COGNITIVE SCIENCE
A Multidisciplinary Journal

Cognitive Science 39 (2015) 584–618

ISSN: 0364-0213 print / 1551-6709 online
DOI: 10.1111/cogs.12157

Inferring Learners' Knowledge From Their Actions

Anna N. Rafferty,[a] Michelle M. LaMar,[b] Thomas L. Griffiths[c]

[a]*Computer Science Department, Carleton College*
[b]*Graduate School of Education, University of California, Berkeley*
[c]*Department of Psychology, University of California, Berkeley*

Received 11 March 2013; received in revised form 11 October 2013; accepted 7 January 2014

Abstract

Watching another person take actions to complete a goal and making inferences about that person's knowledge is a relatively natural task for people. This ability can be especially important in educational settings, where the inferences can be used for assessment, diagnosing misconceptions, and providing informative feedback. In this paper, we develop a general framework for automatically making such inferences based on observed actions; this framework is particularly relevant for inferring student knowledge in educational games and other interactive virtual environments. Our approach relies on modeling action planning: We formalize the problem as a Markov decision process in which one must choose what actions to take to complete a goal, where choices will be dependent on one's beliefs about how actions affect the environment. We use a variation of inverse reinforcement learning to infer these beliefs. Through two lab experiments, we show that this model can recover people's beliefs in a simple environment, with accuracy comparable to that of human observers. We then demonstrate that the model can be used to provide real-time feedback and to model data from an existing educational game.

Keywords: Action understanding; Inverse reinforcement learning; Knowledge diagnosis; Bayesian modeling

1. Introduction

People's actions demonstrate a great deal about their understanding of the world. By observing these actions, one can make inferences about this understanding. For instance, based on observing someone take a needlessly long route to get to a particular location, one might infer that the person does not know that road construction has been completed on a shortcut that would take her there more quickly. This inference is an example of

Correspondence should be sent to Anna N. Rafferty, Computer Science Department, Carleton College, Northfield, MN 55057. Email: arafferty@carleton.edu

recognizing someone's misunderstanding through observation alone: The person believes that the road cannot be traversed, but in fact, the road is passable. Making inferences about the understanding and misconceptions of others can be critical in education settings as it allows one to specifically intervene and correct those misconceptions (Davis, Linn, & Clancy, 1995; Liu, Lin, & Kinshuk, 2010). For example, imagine a student playing a biology game. Her responses to specific situations in the game, such as what sequence of actions she takes to adapt an organism to a new environment, can indicate her knowledge about particular elements of cell biology. If she never makes particular adaptations or takes actions in a suboptimal order, this can indicate gaps in her knowledge, leading to targeted remediation; conversely, the student's actions might indicate that she has mastered the current topic and is ready for the next activity. Automating these assessments is beneficial because it does not require interrupting students to explicitly query their knowledge and can provide a detailed picture of students' misconceptions. The benefits of "stealth assessments" that occur within a student's normal activities have been noted by Shute (2011), and prior work has found that such embedded assessments can be useful in the classroom (Feng, Heffernan, & Koedinger, 2009; Razzaq et al., 2005).

The ability to use complex series of actions to automatically diagnose student knowledge is becoming more relevant with the increasing use of games and interactive virtual environments in education. Within these environments, students often perform many individual actions to complete a task, resulting in fine-grained data about the choices that students make. These data contain much more information than simply whether the student completed the task successfully or not, and we would like to use these data to make fine-grained inferences about a student's knowledge, including her misconceptions, just as a teacher could infer this information by observing the student. However, existing assessment models in education are generally not suited to interpreting such sequential process data. These models typically assume the data are conditionally independent given student ability, and consider only success or failure, rather than the way that these outcomes are achieved. Instead, we propose modeling the process by which people choose their actions based on their beliefs. This detailed model then allows us to gain insight into a student's knowledge by observing her actions.

Specifically, we formalize action planning as a Markov decision process (MDP). MDPs are a way of modeling sequences of individual actions taken within a particular environment (Sutton & Barto, 1998). In this model, we characterize a person's knowledge as her beliefs about how her actions affect the state of the world and what states are the most beneficial for achieving her goals. We then propose a framework for automatically inferring these beliefs. This model could be applicable to a variety of action-understanding tasks, but it is particularly relevant to educational settings in which false beliefs (misconceptions) are likely to be common. MDPs allow us to make inferences about students' beliefs by specifying how those beliefs combine with their goals to determine their actions. The result is a decision-theoretic framework for reasoning about sequential actions (Bertsekas & Tsitsiklis, 1996; Ross, 1983; Sutton & Barto, 1998), which has recently been used to model human action planning (Baker, Tenenbaum, & Saxe, 2006; Baker, Saxe & Tenenbaum 2009, 2011; Tauber & Steyvers, 2011; Ullman et al., 2010).

Within an MDP, the transition model encodes a distribution over how the state of the world changes when a particular action is taken. This transition model can be used to represent a student's beliefs, transforming our problem into one of inferring which transition model the student believes is correct by observing the actions that she takes.

A variety of work focuses on understanding the actions of others, ranging from neuroscience to cognitive science to computer science. Work in neuroscience supports the idea that people can recognize the false beliefs of others simply through observing their actions (Grèzes, Frith, & Passingham, 2004). These researchers found that human observers show different activation patterns when observing a person lift a box when that person has correct versus incorrect expectations of the weight of the box. Related work has examined what other inferences about people's mental states can be made through observing actions, finding that relatively accurate inferences about people's goals and confidence can be made even when differences in actions are minute (e.g., Becchio, Manera, Sartori, Cavallo, & Castiello, 2012; Patel, Fleming, & Kilner, 2012). Cognitive science has also approached the question of what inferences people can make about the beliefs of others. Most closely related to our work is Goodman, Baker, and Tenenbaum (2009), which examined people's inferences about the beliefs that another individual has about the consequences of her own actions. In both domains, research has generally focused on isolated actions, rather than the complex sequences of actions that might occur in an educational setting.

