking student learning. A number of results have demonstrated the benefit of taking a Bayesian probabilistic approach (see, e.g., Chang, Beck, Mostow, & Corbett, 2006; Conati & Muldner, 2007; Corbett & Anderson, 1995; Villano, 1992). There has also been some previous work, such as the KLI framework (Koedinger, Corbett, & Perfetti, 2012), that has considered how to make pedagogical choices based on the types of skills being targeted. KLI synthesizes a variety of work finding that the effects of different teaching strategies on learning are heavily modulated by the domain and task. However, there has in general been limited work on how to automatically compute a teaching policy that leverages a probabilistic learner model in order to achieve a long-term teaching objective.

In this paper, we propose using partially observable Markov decision processes (POMDPs) for automatic tutoring, focusing on cases where the automated tutor teaches a student individually. POMDPs allow one to compute a contingent policy for selecting sequential actions in situations where important information may be unobserved (Sondik, 1971). By using this model, we take a decision-theoretic approach that allows us to customize choices based on the learner's observed behaviors as well as the previous pedagogical actions that have occurred and to consider both immediate and long-term gains. The specification of the POMDP also makes it relatively easy to consider different models of learning and different domain models, meeting our goal of modularity. Here we assume that the model of learning is known and demonstrate how to select teaching actions, given such a model. Within the POMDP model, the automated teacher's beliefs about the learner's knowledge is represented as a distribution, preserving uncertainty about the student's actual knowledge. Given a pedagogical objective and a set of models describing the learning process, POMDPs provide a framework for computing a teaching policy that optimizes the objective. The objective function to be optimized can encompass

multiple goals, such as attaining specific knowledge goals quickly and maintaining motivation, but these functions can be challenging to optimize; we address simpler objectives focused on the student's knowledge state and do not consider motivation or other affective issues.

Though POMDPs are related to other decision-theoretic approaches used in previous education research, they are more powerful in two key respects. First, POMDPs can use sophisticated models of learning, rather than assuming learners' understanding can be directly observed or approximated by a large number of features (as in Barnes & Stamper, 2008; Chi, Jordan, VanLehn, & Hall, 2008), and these models are likely to be more interpretable than feature-based approximations. As opposed to Chi et al. (2008), we focus our investigation on how POMDPs can be used to reason about the consequences for teaching of particular cognitive models; in their work, they focused on empirical investigation of using reinforcement learning to optimize a policy for a model based on observed features. Both approaches make valuable contributions but differ somewhat in their aims. The POMDP approach is likely to be helpful when considering many existing cognitive models from psychology, as these typically include information about the learner's mental state. Second, in contrast to approaches that only maximize the immediate benefit of the next action (Conati & Muldner, 2007; Kujala, Richardson, & Lyytinen, 2008; Murray, Vanlehn, & Mostow, 2004; Tang, Young, Myung, Pitt, & Opfer, 2010), POMDPs reason about both immediate learning gain and long-term benefit of a particular activity.[1] Incorporation of information about the effect of particular actions on learning is automatic in the POMDP framework, allowing one to avoid manually specifying heuristics about which teaching actions will be most effective.

Partially observable Markov decision processes offer an appealing framework for selecting teaching actions, but there are often significant obstacles to practical implementation. Specifically, planning teaching requires modeling learning, and richer, more realistic models of learning lead to computational challenges for planning. We instead compute approximate POMDP policies, which make it feasible to use these more complex, realistic models of human learning. As a demonstration of the modular nature of POMDPs, we examine three different models of concept learning and demonstrate how, given the same pedagogical objective, these lead to qualitatively different teaching policies. The use of several learner models allows us to examine whether effective policies can be computed even when the assumed model and true human learning differ. We explore the impact of these varying models in two simple concept-learning tasks, both through simulations and by teaching human learners. We also compare the effectiveness of policies generated using these models to other control policies, such as randomly selecting actions or choosing the action that maximizes the expected information gain, and find that there exist scenarios where POMDP policies outperform these alternatives. While a few recent papers explore the use of POMDPs to compute teaching policies (Brunskill & Russell, 2010; Brunskill, Garg, Tseng, Pal, & Findlater, 2010; Folsom-Kovarik, Sukthankar, Schatz, & Nicholson, 2010; Theocharous, Beckwith, Butko, & Philipose, 2009), to our knowledge, ours is the first paper to demonstrate with human learners that POMDP planning results

in more efficient learning than baseline performance and the first to explore the impact of different models of learning on the computed policies.

The plan of the paper is as follows. We begin by giving an overview of POMDPs and then show how teaching can be formulated as a POMDP. We next explain specifically how to express concept-learning problems as a POMDP and describe three models of concept learning. Given the now fully specified model, we provide an algorithm for computing a policy for choosing pedagogical actions from the POMDP. We then empirically evaluate the effectiveness of POMDP policies for increasing learning efficiency in an alphabetic arithmetic task and in a more complex concept-learning task involving numerical concepts. We conclude by discussing the implications, limitations, and future directions of this work.

2. Partially observable Markov decision processes

Partially observable Markov decision processes planning is used to compute an optimal conditional policy for selecting actions to achieve a goal, in absence of perfect information about the state of the world (Kaelbling, Littman, & Cassandra, 1998; Monahan, 1982). For example, imagine a robot that needs to find a charging station and knows that it is somewhere in a maze but does not know where. By exploring the environment and making observations of the walls and intersections in the maze, the robot can better localize its location to find the charging station more quickly. However, the robot should only explore to the extent that this will help it achieve its goal: If it knows that it is in one of two possible locations in the maze and that in both locations a charging station can be reached by turning left and moving ten meters, it should simply proceed in that direction without further diagnosing its location. POMDP planning provides a way to choose actions that takes into account how uncovering unknown information (e.g., further diagnosing the robot's location) is likely to impact the agent's ability to achieve its goals. This planning model has been used for a wide variety of control tasks, including robotics (e.g., Kurniawati, Hsu, & Lee, 2008; Pineau, Montemerlo, Pollack, Roy, & Thrun, 2003), healthcare (e.g., Hoey, Poupart, Boutilier, & Mihailidis, 2005; Hu, Lovejoy, & Shafer, 1996), and dialogue systems (e.g., Atrash & Pineau, 2006; Roy, Pineau, & Thrun, 2000; Young et al., 2010). POMDP control policies indicate which actions to take, conditioned on the actions taken so far and observations of the environment, such that the expected cost is minimized (or the expected reward is maximized). These policies are thus updated as an agent gains more information about the environment, allowing it to choose more effective actions with less uncertainty about their effects.

Formally, a POMDP consists of a tuple $\langle S, A, Z, p(s'|s,a), p(z|s,a), r(s,a), \gamma \rangle$, where S is a set of states s, A is a set of actions a, Z is a set of observations z (Sondik, 1971); each component is described in more detail below. As shown in Fig. 1, an action a is taken at each time step, which together with the current state s results in a transition to the next state s'. These transitions are specified by the transition model $p(s'|s,a)$. The state s at any point is unobserved. Instead, information about the state is indirectly

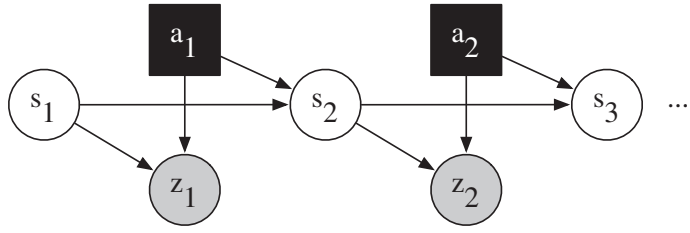

Fig. 1. Graphical model representation of a POMDP. At each time step t, there is an unobserved state s_t and the planner chooses an action a_t. Based on the current state and chosen action, the planner observes z_t, and the state transitions to s_{t+1}.

available via the observations. Given that action a is taken in state s, the observation model $p(z|s,a)$ indicates the probability of observing z. For example, in the case of the robot moving from an unknown starting location, the state s is its true location in the maze, and it chooses whether to move or to turn in a particular direction. The next state s' is the robot's location after that action, which may still be unknown to the robot. However, the robot does know something about its location based on the walls and intersections it sees; this information is encoded by the observation model $p(z|s,a)$. In this paper, we assume that the observation and transition models are known.

Taking one action versus another in a particular state may be more or less costly. Agents may experience either rewards or costs based on their actions; here, "costs" are simply negative rewards. Because we will be mainly referring to costs in our experiments, we describe the POMDP planning framework in terms of costs, but it is also possible to have environments that have both rewards and costs, or only rewards. The costs experienced by an agent may vary based on its objectives and the environment. For instance, in the case of the robot trying to find the charging station, it might wish to accomplish its task in minimal time; the cost structure would then specify that actions that are likely to take more time (e.g., moving longer distances) will incur higher costs. Alternatively, the robot might simply wish to find a charging station before it runs out of energy. In that case, all actions might have cost zero, but entering a state where the robot has no energy would incur a very large cost. The cost model $r(s,a)$ encodes the cost structure; for every state s and action a, this model specifies a real-valued cost. POMDP planning seeks to choose actions that minimize the expected sum of discounted future costs. If the state were known at each time step, this quantity could be calculated as $\Sigma_{t=0}^{\infty}\gamma^t r(s_t, a_t)$, where γ is the discount factor. This factor represents the relative harm of immediate versus delayed costs. If γ is close to one, then costs incurred in the future have almost as much harm as costs incurred now, since γ^t will remain close to one. As γ gets closer to zero, γ^t will be close to zero for even small t; this means that immediate costs will dominate. The discount factor is set by the planner.[2]

Given a POMDP, we want to compute a *policy* for how to select an action at each time step. An optimal policy should map the prior history of actions and received observations to the action that will minimize the expected sum of discounted future costs. Since the prior history grows at each step, this may be difficult to compute directly. A

common alternative is to maintain a sufficient statistic, known as a *belief state b*, that represents the planner's distribution over potential states given the past actions and observations (Åström, 1965). In the case of the robot, this would be a distribution over where it currently is in the maze; some locations might have been ruled out by the observations, while other locations might be equally probable given the observations and actions that have occurred so far. Bayesian updating can be used to compute a new belief b^{az} after taking action a and receiving observation z from belief b:

$$b^{az}(s') = \frac{p(z|s',a)\int_s p(s'|s,a)b(s)ds}{\int_{s'} p(z|s',a)\int_s p(s'|s,a)b(s)dsds'}. \quad (1)$$

Intuitively, this update corresponds to taking the expectation over the next state given the distribution over states at the current time step, adjusted by the probability of receiving the actual observation in that next state. Thus, at each time step, the planner chooses an action to minimize future cost, where information about the past is encoded by the current belief state. This process is equivalent to planning based on maintaining the entire history of actions and observations.

3. Modeling teaching as a POMDP

We now seek to formalize the problem of selecting individual teaching actions within the POMDP framework. Using POMDPs for education was first mentioned by Cassandra (1998), as part of a proposal for diverse applications for POMDPs. POMDP planning has been considered as a way of sequencing units of instruction (Brunskill & Russell, 2010; Brunskill et al., 2010), and simulation-based work has considered how to approximate the student state to use POMDP planning for domains where a "soft" prerequisite model is known (Folsom-Kovarik et al., 2010). Finally, Theocharous et al. (2009) considered the problem of constructing the component models of a POMDP to teach a specific concept. In our work, we consider the problem of selecting individual pedagogical actions using a POMDP policy, where we do not have explicit information about which actions should precede others and where the student's knowledge state does not necessarily decompose into independent components. While many common models of student learning assume this decomposition (e.g., ACT-R, Anderson, 1993), other psychological models of learning are not presented in this form (e.g., Tenenbaum, 1999), and the POMDP framework can be applied to either type of model. In this section, we demonstrate how teaching can be modeled within the POMDP framework by mapping each part of the model to a particular part of the teaching process.

Fig. 2a shows our general formulation of the teaching process. The automated teacher must make a sequence of pedagogical choices. These pedagogical choices map to the actions taken at each time step in the POMDP (see Fig. 2b). For example, the automated teacher might first have a student complete a short quiz (a pre-test), and then have the

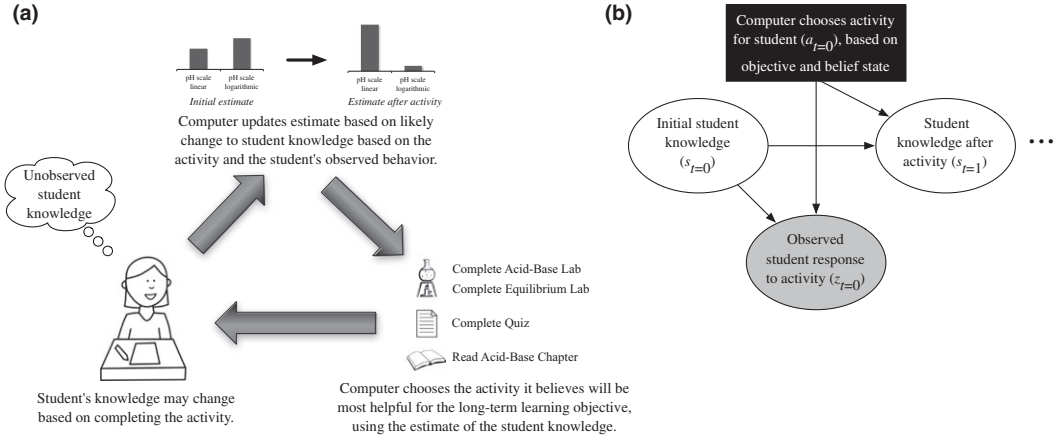

Fig. 2. Mapping the POMDP model to teaching. (a) The teaching process consists of the computer choosing actions, which may be dependent upon its knowledge of the student. The student knowledge evolves based on the activities that she completes. By observing the student's behavior, such as how the student completes a lab, the computer can gain information about what the student understands and what misunderstandings she may have. (b) A graphical model representation showing how teaching corresponds to the components of the POMDP. Actions are pedagogical choices, states correspond to student knowledge, and observations are made of student behavior. The computer teacher can make pedagogical choices to achieve some long-term objective, such as minimizing the time for the student to reach mastery.

student complete a chemistry lab that has concepts that it believes the student has not yet mastered. The learner's state at any given time step t is unobserved and corresponds to the state s_t of the POMDP at time t. We consider the state to be a knowledge state, corresponding to what a learner currently understands about what she is being taught. However, the state could in general be richer and include other information about the learner, such as her current level of motivation or her current affect. The learner's state may change based on the activity that the teacher chooses to give her (i.e., a_t); as shown in the graphical model in Fig. 2b, the next knowledge state is dependent only on the current knowledge state and the pedagogical activity. Intuitively, changes in the learner's state correspond to learning. Such changes may reflect mastery of a new skill, forgetting of a previously learned skill, or some other change in understanding, such as making a new generalization that brings the student closer to correct understanding.