In computer science, work on plan recognition has also examined the problem of interpreting others' actions. A common problem in this domain is how to automatically identify someone's intended plan of action based on a set of observed actions (e.g., Gal, Yamangil, Shieber, Rubin, & Grosz, 2008; Kautz & Allen, 1986; Lesh, Rich, & Sidner, 1999). This task has been recognized as potentially helpful in educational environments. Amir and Gal (2011) used a plan recognition framework to categorize sets of individual behaviors in a virtual chemistry lab, such as pouring one beaker into another, as part of larger semantic actions, such as a titration. Our work differs from plan recognition in that we assume that people may have misunderstandings about their actions, rather than assuming that people have full, accurate knowledge of how their actions affect the world.

In this paper, we begin by reviewing MDPs in the context of planning problems. We also introduce inverse reinforcement learning, which involves reasoning from observed actions using the MDP model. We then describe our inverse planning framework, which is a novel modification of inverse reinforcement learning that can be used to infer people's beliefs about the effects of their actions. We next introduce a simple environment that we use for three experiments exploring this framework. In Experiment 1, we show that the inverse planning model can recover learners' beliefs within this environment, and in Experiment 2, we show that the model's inferences are about as accurate as those of human observers. We then examine the practical uses of this model. In Experiment 3, we demonstrate that feedback informed by the model's inferences speeds learning in the planning environment relative to uninformed feedback, and we show that the model can easily be extended to handle a more complex space of possible beliefs that people might

have. Finally, we show that this approach can be applied to data from an existing educational game and use the model to analyze students' behavior within the game.

2. Markov decision processes

When deciding what actions to take to accomplish a goal, people must reason about the immediate value of an action and how that action affects the ease with which the goal can be achieved in the future. MDPs provide a natural framework for such sequential planning problems, in which a series of actions must be taken (see Sutton & Barto, 1998, for an overview). An MDP models an agent's actions over time, in conjunction with the environment in which the agent is acting. The environment is formalized as being in some state s at any given time; we will consider MDPs where the set S of possible states is discrete, although MDPs can be generalized to continuous state spaces. After the action is taken, the environment transitions into a new state based on the action that was chosen as well as the current state. The model that describes what state will occur next given current state s and action a is known as the *transition model* and is represented as a distribution $p(s'|s, a)$ over all possible states s'. The transition model provides a flexible way of specifying how the state of the environment is affected by the agent's actions, allowing for the possibility that the environment may be probabilistic.

Many environments in which agents make choices can be formalized as MDPs. For example, most board games can be represented as MDPs: the configuration of the pieces represents the state of the game, and the player has to choose an action that will affect the configuration, resulting in a transition to a new game state. In some games, such as checkers, the transition model is deterministic: Given a state and action, there is only one possible next state. In other games, such as those where a die is rolled or a card is drawn, the transition model is stochastic: The outcome is not determined completely by the choice of action and the current state, but the probability of each possibility can be calculated. In Fig. 1a, a spaceship navigation game is shown; this is a simplified version of the game used in Experiments 1–3. The player is trying to navigate the spaceship from its current position (s_9) to Earth (s_G); the spaceship cannot go past the edges of the grid, nor can it enter a square with a "hostile alien." Each labeled square in the grid represents a position that the spaceship can occupy. The location of the ship thus corresponds to the state of the game. Actions in the game correspond to presses of one of the colored buttons; at each timestep, the player chooses one of the four buttons to press, or chooses to stop pressing buttons and "land" the ship. The buttons usually move the ship one square in the direction indicated by the arrows in Fig. 1a, but due to small meteors, the ship sometimes moves in another direction instead. If the player tries to move the ship off the grid or into a hostile alien square, the ship remains in its current position. The transition model encodes these next state probabilities given the current ship location and the button pressed. For example, if the player pressed the *teal* button with the ship in its current location of s_9, three states would have non-zero probability: s_4, s_{10}, and s_{12}. s_{10} would be the highest probability next state since *teal* usually moves the ship right. Because it does

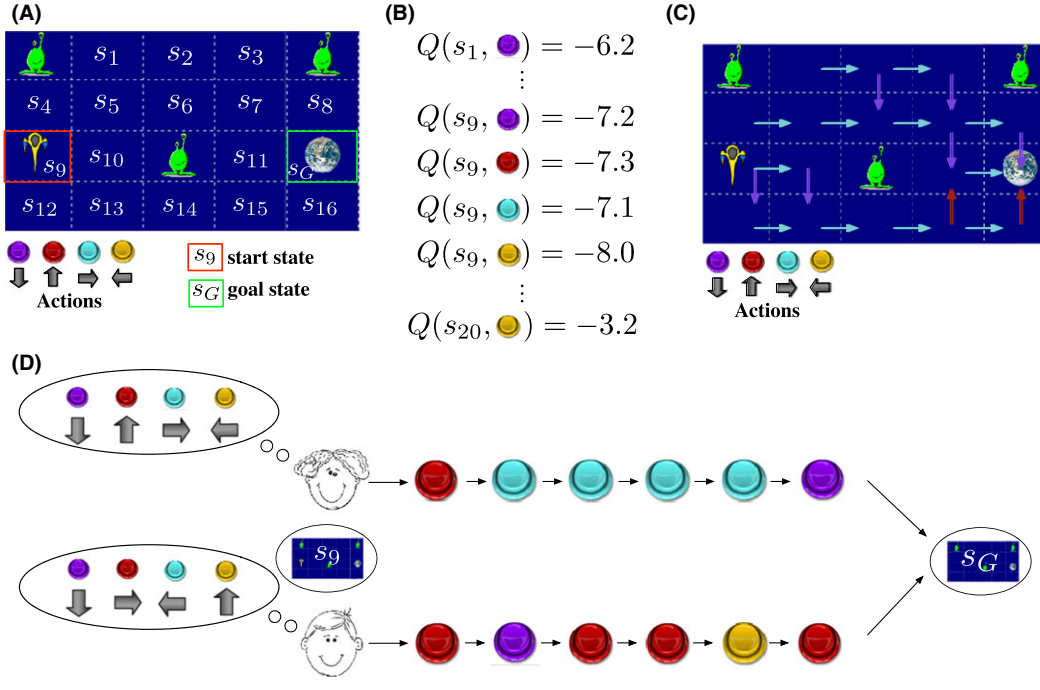

Fig. 1. Modeling a spaceship game using a Markov decision process. (a) States and actions in the game. Each grid square corresponds to a state, and each colored button can be pressed to move the spaceship. Arrows represent the direction that each button usually moves the ship, although movement is noisy. (b) A portion of the Q-function for the game, assuming a Boltzmann policy with low noise. (c) The policy for the game. Arrows are colored to indicate which button has the highest Q-value in each state, with the direction of the arrow indicating the most likely place for the ship to move after that button press. Multiple arrows are shown when the highest Q-values are within 0.05 of one another. (d) Players' beliefs about the buttons lead to different action sequences. Both players use their understanding and their desire to move the ship from the start state (s_9) to the goal state (s_G) to choose their actions. In the model, people's beliefs correspond to different possible transition models T_i.

not matter how the ship reached its current state, the transition process follows the Markov property: The next state is independent of previous states, given the current state.

MDPs encode the reward or incentive structure of an environment in the reward model R: what is the agent trying to achieve (or avoid) through its actions? For any state s, action a, and next state s', $r(s, a, s')$ gives the immediate reward (or cost) of taking action a in state s and the environment transitioning to state s'. When people make choices, they are likely to consider both the immediate consequences of their actions as well as the long-term consequences. The reward model by itself only specifies the immediate rewards or costs: entering a state where one is likely to lose the game in several turns does not incur a cost until the game is actually lost. However, since the specification of the MDP includes both the dynamics of the environment and the reward model, the expected long-term value of taking a particular action in a given state can be calculated. This is computed as the sum of the current reward as well as the expected rewards from future

actions, and is known as the Q-value:

$$Q(s,a) = \sum_{s' \in S} p(s'|a,s) \left(r(s,a,s') + \gamma \sum_{a' \in A} p(a'|s') Q(s',a') \right), \quad (1)$$

where γ represents the relative value of immediate versus future rewards.

The Q-value calculation must take into account how an agent is likely to act in future timesteps. For example, if an agent chooses actions completely randomly, she will likely achieve far fewer rewards than an agent who always chooses the action with the highest expected reward in each state. The *policy* $p(a|s)$ gives the probability that an agent will choose action a while in s for all states $s \in S$ and $a \in A$. For a given state, an agent who acts randomly would have a uniform probability $p(a|s)$ of choosing each action. Given a particular MDP and policy, the Q-function can be calculated using a dynamic program known as value iteration (Bellman, 1957). An optimal policy for an MDP is defined as a policy that maximizes the expected value of the Q-function over all possible states. This optimal policy can also be calculated using value iteration.

In the spaceship game, the player's goal is to land the ship on Earth in as few moves as possible. At each timestep, the player chooses to either press one of the buttons or to land the ship. There is a small cost for each button press and a large cost for landing the ship anywhere other than Earth. After landing, the game terminates, so there are no future costs or rewards. In Fig. 1b, part of the Q-function for the game is shown, where we use a policy that has higher probability of choosing actions with higher Q-values. In state s_9, for instance, the Q-values for buttons that usually move right, up, or down have similar values, while the button that usually moves the ship left has a more negative Q-value: moving left does not help the ship get closer to Earth, and thus has higher long-term costs. Fig. 1c shows the policy that results from the Q-function, with colored arrows representing the best direction to move from each square.

The MDP framework has traditionally been used in planning and decision making. By specifying the components of the MDP and solving for an optimal policy, one can calculate the best action to take in any given state. This approach has a diverse array of applications, from robotics to recommender systems (for an overview, see Feinberg, Shwartz, & Altman, 2002; Puterman, 2005). In education and intelligent tutoring systems, MDPs have been primarily used for making instructional decisions. For example, Barnes and Stamper (2008) used MDPs to choose what hint to give students in a logic tutor. Others, such as Chi, Jordan, VanLehn, and Hall (2008), have used MDP policies to decide what action an automated tutor should take next given the previous interaction of the tutor and student.

3. Inferring learners' belief

By using the MDP framework as a generative model of action planning, we can formally define how a person's beliefs connect to the actions that she chooses. In this

section, we describe the Bayesian inverse planning framework that allows us to infer these beliefs based only on observing a person's actions (see Fig. 1d for an intuitive description applied to the spaceship game described in the previous section). This inverse planning framework relies on the insight that people are likely to choose actions that they think will help them achieve their goals. Thus, their beliefs are likely to be consistent with the chosen actions being better than other possible actions. The inverse planning framework simply formalizes this insight.

Using the MDP framework to infer learners' beliefs follows other recent work in which an agent's behavior can be observed and the goal is to infer some part of an MDP based on that behavior. The most common application is known as *inverse reinforcement learning* and focuses on cases in which one would like to infer an unknown reward model based on observing an agent's behavior (Abbeel & Ng, 2004; Ng & Russell, 2000; Ramachandran & Amir, 2007; Russell, 1998). This technique has also been applied to aspects of social reasoning in psychology, where it is known as inverse planning. For example, previous work has modeled the inferences that people make about other's goals after observing their actions (Baker, Saxe, & Tenenbaum, 2009; Tauber & Steyvers, 2011; Ullman et al., 2010). Our approach shares some similarities with this prior work in inverse planning. However, rather than inferring a person's goal, we assume the goal is known, but the person does not necessarily have accurate beliefs about how her actions will affect her progress toward the goal; we seek to infer these beliefs.