3.1. Specifying learner models

Modeling learner knowledge within the POMDP framework requires specifying the space S of possible knowledge states and a transition model $p(s'|a, s)$ for how knowledge changes. Different learner models may make different assumptions about how knowledge is encoded. At the simplest level, the state might only represent whether a particular skill, such as addition, has been mastered. Alternatively, states might represent different possible understandings a student might have about addition, one normative and others

representing incomplete or non-normative understandings. While a particular representation must be specified to compute a policy, one of the advantages of the POMDP framework is that it can work with a variety of possible representations, allowing one to determine what effects different assumptions would have on the optimal policy and how quickly one would expect a particular type of learner to master a skill.

A second part of the learner model, the observation model, maps to modeling the learner's behavior given her knowledge. Intuitively, this model provides noisy information about the learner's understanding by specifying the probability $p(z|s,a)$ that a learner will give a particular response z to an item given her current knowledge state s. For example, imagine the student has been asked for the answer to $3 + 8$ and responds 10. This response is less likely than 11 if the learner has a correct understanding, but it could have occurred due to misreading or a "slip": It is not definitive proof that the student has not mastered addition.

Given that the automated teacher has a learner model, it can update its beliefs about the learner's current knowledge state based on new observations. The teacher begins with a belief state equal to the prior distribution $p(s)$ over possible knowledge states; this distribution might be used to encode known biases in student knowledge for a particular task. After each action, the belief state is updated using the observation model to incorporate the learner's responses and the transition model to incorporate the effects of learning (as in Eq. 1).

3.2. Pedagogical objectives

The final portion of the POMDP framework that must be adapted to the teaching domain is the cost model. The content of this model is dependent on the teacher's objectives. Since POMDP planning is used to find a policy that minimizes expected costs (or maximizes rewards), the cost model should specify the teacher's desired outcome as well as incentives for individual actions or states. For example, one simple learning objective would be to have the learner reach the knowledge state s that reflects mastery in as little time as possible. This could be encoded by having actions in state s cost zero, and other actions cost the expected time for the student to complete them. Then, for instance, if there are only two actions and the student will almost certainly enter state s after completing either action, the planning framework will favor the action with shorter expected time.

While we will use this simple time cost model, many other possible objectives could be encoded in a cost model. For instance, rather than seeking complete mastery of all material, the objective could be for a student to learn as much material as possible in a particular domain. This objective might more naturally be represented using rewards (rather than costs, which are negative), with larger rewards for states where more of the material has been learned. The resulting policies would still attempt to teach all material, but if only a fixed time was available, these policies would focus on some material rather than trying to teach all material with little probability of success. We discuss other objectives in the Discussion.

Formulating teaching as a POMDP has several advantages. It provides a way of deriving an optimal policy for any teaching task and any learner model. This allows one to explicitly determine the expected consequences of making different assumptions about the learner or changing the learning objective. This can be helpful for evaluating the learner model and for determining whether a particular distinction actually has implications for teaching. The general framework is naturally modular, separating the parts of the teaching task and the assumptions made in each part of the model. This may be helpful for comparing or making improvements to automatic tutoring systems. Specifying a general framework also allows one to consider how particular methods for problem selection may be approximations to the optimal policy. By defining a problem selection method with respect to how it approximates the POMDP policy, one can make use of the existing POMDP literature and evaluate in what circumstances such an approximation is likely to perform well.

In the remainder of the paper, we consider how this framework can be applied in concept-learning tasks. In such tasks, we use the time-based cost model described above, such that the computed policies select actions to minimize the expected time for the learner to understand the concept. The space of tutorial actions may vary widely based on the domain being taught. We follow the tradition of concept learning via examples that have been explored in psychology (e.g., Feldman, 1997; Nosofsky, 1998). Within this type of concept learning, it is natural to consider three types of actions: *examples*, *quizzes*, and *questions with feedback*. *Examples* give the student information about the concept, but they do not result in any observed behavior from the learner. *Quizzes* ask the student a specific question about the concept, but they do not directly give the student new information about the concept. *Questions with feedback* combine these two action types by first asking a question and then responding by telling the student the correct answer. *Example* and *question with feedback* actions are equivalent to the *tell* and *elicit* pedagogical actions that have been used previously in optimizations of intelligent tutoring systems (Chi et al., 2008). The POMDP can be used to find the optimal policy for teaching the learner the concept, taking into account the learner's responses to questions and balancing actions aimed primarily at diagnosis with those that provide information to the learner.

4. Learner models for concept learning

We consider three learner models, inspired by the cognitive science literature, that correspond to restrictions of Bayesian learning. Each learner model describes the state space of the POMDP as well as the transition and observation models. While the models we describe are only rough approximations of human concept learning, we will show that they are still sufficient to enable us to compute better teaching policies and that they can be applied to several different concept learning tasks. The three models we consider vary in complexity as well as in how closely they approximate human learning; this allows us to examine how well the POMDP approach can scale to more complex models. By using several different models, some of which we know are better approximations of human

learning than others, we can also examine how closely the learner model must match human learning in order to lead to effective policies.

All of the models we consider share several assumptions about the concept-learning task. They each assume a discrete hypothesis space C of possible concepts. Such an assumption is reasonable in many contexts. For instance, the hypothesis space corresponding to possible meanings of a word might include binary vectors assigning each potential object as part of the concept or not. The models we consider assume that the size of the hypothesis space is finite, although it may be large. Since our models correspond to restrictions of Bayesian learning, they also assume that there is a prior distribution over the hypothesis space of concepts. This distribution intuitively represents learners' biases before they are exposed to any data about the concept. Finally, we assume that the domain is such that for any question the tutor might ask, each concept implies a single possible right answer. This assumption simplifies the problem somewhat, but it could easily be modified to assume that concepts specify a probability distribution over possible answers to a question.

4.1. Memoryless model

We first consider a model in which the learner's knowledge state is the single concept she currently believes is correct, similar to a classic model of concept learning proposed by Restle (1962). In this model, the learner does not explicitly store any information previously seen. If an action is a quiz action, or if the provided evidence in an example or question with feedback action is consistent with the learner's current concept, then her state stays the same. If the action contradicts the current concept, the learner transitions to a state consistent with that action, with probability proportional to the prior probability of that concept:

$$p(s_{t+1} = c_i | s_t = c_j, a_t) \propto \begin{cases} p_0(c_i) & \text{if } c_i \text{ is consistent with } a_t \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where $p_0(c_i)$ represents the prior distribution on concepts.

The observation model is deterministic: When asked to provide an answer to a question, the learner provides the answer z_n that is consistent with her current beliefs. While some prior work suggests that human performance in concept learning tasks like ours appears memoryless (Bower & Trabasso, 1964; Restle, 1962), it is possible that the model underestimates human learning capabilities. It thus provides a useful measure of whether POMDP planning can still accelerate learning when a pessimistic learner model is used. This model is also attractive because it is less computationally complex than the other models we consider: The size of the state space is equal to the number of possible concepts $|C|$. Given this state space, the automated teacher's belief state b over the hidden learner state is a probability distribution over the $|C|$. Belief updating is performed using Eq. 1, which will be an order $|C|^2$ operation.

4.2. Discrete model with memory

The key limitation of our first model is its lack of memory of past evidence. In general, this assumption is not accurate for human learning, although it is sometimes applicable to children (Levine, 1970). A more psychologically plausible state space is one in which learners maintain a finite memory of the past M actions in addition to their current guess of the true concept. This results in factored states that consist of the hidden guess at the true concept (s_c) and the fully observed history of the past M actions (s_h). Like the memoryless model, this model assumes that the learner stores her current guess at the true concept, and this guess is updated only when information is shown that contradicts the guess. In this case, the learner shifts to a concept that is consistent with the current evidence and all evidence in the M-step history. The transition probability is again proportional to the initial concept probability. The transition model for s_h, representing the history, is deterministic. The observation model is also the same as in the memoryless case: The learner responds deterministically based on her current guess. Belief updating for the automated teacher can be updated in the same manner as for the previous model, with the size of the state space now equal to the number of possible concepts multiplied by the number of possible memory states.

4.3. Continuous model

A more complex, but natural, view of learning is that the learner maintains a probability distribution over multiple concepts (Tenenbaum, 2000; Tenenbaum & Griffiths, 2001). Such an account allows one to model cases where a learner is unsure exactly which concept is correct but has ruled out some of the possibilities. Those concepts that she has not ruled out would all have non-zero probability. The state is then a $|C|$-dimensional, continuous-valued vector that sums to 1, where C is the set of possible concepts. The ith position corresponds to the probability mass that the learner places on the ith concept. The state space S is an infinite set of all such vectors, the simplex $\Delta_{|C|}$.

The transition function assumes that for quiz actions, each state transitions deterministically to itself, as in the previous two models. For example and question with feedback actions, state dimensions for concepts that are inconsistent with the provided information are set to zero. Letting $p(s_{(t+1)i})$ be the ith entry in the distribution at time $t + 1$, corresponding to the probability of the ith concept, then:

$$p(s_{(t+1)i}|s_t, a_t) \propto \begin{cases} p(s_{ti}) & \text{if } c_i \text{ is consistent with } a_t \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The full joint transition probability is then re-normalized. This corresponds to a Bayesian generalization model with weak sampling (Tenenbaum & Griffiths, 2001). The observation model assumes the learner gives answer z_n to a question with probability equal to the amount of probability she places on concepts that have z_n as the correct answer for

this question. For this model, the automated teacher must maintain a belief state over the infinite number of possible knowledge states. This requires approximating the belief state; we discuss the details of our approach to this issue in the next section on selecting teaching actions using POMDP planning.

4.4. Capturing deviations from the model

To improve the robustness of our policies to the coarse learner models we employ, all models include two extra parameters, a transition noise parameter ϵ_t and a production noise parameter ϵ_p. The transition noise parameter ϵ_t is the probability that the learner ignores a given teaching action and thus this action does not change the state. The production noise parameter ϵ_p is the probability that the learner produces an answer inconsistent with her current guess; this parameter is similar to the guess and slip parameters common in some models of student knowledge (Corbett & Anderson, 1995).

5. Selecting teaching actions using POMDP planning

Our goal is to compute a policy that selects the best action, given a distribution over the learner's current knowledge state, the belief state. Offline POMDP planners compute such policies in advance. This approach requires pre-computing policies over the continuous space of possible belief states.[3] The space of possible belief states is a simplex as each belief is a probability distribution over the possible knowledge states. As the number of knowledge states grows, the dimensionality of this simplex is increased. Thus, as the size of the state space increases, offline approaches become infeasible. Since many teaching domains are likely to have large state spaces, we instead turn to online POMDP forward search techniques, which have proven promising in other large domains (see Ross, Pineau, Paquet, & Chaib-draa, 2008, for a survey).

We compute the future expected cost associated with taking different actions from the current belief state by constructing a forward search tree of potential future outcomes (see Fig. 3). This tree is constructed by interleaving branching on actions and observations. To compute the values of actions next to the root belief state, the values of the leaf nodes are estimated using the evaluation function, and then their values are propagated up the tree, taking the maximum over actions and the expectation over observations. After the tree is used to estimate the value of each action for the current belief, the best pedagogical action is chosen. The learner then responds to the action, and this response, plus the action chosen, is used to update the belief representing the new distribution over the learner's knowledge state. We then construct a new forward search tree to select a new action for the updated belief.

While forward search solves some of the computational issues in finding a policy, the cost of searching the full tree is $O((|A||Z|)^H)$, where H is the task horizon (i.e., the number of sequential actions considered), and requires an $O(|S|^2)$ operation at each node. This is particularly problematic as the size of the state space may scale with complexity of the

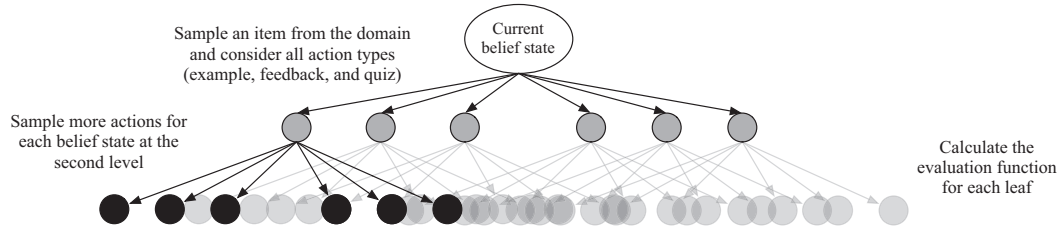

Fig. 3. An example forward search tree for a two step horizon. The search considers the effect of each type of action applied with several different items from the domain; for example, to teach the learner the concept "odd numbers," the search would consider several different numbers that could be used for examples, quizzes, or questions with feedback. The number of actions sampled at each time step in the figure is only an example; in our actual experiments we sampled varying numbers of actions.

learner model: The memoryless model has a state space of size $|C|$, while the discrete model with memory has state space of size $|C||A|^M$ and the continuous model has an infinite state space. To reduce the number of nodes we must search through, we take a similar approach to Ross, Chaib-draa, and Pineau (2008) and restrict the tree by sampling only a few actions. Additionally, we limit H to control the depth of the tree and use an evaluation function at the leaves. This evaluation function is based on the estimated probability that the student knows the correct concept.