Formalizing the model in terms of the MDP, this means we assume that the reward function R, which encodes the person's goals is known. We also assume the set S of possible configurations of the world is known. People's *hypotheses* about how their actions affect the world then formally correspond to transition models T: Their understanding of how actions affect the current state can be encoded as probabilities $p(s'|a, s)$.

We now want to make inferences about how likely it is that someone has a particular hypothesis given that we have observed a series of actions $\mathbf{a} = (a_1, \ldots, a_n)$ that the person took to try to complete a goal. Given a fixed hypothesis space $\mathcal{T}$ of possible transition models and a given starting state s_1, we want to calculate the posterior distribution over possible hypotheses $T \in \mathcal{T}$:

$$p(T|\mathbf{a}, s_1, R, \gamma) \propto p(\mathbf{a}|s_1, T, R, \gamma)p(T). \tag{2}$$

By calculating a posterior distribution, we can determine both what hypothesis is most probable given the person's actions as well as how strongly the evidence supports this hypothesis over alternatives. Calculating this distribution requires knowing the prior distribution $p(T)$ and computing the likelihood for a particular series of actions given a hypothesis T. The prior distribution over hypotheses accounts for the fact that some beliefs about the effects of different actions may be more likely than others. This prior will vary based on the specific task, and it provides a way for known information about likely misconceptions to be incorporated. For instance, educational research may indicate that certain misunderstandings are common in a particular domain, whereas others are less common. The prior can then be constructed to place higher probability on the

common misunderstandings. In cases where no such information is available, a uniform prior can be used.

To compute the posterior, we must also calculate the likelihood, $p(\mathbf{a}|s_1, T, R, \gamma)$. This quantity corresponds to how likely it is that the person would choose the observed sequence actions given that she believes transitions occur as in model T. Note that having high likelihood does not imply that a sequence of actions is likely to result in the goal state, but only that this sequence is more likely than other sequences to be chosen. As shown in detail in Appendix A, the likelihood can be calculated from the MDP if the person's policy for how she chooses her actions is known. Knowing this policy, or an approximation of it, is necessary to make any inferences about why a person chose a particular set of actions, and it is vital that this policy be dependent on the person's beliefs about T in order for the observations to give information about those beliefs. As in Baker et al. (2009), we assume that people can be modeled as following a noisily optimal policy. This policy, known as a Boltzmann policy, states that actions are chosen as follows:

$$p(a|s, T, R, \gamma) \propto \exp(\beta Q(s, a|T, R, \gamma)), \tag{3}$$

where $Q(s, a|T, R, \gamma)$ is the Q-function defined in the previous section and β is a parameter determining how close the policy is to an optimal policy. Intuitively, this policy corresponds to choosing actions that one believes have higher values than other possible actions. In an optimal policy, agents would choose the action a that maximized $Q(s, a|T, R, \gamma)$. As β becomes large, the Boltzmann policy converges to the optimal policy, while as β goes to 0, the policy converges to choosing actions uniformly at random.

We can now calculate the posterior distribution over the hypothesis space $\mathcal{T}$ by combining the prior and the likelihood. If this space is discrete, the posterior distribution can be calculated exactly by first calculating the $|\mathcal{T}|$ different Q-functions for the MDPs associated with each possible transition model, and then evaluating Equation 6 for each MDP. This is the approach we take for Experiments 1 and 2 and when applying the model to data from an educational game. In other cases, the hypothesis space may be continuous or discrete but very large, making it infeasible to calculate the posterior exactly. An approximate posterior distribution can then be calculated using Markov chain Monte Carlo techniques; we use this approach in Experiment 3.

Consider applying this technique in the spaceship game described in the previous section. Players must choose a sequence of buttons to press to take the ship from its current location to Earth, but now, they may have incorrect beliefs about how each button affects the ship's movement. As shown in Fig. 1d, the starting location of the ship is known and both players are trying to move the ship to s_G, but due to differing beliefs about how the buttons affect the ship, they choose different sequences of actions; in this game, players do not get to see the effect of the button press on the state, so they cannot learn from game play that their beliefs are incorrect. Each chosen sequence is a reasonable solution given the player's beliefs, and thus can be used to make inferences about those beliefs.

This inverse planning model has several advantages. First, it is extremely flexible and can be applied in a variety of situations. Given an appropriate definition of the state

space, many tasks can be specified as MDPs, and the same general framework can be applied to make inferences about people's beliefs based on their actions. For instance, as described in the previous section, most board games can be formalized as MDPs, allowing the framework to be applied to all such games. Additionally, inferences can be made after only a few actions, and multiple sets of observed actions can be used to refine inferences as further evidence accumulates. In an educational environment, this opens the possibility for a tutor to intervene about a specific belief in a timely manner, and to accumulate evidence of misunderstanding over a series of different exercises. The model also provides a fine-grained way of evaluating student responses, rather than only focusing on whether the student was successful in the complete task. Thus, the model can diagnose specific gaps in understanding rather than merely labeling all unsuccessful students as "wrong."

4. Validating the method in the laboratory

We have now defined a general model for inferring people's beliefs about how their actions affect a particular environment. This model makes several assumptions about how people choose actions based on their knowledge, so we first validate the model by testing its accuracy in several lab experiments. For these experiments, participants played a more complicated version of the spaceship game described in the previous sections. Participants used a computerized interface to pilot the spaceship to Earth using as short a path as possible (see Fig. 2a). This version of the game had eight buttons, each of which either