Since the belief state in the continuous model is a distribution over an infinite set of states, we approximate the belief state for this model to make inference tractable. We represent the belief state as a weighted set of probabilistic particles and update these particles based on the transition and observation models (see Supplementary Materials for details). Particles inconsistent with the observations are eliminated. If no particles are consistent with the current observation, we reinitialize the belief state with two particles: one with a distribution induced by rationally updating the prior using all previous evidence and one with a uniform distribution. Depending on the number of particles used, this technique may be less computationally complex than the calculations for the other two models.

6. Empirical evaluation of optimal policies

In the remainder of this paper, we explore the effectiveness of the POMDP framework for teaching two concept-learning tasks, alphabet arithmetic and learning numerical concepts, in order to assess whether it is feasible to use in real time and if its actions are effective for teaching human learners. In alphabet arithmetic, learners infer a mapping from letters to numbers based on exposure to equations like $A + B = 1$ that impose constraints on the possible mappings. While this task is artificial, it provides a preliminary evaluation of POMDP planning for problem selection and shares several important characteristics with real teaching domains: It is rich enough that learners may have misunderstandings, such as erroneous beliefs about which letter maps to which number, and that

we expect some teaching policies to be more effective than others. We begin with teaching simulated learners alphabet arithmetic and then turn to teaching human learners. After showing that POMDP planning can be effective for teaching alphabet arithmetic, we turn to the more complex domain of numerical concepts. Concepts like "prime numbers" are taught by showing learners examples of numbers that are and are not a part of the concept. For this task, the space of possible concepts as well as the number of available teaching actions is much larger, providing a test of how well the framework can scale to the larger space. While these tasks differ from traditional learning tasks covered in tutoring systems used in the classroom and have some similarities with problem solving tasks, they are similar to other concept learning tasks in psychology (e.g., Bruner, Goodnow, & Austin, 1956; Shepard, Hovland, & Jenkins, 1961; Tenenbaum, 2000) and unlike traditional problem solving tasks, information is conveyed to learners over a period of time. Thus, the tasks require sequencing information and incorporating the learners' behavior to determine what information is most appropriate, matching the structure of many more typical learning tasks in education.

7. Simulation 1: Teaching simulated learners alphabet arithmetic

We first explore the performance of POMDP planning for teaching simulated learners alphabet arithmetic. These simulations address two questions: (a) How effective is POMDP planning when the assumed learner and the actual learner match, and (b) Is POMDP planning still effective when the actual learner differs from that assumed by the POMDP? As described above, alphabet arithmetic involves learning a mapping between letters and numbers; in this case, we teach a mapping from the letters $A - F$ to the digits 0–6. We assume learners have a uniform prior over mappings. For example actions, learners are shown an equation where two distinct letters sum to a numerical answer. For instance, A could be mapped to 0 and B to 1, and one might show the learner the equation $A + B = 1$. Quiz actions leave out the numerical answer and ask the learner to give the correct sum, providing the system with information about the learner's knowledge. Questions with feedback combine these two actions. The planning goal for alphabet arithmetic is to minimize the amount of time for learners to correctly identify the mapping. Given this space of actions, we expect that modeling and tracking the learner's state may speed learning since teaching actions can explicitly address or diagnose the learner's misunderstandings.

7.1. Methods

We conducted four simulations for each type of learner (memoryless, discrete with memory, and continuous). One simulation used a *random* policy, and the other three simulations used a POMDP policy driven by each of the learner models. This allowed us to determine how quickly, for instance, a memoryless learner could be taught using a

memoryless policy versus a *continuous* policy.[4] Fifty simulations were run for each of these twelve combinations of learner and policy.

To match the experiment presented in the next section, we alternated between *teaching phases*, in which we selected pedagogical actions for the simulated learner, and *assessment phases*, in which we checked whether the simulated learner had identified the mapping. Each teaching phase consisted of a sequence of three pedagogical actions. After the teaching phase, there was an assessment phase that varied slightly based on the type of simulated learner. For the memoryless learners and the discrete representation with memory learners, the current guess of the learner was compared to the true mapping. If they were identical, then the learner had mastered the concept and teaching was terminated. For the continuous learner, a mapping was sampled from the learner's current distribution over possible mappings, and if this mapping matched the true mapping, then teaching was terminated. If a learner failed to achieve mastery after 40 teaching phases, then teaching was also terminated; this was to match the experimental procedure in which teaching of human learners would be terminated after 40 phases regardless of performance.

Finding the POMDP policies requires setting the parameters of the cost function as well as the parameters of each learner model. To make the simulations as realistic as possible, we set these parameters based on data from teaching 20 human participants using a *random* policy; these were the participants in the control condition of Experiment 1, described below. The cost of each action that was used by the POMDP planner was the median time to complete each action type from the participants in the control condition: Example actions took 7.0 s, quiz actions took 6.6 s, and question with feedback actions took 12 s. When computing the action values within the forward search tree, we set the cost for a leaf node to be the probability of not passing the assessment phase multiplied by $10 \cdot \min_a r(a)$, a scaling of the minimum future cost.

We set ϵ_t, the probability of ignoring a teaching action, and ϵ_p, the probability of making a production error when answering a question, by finding the values that maximized the log likelihood under a given model of the data from the participants taught using a random policy; details of this procedure and the resulting values can be found in the Supplementary Materials.

For forward planning, we set the parameters of the algorithm to sample as many actions as possible given the constraints of planning in real-time (see Supplementary Materials for more details about the number of free parameters in the algorithm and how these parameters were set). In particular, we limited all computations to 3 s. Given this constraint, we set the lookahead horizon to two actions. Policies for the first nine actions were precomputed with ten actions sampled at each level. Caching the first nine actions allows us to consider more actions at each horizon, while still using a constrained number of actions to speed computations.

Later actions were computed by sampling the following number of actions at each level: seven and six actions for the *memoryless model*; eight and eight actions for the *discrete model with memory*; and four and three actions for the *continuous model*. Sixteen particles were used for the *continuous model*, and $M = 2$ for the *discrete model with*

memory (both for the simulated learner and the POMDP policy). The effects of varying these parameters are not extreme: Sampling more actions at each level results in less variance, but it does not tend to change the outcome across many simulations. The results are also not very sensitive to changes in the number of particles, although using a very small number of particles results in poor performance. The simulations for the discrete learner with memory are sensitive to changes in the assumed memory capacity of the learner; for instance, if memory capacity is set to zero, this learner is identical to the memoryless learner.

7.2. Results and discussion

For each type of simulated learner, we examined how the expected time to mastery varied based on the teaching policy. Expected time to mastery was computed by assuming that each action took the amount of time assumed by the POMDP planner. For instance, if a simulated learner identified the mapping after three actions, two of which were example actions and one of which was a quiz action, the expected time to mastery would be $2 \cdot 7.0 + 1 \cdot 6.6 = 20.6$ s.[5] Initial inspection showed that the distribution of learning times exhibited a long right tail, so we analyzed results using medians, which are more robust than means to outliers and non-symmetric distributions.

Fig. 4 includes results for this simulation and Simulation 2, which includes several stronger baselines policies for comparison. These results demonstrate that teaching using a POMDP policy can decrease the expected time to mastery relative to the random policy, even if the learner does not match the learner model in the POMDP. For all three learner types, there was a significant main effect of teaching policy on the expected time to mastery (Kruskal–Wallis: memoryless learner, $\chi^2(3) = 21.4$, $p < .001$; discrete representation with memory, $\chi^2(3) = 26.7$, $p < .001$; continuous learner, $\chi^2(3) = 46.5$, $p < .001$). We performed planned, pairwise comparisons between the POMDP policies and the *random* policy. For all three simulated learners, the POMDP policies significantly improved learning efficiency (see Table 1).

We also examined the simulations to see what types of policies emerged from different assumed learner models. Fig. 5 shows part of the *discrete model with memory* policy. Here, the model shows an example, and then follows this with a quiz question. The question allows the model to check whether the learner has mastered part of the concept, and it is less costly than showing another example, since quiz actions are generally completed more quickly than example actions. After the example, the next action is dependent on the response that is given by the student, as the POMDP updates its belief state using the observation model. Some responses may lead to the next action being an example while others may result in more quiz questions, either because the student responded correctly or because the model is attempting further diagnosis. The pattern of differential action types based on whether the quiz was answered correctly was typical of the planner's strategies, although the exact strategy was dependent on previous actions and observations as well as the planner's confidence in the learner's knowledge state. Questions with

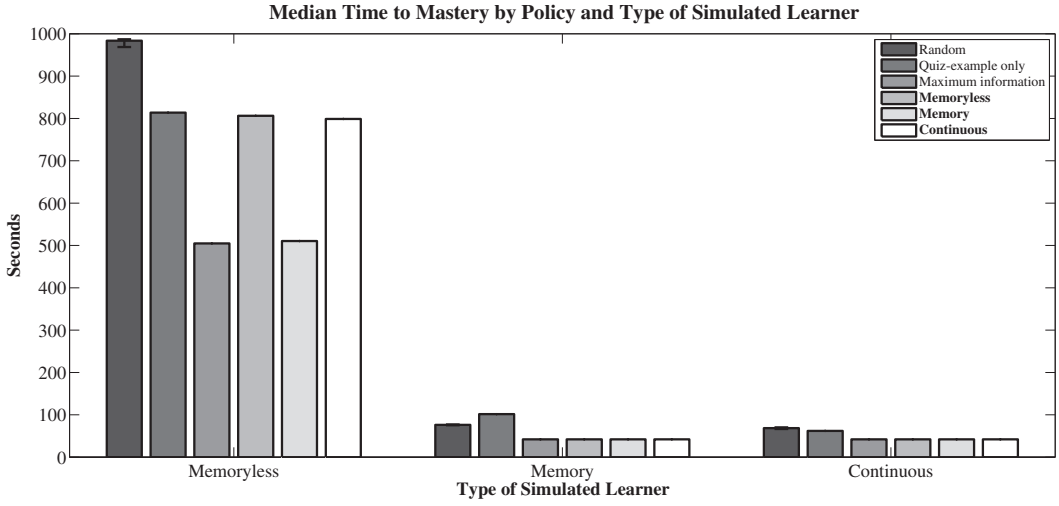

Fig. 4. Results for Simulations 1 and 2: Median time for simulated learners to reach mastery, by policy type. Error bars correspond to bootstrapped 68% confidence intervals (equivalent to one standard error). Bolded conditions are POMDP policies.

Table 1
Simulation results

Simulated Learner	Comparison	$\chi^2(1)$	p
Memoryless	*memoryless* versus *random*	18.8	<.001
Memoryless	*discrete model with memory* versus *random*	11.7	<.001
Memoryless	*continuous* versus *random*	7.1	<.01
Discrete with memory	*memoryless* versus *random*	17.4	<.001
Discrete with memory	*discrete model with memory* versus *random*	15.2	<.001
Discrete with memory	*continuous* versus *random*	18.1	<.001
Continuous	*memoryless* versus *random*	31.9	<.001
Continuous	*discrete model with memory* versus *random*	18.5	<.001
Continuous	*continuous* versus *random*	35.2	<.001

Note: Planned pairwise comparisons using Kruskal–Wallis tests were conducted to assess differences in expected time to mastery based on the teaching policy used. For all three learners, all POMDP policies were significantly more effective than the random policy.

feedback were rarely used by any of the POMDP policies because of their high cost relative to their effectiveness.

There were some differences based on which learner model was assumed, and which type of learner was being taught. Overall, the policies based on the *discrete model with memory* and the *continuous model* used an average of at least 90% example actions, with the remaining actions being almost exclusively quiz questions. One exception was the case of the *discrete memory model* for teaching and a continuous learner, where one simulation never reached mastery and the policy devolved into asking primarily quiz questions. Given a longer period to teach, this policy would presumably diagnose the

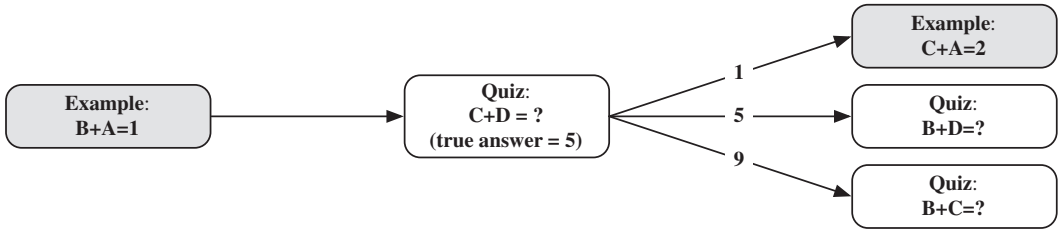

Fig. 5. Part of a policy from the *discrete model with memory*. Possible student answers to the quiz are indicated on the arrows; some are omitted. Based on the student's response, the action after the quiz may correct a misunderstanding, try to better diagnose the cause of an incorrect answer, or continue quizzing to try to detect a misunderstanding. Actions after the quiz are contingent upon the student's response, reflecting the fact that action choices are based on the computer's beliefs about the student's knowledge, which are updated given the student's behavior.

learner's incorrect belief, and then use an example to correct that belief. The proportion of quiz versus example actions was more variable for the *memoryless* policy, probably because the memoryless model assumes a much more stochastic learner than the other two policies. This means that the model has less certainty about the learner's knowledge state, and thus less guidance for choosing what questions to ask the learner.

In summary, POMDP policies were in general effective at improving learning efficiency even when the assumed policy was incorrect. While different types of simulated learners varied dramatically in the time required to master a concept, these differences were generally not greatly mediated by which type of learner was assumed by the POMDP planner. Based on the type of learner assumed by the POMDP planner, however, the policies do have qualitatively different characteristics.

8. Experiment 1: Teaching human learners alphabet arithmetic

We next turn to a behavioral experiment to explore whether the findings from the simulations also hold when teaching human learners. Such an investigation is necessary since human learners may vary more dramatically and in different ways than the simulated learners.

8.1. Methods

8.1.1. Participants

A total of 40 participants were recruited online and received a small amount of monetary compensation for their participation.

8.1.2. Stimuli

All participants were randomly assigned three mappings between the letters $A-F$ and the numbers 0–5. These mappings were learned in succession.

8.1.3. Procedure

Participants were assigned to either the control condition, in which teaching actions were chosen based on the *random* policy, or to the experimental condition. Assignment to condition was based on the time of participation, with all participants in the control condition assigned prior to participants in the experimental condition. This allowed us to use the results from the control condition to set the parameters of the POMDP models, as described above. We have no reason to believe this biased the results. Each participant in the experimental condition experienced all three of the teaching policies in random order, one for each mapping learned. The experiment consisted of a sequence of teaching and assessment phases. In each teaching phase, a series of three teaching actions was chosen based on condition. After each teaching phase, participants completed an assessment phase in which they were asked to give the number to which each letter corresponded. Answers in the assessment phase were not used to update the beliefs of the POMDP models to allow for fair comparisons across conditions. Teaching of a given mapping terminated when the participant completed two consecutive assessment phases correctly or when 40 teaching phases had been completed. Within all phases, the equations the participant had seen were displayed on-screen, and participants could optionally record their current guesses about which letter corresponded to which number.

8.1.4. Computing policies

The cost for each action type and the setting of the ε parameters was the same as that in the simulations above. In all conditions, we also inserted a 3 s delay between actions in order to allow time for planning.

8.2. Results and discussion

We compared the amount of time participants took to learn each mapping and, as in the simulations, analyzed results using medians. There was no significant within-subjects difference in the amount of time or number of phases to learn the first, second, or third mapping (Kruskal–Wallis $p > .8$).[6]

Overall, participants taught by POMDP planning took significantly less time to learn each mapping (232 s vs. 321 s; Kruskal–Wallis: $\chi^2(3) = 16.5$, $p < .001$); see Fig. 6. Planned pairwise comparisons show that all POMDP policies but the *memoryless* policy resulted in significantly faster learning (Kruskal–Wallis: *memoryless* versus *random*, $\chi^2(1) = 2.9$, *n.s.*, $p = .087$; *discrete model with memory* versus *random*, $\chi^2(1) = 7.4$, $p < .01$; *continuous* versus *random*, $\chi^2(1) = 12.5$, $p < .001$).

As in the simulations, differences in policies occurred based on the learner model used. Policies for both the *discrete model with memory* and the *continuous model* began with six independent equations that fully specify the mapping. This is the policy one might have hand-crafted to teach this task, demonstrating that despite approximations in planning, the POMDP planner finds reasonable teaching policies. Each of the policies for these two models gives examples until there is a high probability the learner is in the

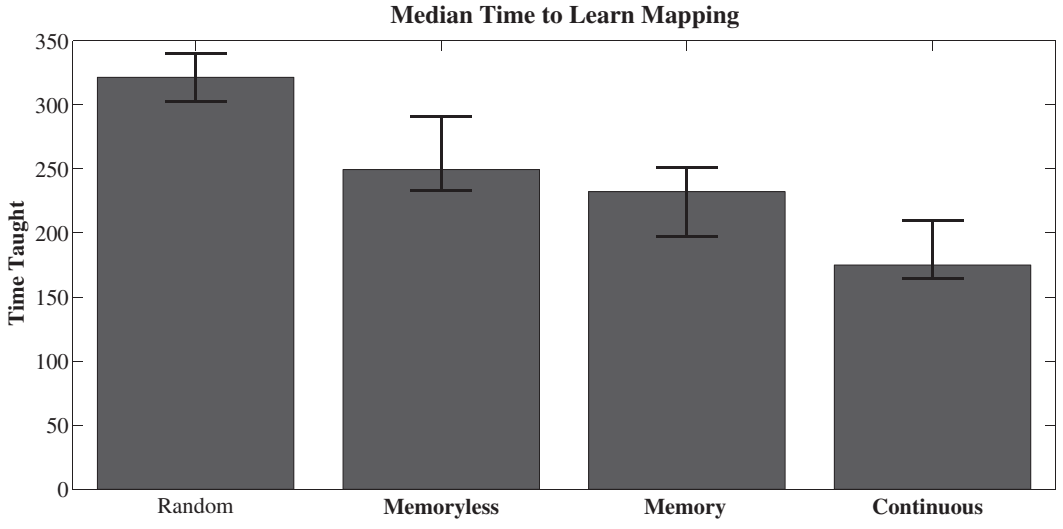

Fig. 6. Median time to learn each mapping in Experiment 1, by policy type (POMDP policies are bolded); error bars correspond to bootstrapped 68% confidence intervals (equivalent to one standard error). The *continuous* and *discrete model with memory* policies result in significantly shorter time to mastery than the *random* policy.

correct state, and then asks quiz questions, which are less costly than examples, to detect errors in the learned mapping.

The *memoryless* policy repeats specific example actions more often than the other policies since it assumes that the learner does not store previous actions in memory. This is clearly a pessimistic assumption, especially given that previously seen equations were displayed on-screen during the experiment. The fact that this model did not significantly decrease time to learn suggests such an unrealistic assumption may be detrimental for problem selection. However, the actions that are chosen do seem to be those that most limit the number of consistent hypotheses given both the structure of the mapping and the immediate preceding action, suggesting that we are finding a relatively good policy given the constraints of this learner model.

In the simulations, the types of actions chosen varied based on the assumed learner model. These variations persisted in the experiment. Fig. 7 shows number of actions of each type at each point in time in the experiment where at least three participants remained. Overall, the *continuous* policy asked the fewest quiz questions, while the *memoryless* policy asked the most (39% of actions). The *memoryless* policy tended to ask quiz questions later than the *continuous* policy, though, resulting in fewer participants receiving any quiz questions. The *memoryless* policy likely asked questions more frequently than the other policies because the state of a memoryless learner after an example is known with less certainty than in the other two models: In those models, the new state is constrained to be consistent with multiple pieces of past evidence, whereas the memoryless learner's state is constrained only to be consistent with the current example. None of

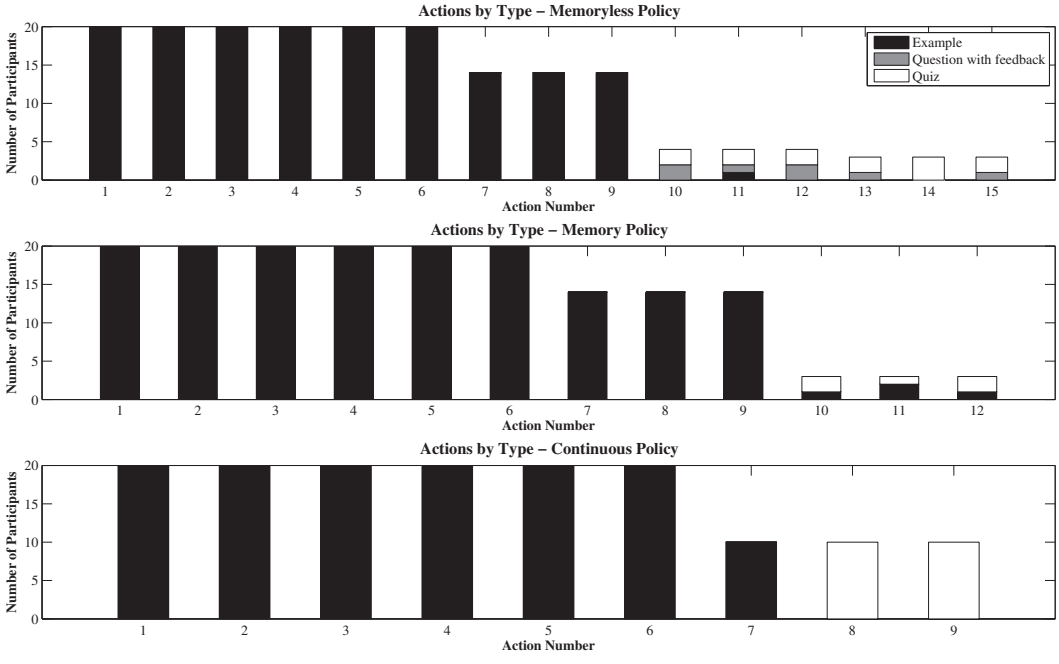

Fig. 7. Action types at each time in the experiment, by condition. Each graph shows the number of participants for whom the *n*th action was an example, question with feedback, or quiz. Data are shown only for time points where at least three participants had not mastered the concept.

the policies used many feedback actions due to the fact that these actions are considerably more expensive than other actions.

While the actions did vary between the policies, one might wonder why these different policies had little variance in their teaching efficacy. Specifically, while the two more complex learner models significantly reduced the time to learn, they did not result in significantly different outcomes from one another or from the *memoryless* policy. However, the simulation predicts this result if our learners are similar to the discrete learner with memory or the continuous learner. In both of these cases, all of the teaching policies were equivalent to one another. These two learners are quite powerful, and they are able to master the concept within only a few trials; thus, we might expect that there are multiple optimal or near-optimal policies, such that using a policy from one model is still relatively effective for a different model. Since we perform only approximate POMDP planning, we expect that the computed policies are only near-optimal even with respect to the given learner model. Additionally, the lack of difference in outcomes from using different teaching policies may be due to the fact that learning takes place in a small number of actions and we assess knowledge only every three actions; this means we cannot detect small changes in the number of actions required for mastery.

To summarize, the POMDP policies were effective for teaching human learners an alphabet arithmetic mapping, with policies based on the two more complex learner

models significantly decreasing time on task as compared to a random control policy. These gains are similar to what we predicted based on the simulation results. This suggests that while the human learners may not exactly match any of our learner models, the differences are not so large as to prevent these models from being effective guides for pedagogical decisions.

9. Stronger baseline policies for alphabet arithmetic

Experiment 1 showed that the POMDP planning framework could decrease the time to learn a mapping relative to a *random* policy. We now consider two alternative comparison policies: a random policy with only quizzes and examples (*quiz-example only*), and a policy that chooses the example that would result in the maximum information gain for the learner (*maximum information*). If the POMDP policies are leading to faster learning only due to avoiding costly actions, rather than by choosing effective sequences of actions based on the learner's likely understanding, then we would expect the *quiz-example only* policy to perform as well as the POMDP policies. Comparison to the *maximum information* policy, which does not consider the consequences of its actions beyond their immediate effects, can help to evaluate how important planning is in this task, especially given that planning requires us to consider only a randomly sampled subset of possible actions at each step. We first briefly describe the two new policies and their effectiveness in simulation, and then turn to a second behavioral experiment.

9.1. New baseline policies

The *quiz-example only* policy was included since the results of Experiment 1 indicated that *questions with feedback* took longer than the other actions for learners to complete. Since these actions will make up roughly one third of all actions in the random policy, we thought this might lead to slower learning with the *random* policy based on the mix of actions rather than the intelligent sequencing of actions. While part of the power of the POMDP policies is to decide what mix of actions is appropriate, we believe that the *quiz-example only* policy provides a comparison that might add value over a *random* policy while not including a model of the learner.

The *maximum information* policy is similar to the POMDP policies in that it includes a model of the learner, but it does not plan over multiple time steps and is not as flexible in the types of learner models that it can be paired with. The *maximum information* policy calculates which action will produce the maximum information gain for the learner, where information gain is defined as the difference between the Shannon entropy of the learner's state before the action and the entropy of the new state after the action has been taken. The entropy of the state is calculated as $-\Sigma_{i=1}^{n} p(c_i) \log(p(c_i))$, where c_i is the ith concept (Shannon & Weaver, 1948); when much of the probability is on only a few concepts, the entropy of the state will be low, while a uniform distribution corresponds to the highest possible entropy. This quantity has been used in other work for selecting data

for human and computer learners (e.g., MacKay, 1992; Tang et al., 2010). This policy only considers examples, since quizzes are assumed to not change the learner's state and as mentioned above, *questions with feedback* are more time consuming for learners without increasing the information gain. Because entropy will always be zero when the learner only has a single hypothesis, this *maximum information* policy requires a learner model where the hypothesis can be represented as a distribution over multiple concepts; this policy thus assumes the continuous learner model. In general, we would expect this model to perform relatively well, although it could perform poorly in cases where learners drastically diverge from the model's assumptions or where planning over multiple time steps leads to better policies.

9.2. Simulation 2: Performance of additional controls for alphabet arithmetic

We simulated the two new control policies and compared the results to those described in the initial simulations. As shown in Fig. 4, the *quiz-example* only policy generally performs similarly to the *random* policy. Both result in slower learning than the POMDP policies. In contrast, the simulated learners learn quite quickly from the *maximum information* policy. None of the POMDP policies are significantly faster than this policy. The *maximum information* policy is effective because this domain allows information to be progressively incorporated, such that the amount of information gained from a single action is a good heuristic for the overall progress in learning the concept. However, it is promising for the POMDP model that it does not in most cases fare worse than the *maximum information* policy, despite the approximations necessary to carry out planning. The POMDP model still maintains the advantage of being able to work with a broader array of learner models and to consider the benefits of diagnosing the learner's knowledge in addition to the benefits of trying to change that knowledge. Overall, these results suggest that while POMDP planning is not the only way to effectively select pedagogical actions, this method generally performs as well or better than the comparison methods for alphabet arithmetic.

10. Experiment 2: Effectiveness of new control policies for alphabet arithmetic

We conducted a second behavioral experiment to replicate the effective performance of the three POMDP policies and to examine the performance of the two new policies.

10.1. Methods

10.1.1. Participants

A total of 100 participants were recruited online and received a small amount of monetary compensation for their participation.