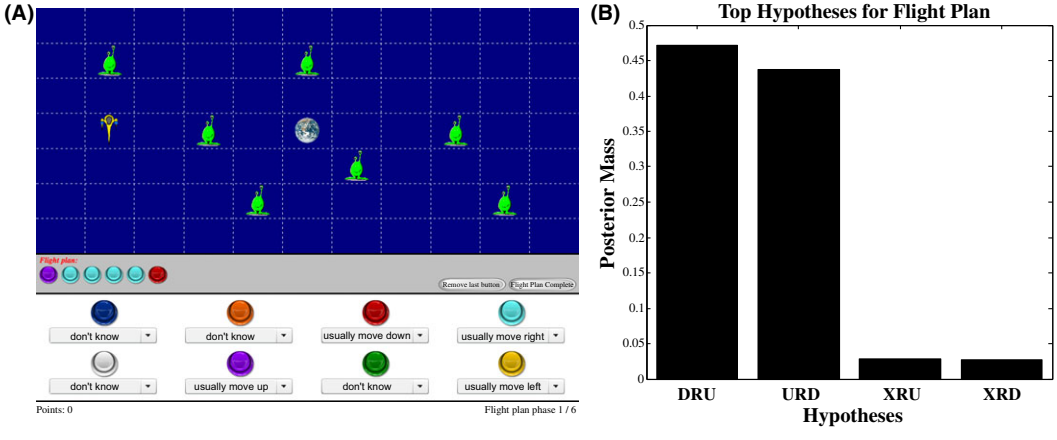

Fig. 2. Diagnosing beliefs about spaceship controls from flight plans. (a) A screen shot from a participant entering a flight plan. The top portion shows the spaceship, Earth, and "hostile aliens." The middle shows the entered flight plan, and the bottom portion shows the buttons as well as menus for participants to indicate how they think each button works. (b) The top hypotheses for the three buttons in the flight plan. The three letter codes indicate, in order, how *purple*, *teal*, and *red* work in the hypothesis: *R* means usually moves the ship right, *D* moves it down, *U* moves it up, and *X* moves it randomly.

usually moved the ship in one direction or moved the ship in a direction at random. Participants were not told exactly how each button affected the ship's movement; instead, they learned this information by observing the effects of the buttons. Using the inverse planning framework, we attempt to infer participants' beliefs about how the buttons work. The interface also included menus for participants to specify their beliefs about each button in order to collect data for validation. While this environment is relatively simple compared to many educational applications, it has the advantage of having many possible misconceptions that are easily articulated. We conduct three experiments in this environment. In Experiment 1, we validate the model by directly comparing its inferences to participants' stated beliefs. To determine how well the model performs compared to human observers, we ask new participants in Experiment 2 to infer the original participants' beliefs using the same information as the model; this allows us to see whether the model makes similar inferences to humans. Experiment 3 uses the model's inferences to guide feedback to participants, and it investigates whether this informed feedback increases the speed of learning.

5. Experiment 1: Comparing the model to participants' beliefs

5.1. Methods

5.1.1. Participants

A total of 25 undergraduates received course credit for their participation.

5.1.2. Stimuli

Participants interacted with the computerized interface shown in Fig. 2a. The top portion of the interface was a 7×11 grid. The spaceship moved around this grid based on the button presses; all other objects remained stationary throughout the experiment. The bottom portion of interface contained the buttons to control the ship's movement and a display field with information about the buttons that were pressed.

5.1.3. Procedure

Participants were told that they would be learning how different buttons affected a spaceship's movements and using that knowledge to pilot the spaceship back to Earth. They were informed that each of the buttons either moved the ship one square in a single direction or in a random direction (due to being broken). Participants were also told that occasionally small meteors caused a button that should move the ship in one direction to move it in another direction, although they could not be sure when a movement was caused by meteors. Meteors caused the spaceship to move in an unexpected direction 15% of the time. Participants were informed that the ship would remain in the same place only if moving would cause it to go off the edge of the grid or to move into a square with a "hostile alien."

Participants alternated between *exploration* phases and *planning* phases. In exploration phases, participants pressed buttons and saw the result of each button press on the

spaceship's location; eight button presses were allowed in each exploration phase. During these phases, the display field informed participants of their last action and its result (e.g., "You pressed the red button, and the spaceship moved up"). In planning phases, the spaceship was relocated to a random square, at least two squares away from Earth, and participants were told to enter a *flight plan*, consisting of a series of button presses, that would take the spaceship from its current square to Earth. The participants could not see the effect of each button immediately, but they had to enter a complete sequence of buttons to guide the ship to Earth. This was to separate learning how the buttons worked from using that knowledge to complete a task. Participants indicated when they were finished, and then were told whether the flight plan had caused the spaceship to reach Earth. The result of the plan was determined by simulating the sequence of button presses using the true transition model for each button, which included the effects of meteors. Each participant completed six exploration and planning phases; the way that each button affected the ship's movement remained the same over these phases.

During all phases, there was a menu below each button containing the possible ways the button could make the ship move ("usually moves left," "usually moves right," "usually moves up," "usually moves down," "moves randomly," or "do not know"). All menus were originally set to "do not know," and participants were told to use the menus to record how they thought the buttons worked. After submitting a flight plan but before they were told the outcome of the flight plan, participants were asked to check that the menus reflected their current beliefs about how each button worked.

5.2. Modeling flight plans

To infer participants' beliefs about the buttons based on their actions in the flight planning phases, we model the flight planning task as an MDP. As in the simplified version of the game, the location of the ship corresponds to the state of the game, and each button is a possible action. There is also a landing action, corresponding to the participant submitting her flight plan: This action has zero reward if chosen when the spaceship is on Earth and a highly negative reward otherwise. All other actions have small negative reward, so shorter sequences of actions are favored over longer sequences. Assuming the magnitude of the negative reward for not reaching Earth is large enough that it is less costly to move to Earth than to stop immediately, the model is insensitive to the exact values used in the reward function, including whether the reward for reaching Earth is zero or a positive value.

For the inverse planning model, we define the hypothesis space $\mathcal{T}$ to match the instructions given to participants: Each button either primarily moves the ship in one direction (left, right, up, or down) or it moves the ship in one of these four directions uniformly at random. In the former case, the button moves the ship in the primary direction 85% of the time and in another direction uniformly at random the other 15% of the time. Each hypothesis consists of the transition models for all eight buttons. We limit hypotheses to those which include buttons that move the ship in each of the four directions, resulting in 166,824 hypotheses. We assume a uniform prior over the hypotheses as there is no reason

to believe people will be biased toward particular hypotheses. If such data were available, the uniform prior could be replaced with a prior that incorporates this information.