10.1.2. Stimuli

All participants were randomly assigned three mappings between the letters $A-F$ and the numbers 0–5. These mappings were learned in succession.

10.1.3. Procedure

Participants were assigned to be taught by one of the five policies. Unlike in Experiment 1, participants taught by a POMDP policy were taught by the same type of policy for all three mappings, rather than one mapping being taught by each of the policies. Thus, each policy was used to teach twenty participants. The remainder of the experimental procedure was the same as in Experiment 1.

10.2. Results

As shown in Fig. 8, the POMDP policies were more effective than the *quiz-example only* policy and about as effective as the *maximum information* policy, mirroring the simulation results. As in Experiment 1, there was a significant effect of policy on the time to learn each mapping (Kruskal–Wallis: $\chi^2(4) = 55.6$, $p < .0001$). Planned-pairwise comparisons showed that the POMDP policies and the *maximum information* policy were all significantly more effective than the *quiz-example only* policy (Table 2). With correction for multiple comparisons, the *maximum information* policy and the POMDP policies were not significantly different from one another (Table 2).

In Experiment 2, participants taught by the POMDP policies tended to learn the mappings more quickly than when taught by the POMDP policies in Experiment 1, and all three POMDP policies had very similar median times to mastery. One reason for this

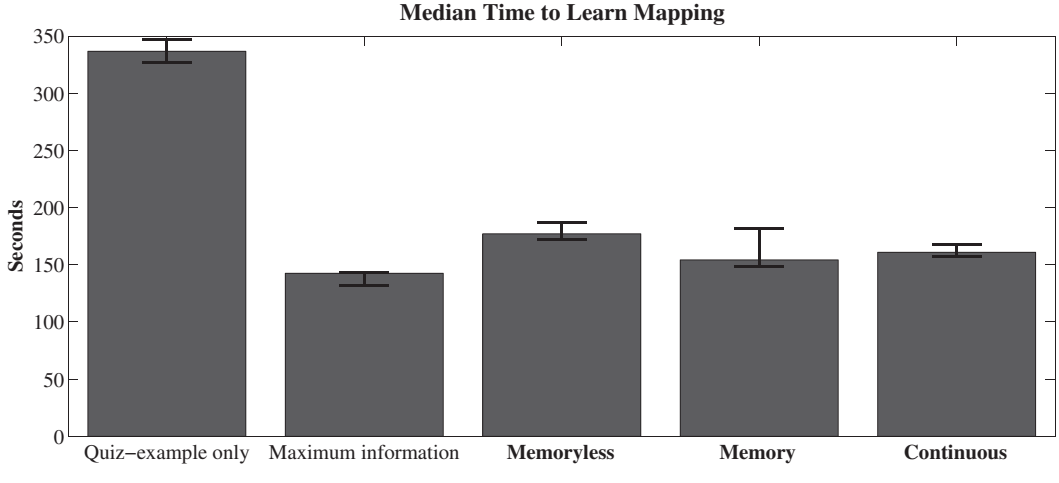

Fig. 8. Median time to learn each mapping in Experiment 2, by policy type; error bars correspond to bootstrapped 68% confidence intervals (equivalent to one standard error). The POMDP policies (bolded) tended to result in faster learning than the two random policies.

Table 2
Experiment results

Comparison	$\chi^2(1)$	p
memoryless versus *quiz-example only*	23.0	<.0001
discrete model with memory versus *quiz-example only*	26.2	<.0001
continuous versus *quiz-example only*	41.7	<.0001
maximum information versus *quiz-example only*	40.0	<.0001
memoryless versus *maximum information*	3.98	=.0461
discrete model with memory versus *maximum information*	1.11	>.2
continuous versus *quiz-example only*	1.29	>.2

Notes: Planned pairwise comparisons using the Kruskal–Wallis tests were conducted to assess differences in expected time to mastery based on the teaching policy used. All other policies resulted in significantly less time to learn than the *quiz-example only* policy, and after correcting for multiple comparisons, no significant differences were found between any of the three POMDP policies and the *maximum information* policy.

discrepancy may be because participants in Experiment 2 were always taught by the same policy. This may result in more familiarity with how a concept is taught, resulting in faster learning. This is reflected in the data: Unlike in Experiment 1, there was a significant effect on the time to mastery based on whether the mapping was the first, second, or third mapping learned (Kruskal–Wallis: $\chi^2(2) = 59.6$, $p < .0001$). A follow-up multiple comparison test showed that the second two mappings were learned significantly more quickly than the first mapping.

Overall, Experiment 2 replicates the main results of Experiment 1, demonstrating that the POMDP policies are more effective than a simple control policy. The trend of increasing complexity in the learner model leading to faster time to mastery was not replicated in this experiment, suggesting that simple models can be effective even if they are known to not match human learners exactly. This experiment was unable to determine whether the POMDP policies are more effective than the *maximum information* policy, which represents a relatively sophisticated way of automatically choosing pedagogical actions.

11. Evaluating effectiveness in a larger state space: The number game

In our third experiment, we explore whether the POMDP framework can accelerate learning in a larger and more complex concept space, the space of numerical concepts used in the Number Game (Tenenbaum, 2000). In the Number Game a participant is trying to infer a number concept, which consists of a subset of numbers between 1 and 100. For example, both "even numbers" and "numbers that end in three" are possible concepts. We use a hypothesis space consisting of the 6,412 most psychologically salient of the 2^{100} possible concepts, and a hierarchical prior p_0 over these concepts that was developed in prior work (Tenenbaum, 2000). In past Number Game research, information about the concept is typically given as a static set of one or more examples of numbers that are in the target concept, although other variations exist (Nelson & Movellan, 2001).

Good performance generally requires multiple examples, which suggests that a sequential teaching strategy has the potential to accelerate this process. We modify the Number Game so that learning occurs over a sequence of steps. Each step consists of one teacher action; since there are 100 numbers, and three action types, the action space A consists of 300 potential actions.

12. Simulation 3: Teaching simulated learners the number game

We begin with simulations of the Number Game to understand how differences between the simulated learner and the learner assumed by the POMDP policy affect the speed with which concepts are learned in this domain.

12.1. Methods

The methods for this simulation closely mirror those in Simulations 1 and 2: Simulations are conducted for each combination of learner (memoryless, discrete with memory, and continuous) and teaching policy (*random*, *quiz-example only*, *maximum information*, and the three POMDP policies). There are many different number concepts that could be taught. We taught three concepts in both this simulation and Experiment 3: multiples of seven; multiples of four minus one; and numbers between 64 and 83 (inclusive). Fifty simulations were run for each combination of simulated learner, teaching policy, and target concept. To match the experiment described below, the *random* and *quiz-example* only policies were modified to sample half of the numbers from within the concept and half from outside the concept; further explanation for this change is provided in Experiment 3.

As in Experiment 1, the simulations alternate between teaching and assessment phases. In each teaching phase, a series of five teaching actions was chosen based on condition. After each teaching phase, participants completed an assessment phase in which they were shown a sequence of ten numbers, five randomly chosen from within the concept and five from outside of the concept, and asked whether each number was in the concept. As in the previous experiment, answers in this phase were not used to update the POMDP models. Teaching was terminated when the simulated learner correctly responded to all numbers within a single assessment phase, or when 40 teaching phases had been completed.

To set the parameters of the POMDP policies, we followed the same procedure in Experiment 1, conducting the random condition prior to any other conditions and using these data to set action costs and ε parameters. The cost of each action type was the median time for participants in the random policy condition to complete these actions: 2.4 s for example actions, 2.8 s for quiz actions, and 4.8 s for question with feedback actions. The two additional parameters for each learner model, ϵ_p and ϵ_t, were again set to maximize the log-likelihood of the data in the control condition (see Supplementary Materials for details). Actions for the first four teaching phases (20 total actions) were

precomputed. For later actions, we set the planning parameters such that all models would take about 3 s to compute an action. As before, these parameters were not optimized, and small changes in their values did not have large impacts on the approximate policies. The lookahead horizon was set to three for the *continuous model*, and two for the other models. At each level, the following number of actions were sampled: six and eight actions for the *memoryless model*; six and six actions for the *discrete model with memory*; and six, six, and eight actions for the *continuous model*. Sixteen particles were used for the *continuous model*, and $M = 2$ for the *discrete model with memory*.

12.2. Results and discussion

For each type of simulated learner, we compared how the expected time to mastery varied based on the teaching policy. Expected time to mastery was computed by assuming that each action took the amount of time assumed by the POMDP planner, as in Simulation 1. Following our analysis of alphabet arithmetic, we analyze results using medians and use a bootstrapped Friedman's test to determine significance.

As shown in Fig. 9, there is considerable variation in effectiveness for each policy based on what type of learner is assumed as well as what type of concept is being taught. For the simulated continuous learner, there was a significant effect of teaching condition on time to mastery (Friedman: $\chi^2(5) = 111.5$, $p < .0001$). Both the *continuous* POMDP policy and the *maximum information* policy performed well, with neither outperforming the other. Planned pairwise comparisons between the POMDP policies and the control policies showed that the *continuous* policy outperformed the two random policies and there were no significant differences found between this policy and the *maximum information* policy; however, the *memoryless* and *discrete model with memory* policies were outperformed by the *maximum information* policy (Table 3).

For the simulations with the memoryless and memory learners, the *maximum information* policy tended to lead to slower learning than the discrete POMDP policies. While for the memory learner, the *maximum information* policy performed as well as the POMDP policies when aggregated across concepts, it performed relatively poorly for the range concept. This poor performance also occurred with the memoryless simulated learner, where the *maximum information* policy was significantly worse than the *memory* and *memoryless* POMDP policies (Friedman: *memoryless* versus *maximum information*, $\chi^2(1) = 55.4$, $p < .0001$; *discrete model with memory* versus *maximum information*, $\chi^2(1) = 65.0$, $p < .0001$). This discrepancy is due to the fact that the *maximum information* policy assumes that the learner remembers information, and narrows too quickly to only the endpoints of the range. We discuss this issue further in the results of Experiment 3.

For both the simulated memory and memoryless learners, there was a significant effect of teaching policy on time to mastery (Friedman: memoryless learner, $\chi^2(5) = 144.8$, $p < .0001$; memory learner, $\chi^2(5) = 296.9$, $p < .0001$). For both of these learners, the POMDP policies outperformed the two random control policies (Table 3). Overall, these simulations show that there is considerable variability in the best way to teach a concept

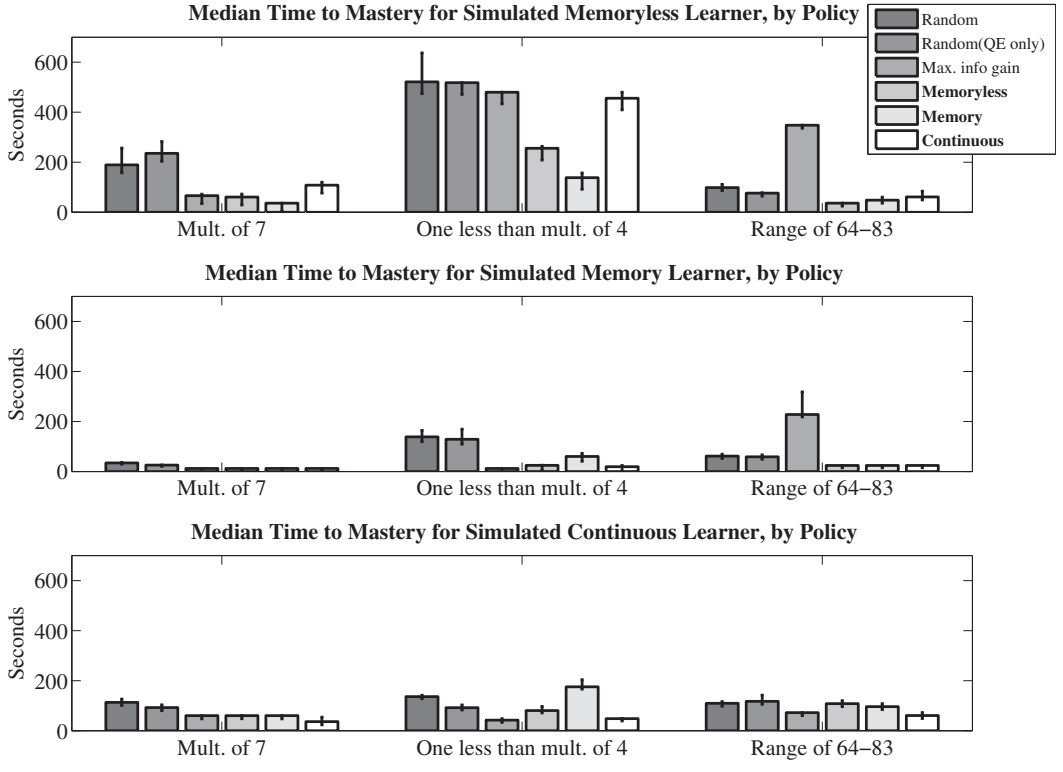

Fig. 9. Median time for simulated learners to reach mastery, by policy type (POMDP policies are bolded); error bars correspond to bootstrapped 68% confidence intervals (equivalent to one standard error). In some cases, mismatches between the actual and assumed learner type led to increases in time to mastery.

depending on the characteristics of both the learner and the particular concept being taught. In this larger and more complex domain, mismatches between the computer teacher's assumptions and the learner can lead to suboptimal policies and slower learning.

13. Experiment 3: Teaching human learners the number game

We now turn to experimentally investigating how well the POMDP policies can teach number concepts to human learners.

13.1. Methods

13.1.1. Participants

A total of 360 participants were recruited from the University of California, Berkeley, and received course credit for their participation.

Table 3
Number Game simulation results

Simulated Learner	Comparison	$\chi^2(1)$	p
Continuous	*memoryless* versus *random*	46.0	<.0001
Continuous	*discrete model with memory* versus *random*	25.3	<.01 (*n.s.*)
Continuous	*continuous* versus *random*	75.7	<.0001
Continuous	*memoryless* versus *quiz-example only*	33.0	<.0005
Continuous	*discrete model with memory* versus *quiz-example only*	12.2	>.1 (*n.s.*)
Continuous	*continuous* versus *quiz-example only*	62.6	<.0001
Continuous	*memoryless* versus *maximum information*	35.6	<.005
Continuous	*discrete model with memory* versus *maximum information*	56.4	<.0001
Continuous	*continuous* versus *maximum information*	6.0	>.1 (*n.s.*)
Discrete with memory	*memoryless* versus *random*	85.1	<.0001
Discrete with memory	*discrete model with memory* versus *random*	88.2	<.0001
Discrete with memory	*continuous* versus *random*	118.9	<.0001
Discrete with memory	*memoryless* versus *quiz-example only*	85.2	<.0001
Discrete with memory	*discrete model with memory* versus *quiz-example only*	88.3	<.0001
Discrete with memory	*continuous* versus *quiz-example only*	118.1	<.0001
Memoryless	*memoryless* versus *random*	81.9	<.0001
Memoryless	*discrete model with memory* versus *random*	91.6	<.0001
Memoryless	*continuous* versus *random*	47.6	<.0001
Memoryless	*memoryless* versus *quiz-example only*	73.8	<.0001
Memoryless	*memory* versus *quiz-example only*	83.5	<.0001
Memoryless	emphcontinuous versus *quiz-example only*	39.6	<.0001

Note: Planned pairwise comparisons using Friedman tests. The POMDP policies generally outperformed the random and quiz-example only policies, but they were outperformed by the *maximum information* policy except in the case of the continuous policy.

13.1.2. Stimuli

Each participant learned one randomly chosen number concept. The possible number concepts were the same as in Simulation 3: multiples of seven; multiples of four minus one; and numbers between 64 and 83 (inclusive).

13.1.3. Procedure

The procedure was similar to that in Experiments 1 and 2. Participants in Experiment 3 learned only a single concept, and they were assigned to be taught using actions chosen based on one of the three POMDP policies or by one of the three control policies (*random*, *quiz-example only*, or *maximum information*). Pilot testing demonstrated that randomly chosen teaching actions were extremely frustrating for participants, making disengagement likely, so we modified the *random* and *quiz-example only* policies to place higher probability on numbers within the concept. These policies first sampled whether to choose a number within the concept or a number outside of the concept, with equal probability on each of the two possibilities, and then sampled uniformly within the chosen class of numbers.

As in Experiment 1, participants alternated between teaching and assessment phases; these phases had the same structure as those used in Simulation 3. Within all phases, the

numbers that the participant had seen, as well as any category information that had been shown, were displayed on-screen.

13.1.4. Computing policies

As described in Simulation 3, parameters to compute the POMDP policies were set based on data from the *random* condition. Mirroring Experiment 1, a 3 s delay was inserted between all actions in all conditions to allow time for planning; if a model did not return an action within 3 s, the search was interrupted and the best action found so far was returned.

13.2. Results and discussion

We analyzed the median time on task for participants to learn the number concept, following the same methods as in Simulation 3. As shown in Fig. 10, there was a main effect of teaching condition: Participants taught using one of the POMDP teaching policies spent less time on task (Friedman: $\chi^2(5) = 65.8$, $p < .001$). Pairwise tests between each teaching policy and the *random* condition showed that each individual policy was more effective than the *random* policy (Friedman: *memoryless* versus *random*, $\chi^2(1) = 27.5$, $p < .001$; *discrete model with memory* versus *random*, $\chi^2(1) = 18.7$, $p < .001$; *continuous* versus *random*, $\chi^2(1) = 23.5$, $p < .001$). Unlike in Experiment 1, the *quiz-example only* policy was generally more effective than the *random* policy, and it performed as well as the POMDP policies in some cases. However, this policy performed poorly compared to the POMDP policies for the range concept (numbers between 64 and 83), and aggregated across concepts, each POMDP policy in general performed better than the *quiz-example only* policy.

While the teaching policies taught learners more quickly overall than the two random policies, these POMDP policies were not significantly different from one another in terms of effectiveness, and there was considerable variation in which policy was most effective

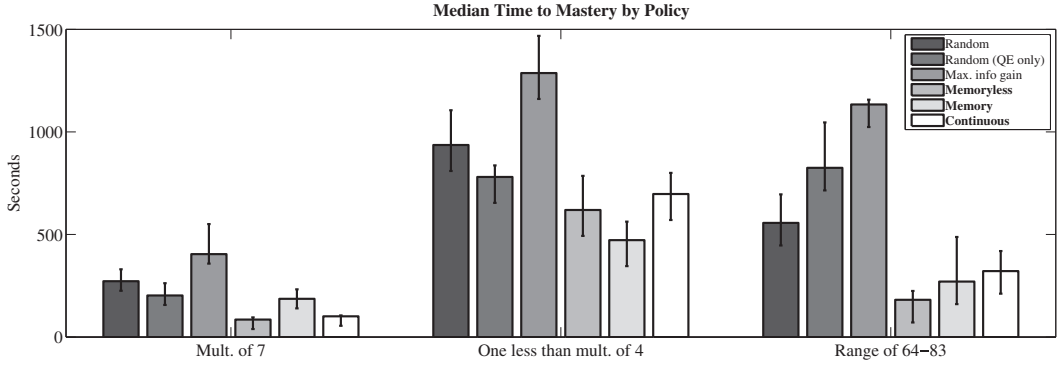

Fig. 10. Median time for participants to learn the concept in the Number Game. Participants taught using one of the POMDP policies (bolded) are significantly faster than participants taught using a *random* policy.

for each concept, similar to the variation shown in Simulation 3. In Experiment 1, we also saw that the different models did not result in significantly different outcomes from one another. Experiment 3 introduces the additional difficulties of a more complex concept space in which we might expect learners to differ more from one another and from our assumed policies. For example, some learners might have little familiarity with modular arithmetic, and thus not consider concepts like "multiples of four minus one." This would make this concept more difficult to learn, and it would likely cause the learner to exhibit different probabilities of particular knowledge state transitions than we assumed. These differences in assumptions about what concepts are possible would also lead to different learning behavior for other concepts, since our learner models assume that the space of possible concepts is known and fixed. Thus, the reason that a single policy is not consistently more effective than others may be due to mismatches between the true learner model and the model assumed by the policy.

This difference is likely also the reason that the *maximum information* policy performed so poorly. The continuous model underlying this policy was likely overconfident and estimated learners as learning more quickly than they actually did. While this may have hurt performance of the *continuous* POMDP policy, it was even more detrimental for the *maximum information* policy, an effect that also appeared when this policy was teaching the memoryless and memory learners in Simulation 3. The *continuous* POMDP policy rarely chose quiz actions, but these actions could never be chosen by the *maximum information* policy. It thus could not revise its beliefs about the learner's state. Additionally, once the model estimated that the participant had learned the concept, it would choose examples essentially at random, since no example was expected to change the learner's state, or focus on only a small subset of actions that were estimated as potentially state changing given the transition ε. Anecdotally, this was frustrating to participants, and it may have led to disengagement. This frustration was exacerbated in the *maximum information* condition compared to the two random conditions since those policies chose half of their numbers from within the concept and half from outside; the *maximum information* policy was more likely to choose numbers outside of the concept when sampling at random as fewer than half of the numbers were in the concept. For the range concept of numbers between 64 and 83, the policy only showed examples near the endpoints in later teaching phases (e.g., 63 and 64), since it estimated that learners would have ruled out all non-range concepts based on the previous examples and there was still a small amount of probability mass on range concepts near the true concept (e.g., numbers between 63 and 83). However, these examples were frequently repeated, again leading to frustration and reflecting the overconfidence of the model.

Problems due to discrepancies between the assumed learner model and human learners are likely to be exacerbated by the cost structure in this experiment. As in Experiment 1, we set the costs of different action types based on the median time it took participants in the control condition to complete each of the action types. In Experiment 1, this resulted in quizzes being less costly than examples; conversely, in this experiment, examples were the least costly action. This resulted in the POMDP policies having relatively few non-example actions: As mentioned above, there were almost no quiz actions for participants

taught using the *continuous model*, and the *discrete model with memory* had the most quiz actions at 6.6%. This means that the *continuous* policy may have been overly confident in its estimate of the learner's state, and that it did not gain information about when that estimate was inaccurate. This issue occurs because the policy assumes that the learner model is accurate; incorporating more uncertainty into the learner models might help to alleviate this problem. Additionally, one could modify the incentive structure to make quiz actions less costly when significant time has elapsed between information-seeking actions.

Examination of the teaching policies can shed light on how different learner models lead to different characteristic actions. All participants began with the same precomputed policy for the first four teaching phases (20 teaching actions). In cases where there was a quiz question, contingent policies were precomputed. A quiz question occurred only for the *memoryless* policy, as the fifteenth action for the concept "multiples of seven." For the multiples of seven concept, most of these beginning example actions showed examples of numbers in the concept (positive examples) for both the *memoryless* and the *discrete model with memory* policies (see Fig. 11). The *continuous* policy showed half positive examples and half negative examples, with the first negative example appearing in the fourth teaching action. One reason for this may be that multiples of seven can be uniquely defined in the concept space using only a few positive examples; thus, for the *continuous* policy and the policy for the *discrete model with memory*, the learner is expected to learn the concept relatively quickly unless she does not update her state.

The concept "multiples of four minus one" was harder for participants to learn than the other concepts. The prior in our model, which was created in previous work

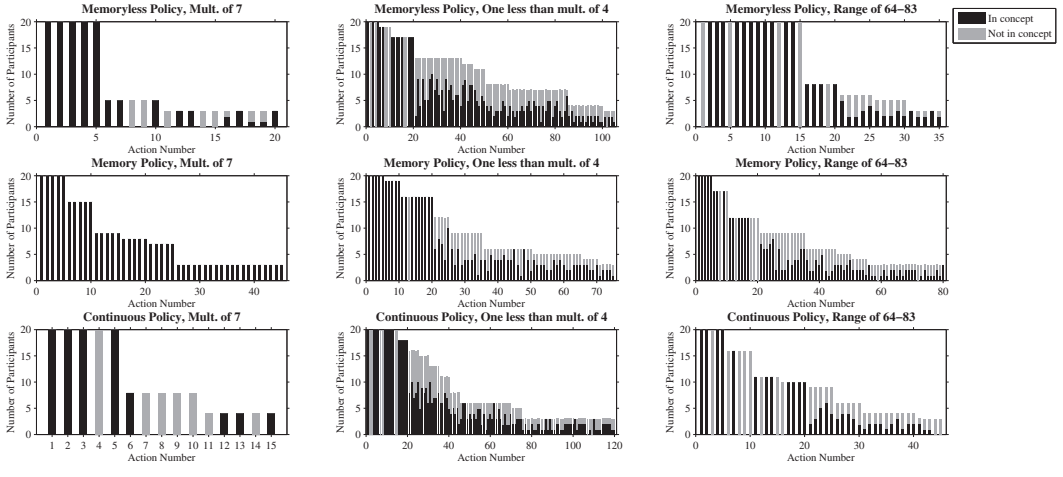

Fig. 11. Number of participants where the action involved a number in or out of the concept at each time in the experiment, by concept and condition. Time points where there are data for at least three participants are shown. The graphs demonstrate that number choices differed both by what learner model was assumed and the concept that was taught.

(Tenenbaum, 2000), predicts this result: This concept has substantially lower prior probability than the other concepts we considered. This concept may also be harder to learn due to the fact that it cannot be defined using only positive examples; all positive examples of this concept are also positive examples of "odd numbers." For all models, the examples shown are mainly odd numbers: In the first twenty examples, few even numbers are shown by both policies for the *continuous* model and *memoryless* model (three and four examples, respectively), and no even numbers are shown by the policy for the *discrete model with memory*. All three policies also show relatively few negative examples that are odd numbers: two for the *memoryless* model, one for the *discrete model with memory*, and three for the *continuous* model. While either of the policies based on discrete learners assumes that the learner could learn the correct concept without negative examples of odd numbers, the continuous learner cannot converge on the correct concept unless some negative examples of odd numbers are shown.

Finally, for the range concept, all three policies again concentrate their actions on positive examples, and qualitatively, all three models look relatively similar. The policies do not seem to focus as much on the endpoints of the range as one might intuitively expect. This may stem from the fact that the actions to choose from are sampled, so in most cases, the tutor may not have the option to show one of the endpoints (or a point just outside the endpoints). This points to the potential advantage of using a smarter sampling strategy for choosing which actions to consider, although the performance of the *maximum information* policy shows the potential downside of this approach when the tutor is too confident about the learner model.

Overall, the results from the Number Game demonstrate that POMDP policies can be effective for teaching a variety of concepts using the same computational framework. The policies that emerged for each model match what one might intuitively expect for teaching these number concepts, and they also highlight how some characteristics of each model manifest in what patterns of choices are optimal. These results demonstrate that POMDPs can be more successful than even sophisticated strategies in the existing literature, such as policies using information gain.

14. General discussion

Teaching effectively is a complex task. It can be difficult to determine what choices will result in learning and to consider the tradeoffs of different pedagogical decisions, especially given that one cannot directly observe the learner's knowledge. We have approached teaching as a decision problem in which a sequence of individual but interdependent pedagogical choices must be made. By framing teaching using the POMDP framework, we take into account the immediate and long-term gains of each possible choice. This framework highlights the modular nature of the different components of the problem, such as the learner model and the pedagogical objective. The POMDP framework allows one to determine an optimal teaching policy with respect to a specified objective.