Because we do not know a priori how optimal people's plans will be, we allow for uncertainty in the value of the noise parameter β. We marginalize over a discretized set of possible values for β to infer the posterior over $\mathcal{T}$ for each plan, with:

$$p(T|\mathbf{a}, s_1, R) = \sum_{\beta} p(T, \beta|\mathbf{a}, s_1, R), \tag{4}$$

where $p(T, \beta|\mathbf{a}, s_1, R) \propto p(\mathbf{a}|T, \beta, s_1, R)p(T)p(\beta)$. We consider values of β from 0.5 to 5 in increments of 0.5, and place a uniform prior $p(\beta)$ over these values. This procedure allows inference about participants' beliefs to be made without fixing β to any particular value.

5.3. Baseline model

In addition to using the inverse planning model, we evaluated plans using a simple baseline model. In this model, the horizontal and vertical displacements from the spaceship to Earth are calculated and compared to the number of button presses of each color. A button is then matched to a direction if it had the same number of presses as the displacement in that direction. For example, if the flight plan *(blue, blue, red, red, red)* was entered and the spaceship began two squares to the right and three squares down from Earth, the model would predict that the blue button moved the spaceship left and the red button moved it up. This model does not account for obstacles between the spaceship and Earth, and it cannot make predictions about buttons that were not pressed.

5.4. Results and discussion

We ran the model on each flight plan that a participant entered to guide the ship to Earth. Our goal was to infer participant's beliefs at a given phase using data from the flight plan they created at that phase. Multiple phases were completed by each participant, but since we want to determine how well the model can infer beliefs from limited data, we do not seek to model learning or to use information from other flight plans created by the same participant. Initial inspection of the flight plans suggested that some participants may not have understood that each button press could move the ship only one square. For example, a plan with only one button press might be entered when the ship needed to travel a minimum of five squares to reach Earth. We eliminated these plans by having uninformed evaluators examine each of the original flight plans and determine whether there was any way the flight plan could bring the ship to Earth; more detail about the task that these evaluators were completing will be given in Experiment 2. To make clear when we are referring to participants in this experiment and when we are referring to the new evaluators (the participants in Experiment 2), we will refer to participants in this experiment as "planners."

We evaluated the model on the 101 flight plans that at least three of the four evaluators thought were valid plans. Within the model, each hypothesis specifies a full transition model with beliefs about all buttons. However, flight plans included an average of only 2.4 unique buttons, which is insufficient to fully specify the transition model. For example, if the ship started one square up and to the left of Earth, one might infer that buttons not used in the flight plan are less likely to move the ship down or to the right, but there is no information to distinguish among other possible transition models for these buttons. The inverse planning model places the same probability on all such hypotheses. We thus primarily evaluate the model based only on predictions about buttons that were used in the flight plan.

We first examine the *maximum a posteriori* (MAP) estimates of the inverse planning model for each flight plan. This is the mode of the probability distribution: the hypothesis on which the model placed the greatest posterior probability. For example, if a particular hypothesis T_1 had a posterior probability of 0.4 and all other hypotheses had smaller probabilities, the MAP hypothesis would be T_1. For 73% flight plans, the MAP hypothesis matched the planner's stated beliefs about all buttons pressed, and for 93% of plans, the MAP hypothesis matched the stated beliefs for at least some of the buttons in the plan.[1] As shown in Fig. 3a, these results are significantly better than those of the baseline

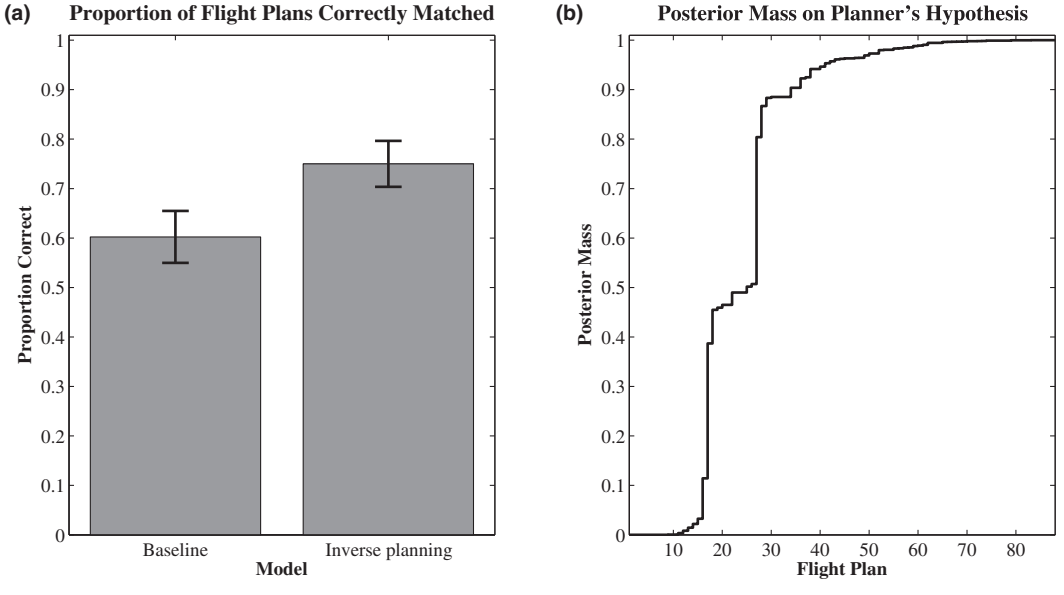

Fig. 3. Model performance on matching the original planner's hypothesis in Experiment 1. (a) Comparison between the baseline model and the inverse planning model. In 60% of flight plans, at least one of the hypotheses given by the baseline model matched the planner's beliefs. In 73% of flight plans, the maximum a posteriori estimate of the inverse planning model matched the planner's beliefs. Each flight plan is one instance of a participant entering a sequence of buttons to guide the ship to Earth. Error bars are equal to 1 *SE*. (b) Posterior mass by flight plan on hypotheses that matched the planner's hypothesis. Flight plans are ordered by the model's performance.

model, which gave a hypothesis that matched the planner's beliefs for only 60% of flight plans (exact McNemar's test, $\chi^2(1) = 8.89$, $p < .01$). These rates are also much higher than a chance baseline, which chooses one of the five options randomly for each button in the plan. This baseline would result in a complete match an average of 5.8% of the time (averaged across plans), with at least one button correct 36% of the time.