We have fleshed out how to apply the POMDP framework to teaching concepts and demonstrated the effectiveness of POMDP planning experimentally. This framework has not previously been fully explored for the general problem of teaching at the level of problem selection. We have developed this formulation such that it can be applied to a variety of teaching tasks by specifying the appropriate parameters. Our experiments suggest that POMDPs can sequence information effectively, leading to faster learning than when the ordering of actions is chosen randomly. One of the potential advantages of this framework is that the produced policies automatically consider the utility of selecting instructional actions that aid learning versus diagnostic actions that result in a better estimate of the learner's state. Both of these types of actions occurred in the optimized policies, with their frequency varying based on the learner model and task. This suggests that at least in some cases, it is beneficial to monitor a learner's state and customize teaching based on that state estimate. When monitoring the learner's state, it is often useful to plan several steps into the future; this planning is automatically incorporated with the POMDP formulation. The results of our experiments using the Number Game suggest that planning is beneficial in more complex domains. In principle, monitoring of the state could also be used to terminate teaching when the model has sufficient evidence that the student knows the mapping, rather than using assessment phases. The information necessary to decide whether to terminate can easily be retrieved from the belief state at any given time. The experimental results showed that different learner models result in systematically different policies. This illustrates that optimal problem selection depends not only on knowledge of the domain but also on one's assumptions about the learner.

14.1. Incorporating existing models

One of the advantages of the POMDP framework is that it can be used to explore how assumptions about learning affect optimal teaching policies. Many features of teaching, such as the need to sequence actions and the problem of diagnosing a learner's knowledge, are naturally integrated into POMDP planning, and the modular nature of this framework means that improvements in planning algorithms and improvements in learner models can be developed independently. While feature-based models that assume the student state is observable can also consider the implications of existing student models, additional work may be required to decide what features to use and it may be less natural to specify learner models without appealing to an unobserved student state. In future work, it would be useful to test the POMDP framework within an existing intelligent tutoring system and with student models that have been developed for particular educational domains.

There are many types of existing student models that one could use within the POMDP framework (e.g., Corbett & Anderson, 1995; Corbett & Bhatnagar, 1997; Li, Cohen, Koedinger, & Matsuda, 2011; Pardos, Heffernan, Anderson, & Heffernan, 2010). POMDP planning is likely to work especially well with models that assume students may make incorrect generalizations or that assume the effect of items on learning is contingent upon current skill levels and with domains in which items may involve multiple skills. These

types of models and domains predict that some sequences of actions may be more beneficial than others, and that recognizing the students' level of understanding, including particular misunderstandings, can lead to more effective pedagogical choices. For instance, Contextual Factors Analysis (Pavlik, Yudelson, & Koedinger, 2011) learns how different items contribute to particular skills, some of which may have transfer effects whereby practice on one type of item is beneficial to performance on other types of items. One could learn a student model using this method, and then optimize problem selection for the learned model.

Additionally, using a formal framework like POMDPs can allow designers to determine the impact of changing the learner model on the optimal policy and on the expected time to mastery. Such an examination could allow one to determine which assumptions in the model are most crucial for effective instruction and which have few practical consequences. Lee and Brunskill (2012) examine this question with respect to whether individualized parameters in knowledge tracing would lead to substantially different numbers of practice opportunities for mastery.

The POMDP framework can also be used as a way of comparing existing formulations for selecting teaching actions. Some action selection methods, such as choosing the action with the highest immediate expected utility, are approximations of the POMDP policy. By considering how these methods approximate the POMDP, one can use existing work to assess how close the approximation is to optimal (or whether it is optimal) and what simplifying assumptions are being made. This theoretical framework can thus help to unify existing methods, even in cases where using an optimal policy is impractical.

14.2. Improving learner models

Using POMDPs for teaching relies on having models of learning for a given domain. These models are often hand-created and can be time consuming to construct, although recent research has made progress on constructing learner models from data (Barnes, 2005; Cen, Koedinger, & Junker, 2006; González-Brenes & Mostow, 2012). Even models learned from data, however, are often constrained to a structure that may not be appropriate for all types of tasks. Instead, we need a general approach for constructing probabilistic learner models that can be used within the POMDP framework and for improving these models to reduce mismatches between the learner model and the actual learner.

The approach we have taken in this work is to build on work in cognitive science on probabilistic models of cognition. These probabilistic models have been successful in a variety of areas of cognition (Griffiths, Chater, Kemp, Perfors, & Tenenbaum, 2010; Tenenbaum, Griffiths, & Kemp, 2006), including language (Chater & Manning, 2006; Seidenberg & MacDonald, 1999) and reasoning tasks (Hahn & Oaksford, 2007; Oaksford & Chater, 1994). We considered three learner models based on the literature, fit the parameters of these models using human data, and then used these models within the POMDP framework to derive optimal policies. While this generally resulted in successful learning, the models could clearly still be improved. For example, the models' estimates of the learner's state are currently not accurate enough to use to terminate teaching: the

memoryless model tends to underestimate the probability of mastery, while the continuous model tends to overestimate this probability. The extent to which learner models must be consistent with human learning is thus partially a function of the models' intended use. The models could be further refined by fitting additional parameters from learner data, reducing the number of assumptions about how learning occurs. Many of these probabilistic models have primarily been tested in laboratory settings, so further investigation to improve their fit to human performance should also explore how well these models generalize to more complex academic domains.

Models from the literature also suggest additional issues to consider. For example, Walsh and Goschin (2012) provide a formal proof that a learning agent can learn more from less information if it is aware that it is being taught, rather than simply receiving information from the environment. Shafto, Goodman, and Griffiths (2014) found that human learners are similarly sensitive to whether examples are provided by a teacher or selected randomly. This has implications for what expectations to cultivate in the learner as well as for how to model the learner's knowledge. Since Shafto et al. (2014) present a probabilistic model of learning, the implications of this theory of teaching can be directly determined by computing the optimal policy. Other research is likely to further expand what factors we consider in the learner model and to propose new theories about how people learn. By adopting a strategy in which the models in these theories are starting points that can be parameterized and fit to data, the time to create learner models can be reduced and we can benefit from increasing knowledge about how people learn.

14.3. Model limitations

In our use of POMDPs, we have focused on learner models that represent only the student's knowledge, rather than considering all relevant factors, such as motivation and affect. In principle, POMDPs can use student models that account for these factors, which have received increasing attention in educational technologies (e.g., Conati & Maclaren, 2009; Robison, McQuiggan, & Lester, 2009). However, these factors are not automatically incorporated into the POMDP framework, and they may necessitate a more complex objective function and cost structure. For example, one might want to maximize an objective function that incorporates both motivation and knowledge, while minimizing time; this is likely to make computation of the policy more difficult. One way to incorporate the learners' motivation and goals would be to assume that learners have their own reward functions; for instance, some learners may be more inclined to experiment in a learning environment, while others simply want to complete the activity in minimal time. This would lead to different observation models, and it might lead to different policies for achieving the objective. Inferring the learners' reward functions could be incorporated into the observation model (e.g., for experimental work using this method, Baker, Saxe, & Tenenbaum, 2009; Rafferty, Zaharia, & Griffiths, 2012). Again, this technique would increase the complexity of computing the POMDP policy.

In this work, we have also primarily focused on short-term objectives, directed towards mastery of a small set of relevant material. However, when expanding this framework to larger tasks, teachers may be concerned with longer-term objectives, such as ensuring retention or robust learning. These objectives include maximizing the probability that a student will retain her knowledge over a period of time or that she will be able to transfer her knowledge to new problems. Such objectives imply a learner model in which there must be sets of states in which the learner will show similar responses to certain types of items, such as items of the same type she has already been exposed to. However, when asked about a transfer item, students whose state reflects deeper learning will give different responses than if they had achieved more shallow learning. This could be represented in the cost model by a cost for knowledge states other than those which reflect deep learning. The cost of the shallow knowledge state might be less than that of an arbitrary knowledge state but still non-zero. In this case, there must be some actions that are likely to help the student learn the knowledge robustly, and ideally, there will be actions available to assess the student's generalization abilities. One could further combine the robust learning objective with costs for increased time to mastery, leading to policies that aim to quickly achieve robust learning. In general, richer objectives and learner models can be incorporated through changes to the state space and cost functions. These richer models are likely to have important and interesting consequences for pedagogical action selection and to provide opportunities to explore more complex planning algorithms within the teaching domain.

14.4. Computational limitations

Computational challenges still exist for using POMDP planning: Despite sampling only a fraction of possible actions and using very short horizons, planning took 2–3 s per action. While some strategies could be used to make this delay less apparent to the learner, such as by beginning computations for the next action before the learner submits her work, and the exact length of the delay is dependent on hardware and implementation details, we suspect planning time will also pose a challenge in many other teaching tasks as exact POMDP planning is computationally intractable. In the simulations for alphabetic arithmetic, we did not find a great deal of improvement when increasing the number of actions sampled or the horizon beyond the limits we imposed in the experiments, but the effect of the computational approximation is likely to vary considerably based on the structure of the domain being taught. There are several possibilities for reducing the time to compute the POMDP policy and improving its quality. Following ideas presented in prior POMDP planning algorithms (Ross et al., 2008), we believe that sampling actions based on the particular belief node in the tree would improve the search quality, as would using a more sophisticated evaluation function at the leaves. In particular, the evaluation function does not use the fact that failure on an assessment phase necessitates a complete additional teaching phase; this is relevant whenever assessments occur at fixed intervals. Finally, the relatively long horizons many teaching tasks suggest that these problems may be better served by Monte Carlo Tree Search (MCTS) planning techniques, which have

been very successful at producing good online policies for long horizon planning problems (Gelly & Silver, 2007). All of these approaches may benefit from parallelization.

There are also cases where the computational challenges may be less severe due to the structure of the domain and the type of student model that is assumed. For example, many student models involve sets of independent skills that have either been mastered or not mastered (or more generally, where mastery is unidimensional; for example, Corbett & Anderson, 1995). In any model with unidimensional mastery, the model can be modified to have independent skills. If each problem requires only a limited number of skills, then the change in student knowledge after each problem will be relatively localized, reducing the complexity of updating the belief state. With some action and observation structures, a greedy strategy is optimal, eliminating the need to plan ahead (Karush & Dear, 1967). Yet, in many cases, there are a variety of actions with different tradeoffs in terms of the diagnostic benefit of an action versus the students' likely learning gain (and the type of learning gain; see Koedinger et al. [2012] for discussion of the interaction between types of pedagogical strategies and learning outcomes); in these cases, the myopic strategy of only looking one step ahead is generally suboptimal. Structured models are likely to reduce the computational challenges of planning with POMDPs, but for general teaching tasks, it is likely that computational limitations will have a continuing impact on the quality of POMDP policies. Even in these cases, framing pedagogical action selection within the POMDP framework allows one to benefit from continuing research on solving POMDPs as well as results concerning what approximations result in the best policies. Such an approach may thus be more scalable and easier to maintain than control policies that rely on heuristics or that are created by experts.

14.5. Conclusion

Deciding what pedagogical decisions to make involves reasoning about a number of different components and balancing conflicting priorities. In this paper, we addressed this issue by developing a framework for applying POMDP planning to teaching. This provided a way of conceptualizing how pedagogical action selection should depend on one's model of learning, the structure of the domain, and one's pedagogical objective. We have shown that this framework can be used to select actions in real time in domains of moderate size, and we have demonstrated how despite mismatches between the assumed learner model and actual human learners, POMDP policies can lead to accelerated learning in concept learning tasks, outperforming promising alternatives such as maximum information gain in some domains. While engineering challenges remain, these results suggest that the formal specification of this framework can lead to both theoretical insights and practical improvements in selecting pedagogical actions. Future work should further explore how differences in student models are manifested in optimal teaching policies, especially given our finding that very different assumptions about learning had similar outcomes, as well as consider how to apply this approach within an existing tutoring system.

Acknowledgments

Parts of this work were previously presented at the 15th International Conference on Artificial Intelligence in Education in 2011 (Rafferty, Brunskill, Griffiths, & Shafto, 2011). We thank research assistants Benjamin Shapiro, HyeYoung Shin, Christina Vu, and Julia Ying for their help with the laboratory experiments. This work was funded by an NSF Graduate Fellowship and the Department of Defense NDSEG Fellowship to Anna N. Rafferty, a National Science Foundation Mathematical Sciences Postdoctoral Fellowship to Emma Brunskill, NSF grant number IIS-0845410 to Thomas L. Griffiths, and NSF CAREER award 1149116 to Patrick Shafto.

Notes

1. The objective can be framed as a function to be maximized that considers the reward of various actions, or as a function to be minimized that considers the cost of actions. Any objective in one form can be transformed into the other form by negating all costs or rewards. In the remainder of this paper, we use the cost framing as this is most natural for our experiments.
2. See Supplementary Materials for a list of free parameters related to our use of POMDP planning and information about how these parameters were set.
3. Most state-of-the-art offline algorithms try to compute a policy over a subset of the reachable subspace, but this is still typically a very large region.
4. For simplicity, we distinguish the POMDP policies based on the learner model they assume. For example, the *memoryless* policy is the POMDP policy that assumes a memoryless learner model.
5. We also compared average phases to mastery based on the teaching policy. Results were substantially similar to those based on time. Expected time results are reported because the objective for the planner was based on expected time.
6. We checked whether such a difference existed because participants could potentially use the assessment phases in later mappings to update their hypotheses, if they learned from earlier mappings that teaching continued based on their performance in consecutive assessment phases.

References

Åström, K. (1965). Optimal control of Markov decision processes with incomplete state estimation. *Journal of Mathematical Analysis and Applications*, *10*, 174–205.

Anderson, J. R. (1993). Problem solving and learning. *American Psychologist*, *48*(1), 35.

Atrash, A., & Pineau, J. (2006). Efficient planning and tracking in POMDPs with large observation spaces. In P. Poupart, S. Seneff, J. Williams & S. Young (Eds.), *AAAI-06 workshop on empirical and statistical approaches for spoken dialogue systems* (pp. 7–12). Boston: AAAI Press.

Baker, C. L., Saxe, R. R., & Tenenbaum, J. B. (2009). Action understanding as inverse planning. *Cognition*, *113*(3), 329–349.

Barnes, T. (2005). The Q-matrix method: Mining student response data for knowledge. In J. E. Beck (Ed.), *Proceedings of AAAI 2005 educational data mining workshop* (pp. 39–46). Menlo Park, CA: AAAI Press.

Barnes, T., & Stamper, J. (2008). Toward automatic hint generation for logic proof tutoring using historical student data. In *Intelligent tutoring systems* (pp. 373–382). Berlin: Springer.

Bower, G. H., & Trabasso, T. R. (1964). Concept identification. In R. C. Atkinson (Ed.), *Studies in mathematical psychology* (pp. 32–94). Stanford, CA: Stanford University Press.

Bruner, J. S., Goodnow, J. J., & Austin, G. A. (1956). *A study of thinking*. New York: Wiley.

Brunskill, E., & Russell, S. (2010). RAPID: A reachable anytime planner for imprecisely-sensed domains. In P. Grünwald & P. Spirtes (Eds.), *Proceedings of the 26th conference on uncertainty in artificial intelligence* (pp. 83–92). Corvallis, OR: AUAI Press.

Brunskill, E., Garg, S., Tseng, C., Pal, J., & Findlater, L. (2010). Evaluating an adaptive multi-user educational tool for low-resource regions. In K. Toyama (Ed.), *Proceedings of the 4th ACM/IEEE international conference on information and communication technologies and development*. New York: ACM.

Carpenter, S. K., Pashler, H., Wixted, J. T., & Vul, E. (2008). The effects of tests on learning and forgetting. *Memory & Cognition*, *36*(2), 438–448.

Cassandra, A. (1998). A survey of POMDP applications. Technical Report MCC-INSL-111-98. Microelectronics and Computer Technology Corporation (MCC). Presented at the AAAI Fall Symposium, 1998.

Cen, H., Koedinger, K., & Junker, B. (2006). Learning factors analysis—A general method for cognitive model evaluation and improvement. In *Intelligent tutoring systems* (pp. 164–175). Berlin: Springer.

Chang, K., Beck, J., Mostow, J., & Corbett, A. (2006). A Bayes net toolkit for student modeling in intelligent tutoring systems. In *Intelligent tutoring systems* (pp. 104–113). Berlin: Springer.

Chater, N., & Manning, C. D. (2006). Probabilistic models of language processing and acquisition. *Trends in Cognitive Sciences*, *10*, 335–344.

Chi, M., Jordan, P., VanLehn, K., & Hall, M. (2008). Reinforcement learning-based feature selection for developing pedagogically effective tutorial dialogue tactics. In R. S. Baker et al. (Eds.), *Proceedings of the 1st international conference on educational data mining* (pp. 258–265). Worchester, MA: International Educational Data Mining Society.

Conati, C., & Maclaren, H. (2009). Empirically building and evaluating a probabilistic model of user affect. *User Modeling and User-Adapted Interaction*, *19*(3), 267–303.

Corbett, A. T., & Anderson, J. R. (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, *4*(4), 253–278.

Corbett, A. T., & Bhatnagar, A. (1997). Student modeling in the ACT programming tutor: Adjusting a procedural learning model with declarative knowledge. In A. Jameson, C. Paris, & C. Tasso (Eds.), *Proceedings of the 6th international conference on user modeling* (pp. 243–254). New York: Springer-Verlag.

Doucet, A., de Freitas, N., & Gordon, N. (2001). *Sequential Monte Carlo methods in practice*. New York: Springer.

Feldman, J. (1997). The structure of perceptual categories. *Journal of Mathematical Psychology*, *41*, 145–170.

Folsom-Kovarik, J., Sukthankar, G., Schatz, S., & Nicholson, D. (2010). Scalable POMDPs for diagnosis and planning in intelligent tutoring systems. In F. Meneguzzi & J. Oh (Eds.), *AAAI fall symposium on proactive assistant agents* (pp. 2–7). Menlo Park, CA: AAAI Press.

Gelly, S., & Silver, D. (2007). Combining online and offline learning in UCT. In Z. Ghahramani (Ed.), *Proceedings of the international conference on machine learning (ICML)* (pp. 273–280). New York: ACM.

González-Brenes, J. P., & Mostow, J. (2012). Dynamic cognitive tracing: Towards unified discovery of student and cognitive models. In K. Yasef et al. (Eds.), *Proceedings of the fifth international conference on educational data mining* (pp. 49–56). Worchester, MA: International Educational Data Mining Society.

Griffiths, T. L., Chater, N., Kemp, C., Perfors, A., & Tenenbaum, J. B. (2010). Probabilistic models of cognition: Exploring representations and inductive biases. *Trends in Cognitive Sciences*, *14*(8), 357–364.

Hahn, U., & Oaksford, M. (2007). The rationality of informal argumentation: A Bayesian approach to reasoning fallacies. *Psychological Review*, *114*(3), 704.

Hoey, J., Poupart, P., Boutilier, C., & Mihailidis, A. (2005). POMDP models for assistive technology. In T. Bickmore (Ed.), *Proceedings of the AAAI fall symposium on caring machines: AI in Eldercare* (pp. 51–58). Menlo Park, CA: AAAI Press.

Hu, C., Lovejoy, W., & Shafer, S. (1996). Comparison of some suboptimal control policies in medical drug therapy. *Operations Research*, *44*(5), 696–709.

Kaelbling, L., Littman, M., & Cassandra, A. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, *101*, 99–134.

Karush, W., & Dear, R. (1967). Optimal strategy for item presentation in a learning process. *Management Science*, *13*(11), 773–785.

Koedinger, K. R., Corbett, A. T., & Perfetti, C. (2012). The Knowledge-Learning- Instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive Science*, *36*(5), 757–798.

Kujala, J., Richardson, U., & Lyytinen, H. (2008). A Bayesian-optimal principle for child-friendly adaptation in learning games. *Journal of Mathematical Psychology*, *54*(2), 247–255.

Kurniawati, H., Hsu, D., & Lee, W. (2008). SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In O. Brock et al. (Eds.), Proc. *Robotics: Science and systems* (pp. 65–72). Cambridge, MA: MIT Press.

Lee, J. I., & Brunskill, E. (2012). The impact on individualizing student models on necessary practice opportunities. In K. Yacef et al. (Eds.), *Proceedings of the fifth international conference on educational data mining* (pp. 31–40). Worchester, MA: International Educational Data Mining Society. 118–125.

Levine, M. (1970). Human discrimination learning: The subset-sampling assumption. *Psychological Bulletin*, *74*(6), 397–404.

Li, N., Cohen, W. W., Koedinger, K. R., & Matsuda, N. (2011). A machine learning approach for automatic student model discovery. In M. Pechenizkiy et al. (Eds.), *Proceedings of the Fourth International Conference on Educational Data Mining* (pp. 31–40). Worchester, MA: International Educational Data Mining Society.

MacKay, D. J. C. (1992). Information-based objective functions for active data selection. *Neural Computation*, *4*(4), 590–604.

Monahan, G. (1982). A survey of partially observable Markov decision processes: Theory, models, and algorithms. *Management Science*, *28*, 1–16.

Muldner, K., & Conati, C. (2007). Evaluating a decision-theoretic approach to tailored example selection. In M. M. Veloso (Ed.), *Proceedings of the 20th international joint conference on artificial intelligence* (pp. 483–488). San Francisco: Morgan-Kaufmann.

Murray, R., Vanlehn, K., & Mostow, J. (2004). Looking ahead to select tutorial actions: A decision-theoretic approach. *International Journal of Artificial Intelligence in Education*, *14*(3), 235–278.

Nelson, J., & Movellan, J. (2001). Active inference in concept learning. In T. K. Leen (Eds.), *Advances in neural information processing systems 13* (pp. 45–51). Cambridge, MA: MIT Press.

Nosofsky, R. M. (1998). Optimal performance and exemplar models of classification. In M. Oaksford & N. Chater (Eds.), *Rational models of cognition* (p. 218–247). Oxford, England: Oxford University Press.

Oaksford, M., & Chater, N. (1994). A rational analysis of the selection task as optimal data selection. *Psychological Review*, *101*, 608–631.

Pardos, Z. A., Heffernan, N. T., Anderson, B., & Heffernan, C. L. (2010). Using fine-grained skill models to fit student performance with Bayesian networks. In C. Romero, S. Ventura, M. Pechenizkiy, & R. S. J. d. Baker (Eds.), *Handbook of educational data mining* (pp. 417–426). Boca Raton, FL: CRC Press.

Pavlik, P., Yudelson, M., & Koedinger, K. R. (2011). Using contextual factors analysis to explain transfer of least common multiple skills. In G. Biswas et al. (Eds.), *Proceeding of the 15th international conference on artificial intelligence in education* (pp. 256–263). Heidelberg: Springer.

Pineau, J., Montemerlo, M., Pollack, M., Roy, N., & Thrun, S. (2003). Towards robotic assistants in nursing homes: Challenges and results. *Robotics and Autonomous Systems*, *42*(3), 271–281.

Rafferty, A. N., Brunskill, E., Griffiths, T. L., & Shafto, P. (2011). Faster teaching by POMDP planning. In G. Biswas et al. (Eds.), *Proceedings of the 15th annual international conference on artificial intelligence in education* (pp. 280–287). Heidelberg: Springer.

Rafferty, A. N., Zaharia, M., & Griffiths, T. L. (2012). Optimally designing games for cognitive science research. In N. Miyake et al. (Eds.), *Proceedings of the 34th Annual Conference of the Cognitive Science Society* (pp. 893–898). Austin, TX: Cognitive Science Society.

Restle, F. (1962). The selection of strategies in cue learning. *Psychological Review*, *69*(4), 329–343.

Robison, J., McQuiggan, S., & Lester, J. (2009). Evaluating the consequences of affective feedback in intelligent tutoring systems. In J. Cohn et al. (Eds.), *Affective computing and intelligent interaction and workshops* (pp. 37–42). Washington, D.C.: IEEE.

Ross, S., Chaib-draa, S., & Pineau, J. (2008). Bayesian reinforcement learning in continuous POMDPs with application to robot navigation. In E. Mataric et al. (Eds.), *Proceedings of the international conference on robotics and automation* (pp. 2845–2851). Washington, D.C.: IEEE.

Ross, S., Pineau, J., Paquet, S., & Chaib-draa, B. (2008). Online planning algorithms for POMDPs. *Journal of Artificial Intelligence Research*, *32*(1), 663–704.

Roy, N., Pineau, J., & Thrun, S. (2000). Spoken dialogue management using probabilistic reasoning. In H. Iida (Ed.), *Proceedings of the 38th annual meeting of the association for computational linguistics* (pp. 93–100). Stroudsburg, PA: Association for Computational Linguistics.

Seidenberg, M. S., & MacDonald, M. C. (1999). A probabilistic constraints approach to language acquisition and processing. *Cognitive Science*, *23*(4), 569–588.

Shafto, P., Goodman, N. D., & Griffiths, T. L. (2014). A rational account of pedagogical reasoning: Teaching by, and learning from, examples. *Cognitive Psychology*, *71*, 55–89.

Shannon, C., & Weaver, W. (1948). The mathematical theory of computation. *Bell System Technical Journal*, *27*, 379–423.

Shepard, R. N., Hovland, C. I., & Jenkins, H. M. (1961). Learning and memorization of classifications. *Psychological Monographs*, 75. (13, Whole No. 517).

Sondik, E. J. (1971). The optimal control of partially observable Markov processes (Unpublished doctoral dissertation). Stanford University.

Tang, Y., Young, C., Myung, J., Pitt, M., & Opfer, J. (2010). *Optimal inference and feedback for representational change*. In R. Camtrabone & S. Ohlsson (Eds.), *Proceedings of the 32nd Annual Meeting of the Cognitive Science Society* (pp. 2572–2577). Austin, TX: Cognitive Science Society.

Tenenbaum, J. B. (1999). Bayesian modeling of human concept learning. In M. S. Kearns, S. A. Solla, & D. A. Cohn (Eds.), *Advances in neural information processing systems 11* (pp. 59–65). Cambridge, MA: MIT Press.

Tenenbaum, J. B. (2000). Rules and similarity in concept learning. In S. A. Solla et al. (Eds.), *Advances in neural information processing systems 12* (pp. 59–65). Cambridge, MA: MIT Press.

Tenenbaum, J. B., & Griffiths, T. L. (2001). Generalization, similarity, and Bayesian inference. *Behavioral and Brain Sciences*, *24*, 629–641.

Tenenbaum, J. B., Griffiths, T. L., & Kemp, C. (2006). Theory-based Bayesian models of inductive learning and reasoning. *Trends in Cognitive Sciences*, *10*(7), 309–318.

Theocharous, G., Beckwith, R., Butko, N., & Philipose, M. (2009). Tractable POMDP planning algorithms for optimal teaching in "SPAIS." In C. Geib et al. (Eds.), *Proceedings of the IJCAI workshop on plan, activity, and intent recognition (PAIR)*.

VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, *46*(4), 197–221.

Villano, M. (1992). Probabilistic student models: Bayesian belief networks and knowledge space theory. In *Intelligent tutoring systems* (pp. 491–498). Heidelberg: Springer.

Walsh, T. J., & Goschin, S. (2012). Dynamic teaching in sequential decision making environments. In N. de Freitas & K. Murphy (Eds.), *Proceedings of the 28th conference on uncertainty in artificial intelligence* (pp. 863–872). Corvallis, OR: AUAI Press.

Young, S., Gasic, M., Keizer, S., Mairesse, F., Schatzmann, J., Thomson, B., & Yu, K. (2010). The hidden information state model: A practical framework for POMDP-based spoken dialogue management. *Computer Speech & Language*, *24*(2), 150–174.

Supporting Information

Additional Supporting Information may be found in the online version of this article:

Appendix S1. Additional information.