The MAP estimates provide one way of evaluating model performance, but the full posterior distribution can give more information about the strength of the model's estimates. Because this distribution gives the probability of each individual hypothesis, we can use it to explore whether the MAP estimate is strongly favored or whether several hypotheses have similar probabilities. This is especially important since some flight plans are inherently ambiguous. Consider the plan *(purple, teal, teal, teal, teal, red)* shown in Fig. 2a, which was entered by a planner during one flight planning phase. Based on the ship's position, this is relatively unambiguous evidence that *teal* usually moves the ship right, but seems equally supportive of *purple* moving the ship up and *red* moving it down as of *red* moving the ship up and *purple* moving it down. Fig. 2b demonstrates that inverse planning model is sensitive to this distinction by showing the probability mass on the four most probable hypotheses, denoted as beliefs about *purple*, *teal*, and *red* moving the ship *right* (*R*), *down* (*D*), *up* (*U*), or *randomly* (*X*). Almost all of the posterior mass is on hypotheses in which *teal* moves the ship right, which is the belief for all four of the most probable hypotheses, but about half of the mass is on each of intuitively plausible possibilities for *red* and *purple*.

To assess how well the posterior distribution matches the planner's self-reported beliefs, we calculate the total posterior mass on hypotheses that are consistent with those beliefs. As shown in Fig. 3b, the model places most of this mass on the correct hypothesis for the majority of the flight plans. The flight plans where the model places about half of the posterior mass on the correct hypothesis tend to be those that were ambiguous. Overall, the model placed an average of 0.74 of the posterior mass on transition models that matched the participant's description of how the buttons worked.

The shape of the posterior distribution can also provide indicators of participant misconceptions. If little posterior mass is placed on the transition model that governed the actual effects of the buttons, this suggests the participant likely has some misunderstanding. This is shown in the data: In cases where the participant's stated beliefs reflect an incorrect understanding, an average of only 0.039 of the posterior mass is placed on hypotheses that correspond to the actual transition model for the buttons. In contrast, 0.82 of the posterior mass is on these hypotheses in cases where the participant is correct.

We can combine the model's diagnosis of correctness with its predictions about participants' beliefs in order to detect misunderstandings. For example, one participant began with the ship in the upper right corner and entered a flight plan with two *teal* buttons, followed by five *red* buttons, and finally a *green* button. The participant correctly understood that *teal* usually moved the ship down, and the model placed 0.97 of the posterior mass on such hypotheses. The participant had misconceptions about the other two buttons, believing that *red* moved the ship left and *green* moved it down. In fact, both buttons moved the ship up. The model's predictions show this misconception: It places 0.99

of the posterior mass on *red* moving the ship left, and 0.95 of the mass on *green* moving it down. In contrast, < 0.01 of the posterior mass is placed on either button moving the ship up. This information could be used to provide customized feedback to the learner, a possibility we explore in Experiment 3.

Overall, the results of Experiment 1 suggest that the inverse planning model is reasonably accurate at inferring planners' beliefs. It outperforms a simple baseline model, and because it outputs a posterior distribution, it provides information about how strongly the evidence supports its inferences and pinpoints participant misunderstandings.

6. Experiment 2: Comparing the model to human inferences

Experiment 1 demonstrated the accuracy of the inverse planning model. However, it is challenging to evaluate how good this accuracy is without a measure of the difficulty of matching planners' beliefs. In general, we would not expect the model to outperform human abilities to infer the beliefs of others, and we know that humans are not always able to make completely accurate inferences from observing someone else's actions. This is clearly the case in the flight planning task: As observed in the plan shown in Fig. 2, there are cases in which multiple hypotheses are equally plausible given the observed actions, and because of the limited number of observations, accurately inferring a full transition model for all buttons is likely to be impossible for either human observers or any given model. In Experiment 2, we asked new participants to evaluate the plans made by the original planners. These observers provide a gold standard for how accurately it is possible to match the planners' beliefs. This experiment also allowed us to identify those plans that could not take the ship to Earth regardless of how the buttons worked.

6.1. Methods

6.1.1. Participants

A total of 24 students received course credit for their participation.

6.1.2. Procedure

Each participant evaluated 25 flight plans, one from each of the planners in the original experiment; all flight plans were thus evaluated by four different participants. The initial instructions about the possible ways the buttons could work were the same as in Experiment 1. Rather than being told that they would be piloting the spaceship, however, participants were told that they would watch the aliens try to fly different spaceships to Earth. For each plan, participants were shown the same display of the spaceship and Earth as in Experiment 1, as well as the series of buttons that was pushed for the plan. Participants were told that this plan had been generated by aliens. Participants were asked to choose one of five options for how they thought that the alien who made the plan had believed the button worked: usually moves left, usually moves right, usually moves up, usually moves down, or moves randomly. Participants were only asked to evaluate the buttons

that were actually used in each plan, and they were told that the way each button worked could change from one plan to another, due to being generated by different aliens in different ships. Three additional questions were asked about each plan. First, whether the plan is likely to take the spaceship back to Earth assuming the buttons work as the participant indicated. Second, whether the plan contains enough button presses to get the spaceship back to Earth. Finally, whether the plan was longer than the shortest plan that could plausibly take the spaceship back to Earth.

6.2. Results and discussion

As in Experiment 1, plans that fewer than three of the four evaluators (participants in Experiment 2) thought were likely to successfully bring the ship to Earth were eliminated. We first examined how well the human evaluators' responses matched the original planners' stated beliefs compared to how well the model matched the planners' beliefs. For each plan, we computed an evaluation accuracy by calculating the proportion of evaluators who gave the same hypothesis as the original planner. The mean evaluation accuracy over all plans was 0.75. The model's accuracy was thus comparable to human performance: It placed an average of 0.74 of the posterior mass on the hypothesis of the planner.

We next evaluated the inverse planning model's ability to capture the evaluators' inferences. For each flight plan, we computed the total posterior mass assigned to all hypotheses that matched at least one hypothesis produced by an evaluator. As shown in Fig. 4a, the model placed an average of 0.87 of the posterior mass on such hypotheses, significantly outperforming the baseline model which gave the same hypothesis as one of the evaluators for 0.71 of the plans (exact McNemar's test, $\chi^2(1) = 20$, $p < .001$). Fig. 4b shows that in most cases, the inverse planning model put almost all of the posterior mass on the evaluators' hypotheses, demonstrating that the model's inferences were similar to those of human evaluators. The inverse planning model was more successful at matching evaluators' hypotheses than the hypotheses of the original planners primarily due to the fact that many plans are ambiguous. For example, given that the spaceship is one square up and to the left of Earth, the plan *(red, yellow)* is equally good evidence that *red* usually moves the ship down and *yellow* usually moves it right as vice versa. With multiple evaluators, each of these hypotheses is likely to be given, but the original planner could only have a single hypothesis. Neither human observers nor the inverse planning model have sufficient information to infer which of the possible hypotheses was actually believed by the original planner.

One discrepancy between the inverse planning model's inferences and human inferences was in the strength of the model's predictions about the buttons that were not used in the plan. People often use one button repeatedly for multiple moves in the same direction, even if there is another button that they believe also moves the ship in the same direction. The model takes each press of the same button as evidence that no other button works the same way, since if there is another such button, it is likely that the person would choose that button at some point. People might instead be using a model that is

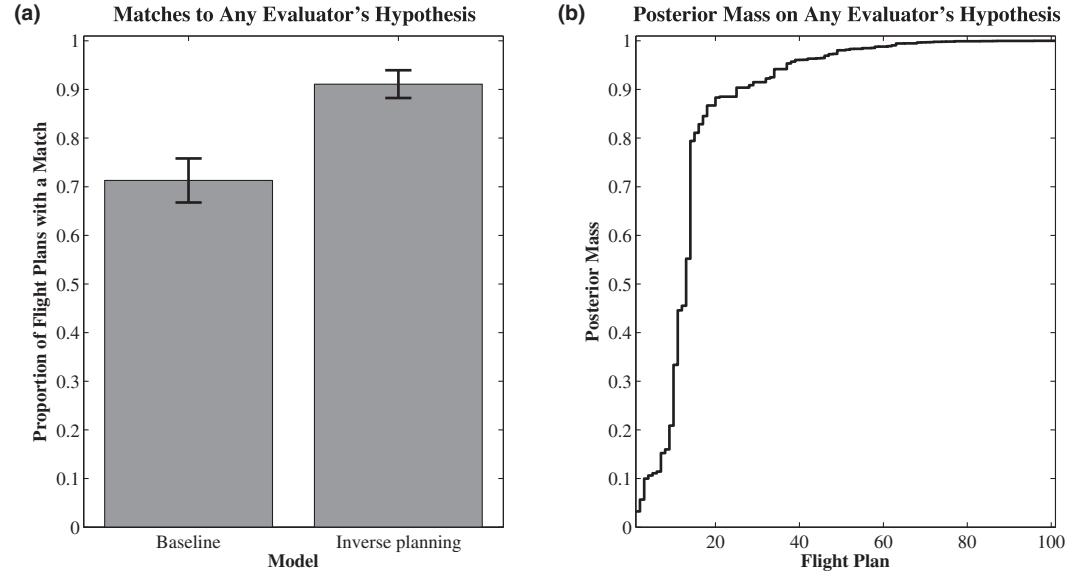

Fig. 4. Model performance on matching evaluators' hypotheses in Experiment 2. (a) The proportion of flight plans for which the baseline model gave a hypothesis matching any of the evaluators' hypotheses versus the proportion of flight plans for which the hypothesis with the highest posterior under the inverse planning model matched any of the evaluators' hypotheses. Error bars are equal to 1 *SE*. (b) Posterior mass by flight plan on a hypothesis that matched any evaluator's hypothesis.

Markov based on the current state of the ship and their last action: choosing to push the same button again is less effort and thus lower cost. While the current inverse planning model does not use such a reward model, it could easily be modified to have a larger state space and more complex reward function to better match the task demands. This flexibility is one of the advantages of the inverse planning model.

7. Experiment 3: Using the model to guide feedback

The previous experiments demonstrate that in the flight planning task, the inverse planning model can diagnose people's beliefs with about the same accuracy as human observers. Given this evidence about the validity of the model's inferences, we now consider how this model might be used in an educational setting. One area where it might be helpful is in guiding automated feedback. Based on the detailed diagnostic information that the model provides, the feedback that is most relevant to a learner can be selected. Previous work has found that more specific feedback can be more helpful than general feedback (Shute, 2008), and that feedback which directly supports learning can have large positive effects (Kluger & DeNisi, 1996).

To test whether feedback provided via the inverse planning model was an improvement over generic feedback, we conducted a new experiment using a modified version of

the flight planning environment. After each flight planning phase, participants were provided with information about how one of the buttons worked. The button about which feedback was given was either selected randomly or via the inverse planning model. If model-based feedback results in participants being able to correctly label how more buttons work after fewer phases of learning, it suggests that the inferences made by the model are sufficiently accurate to guide feedback and that at least in some circumstances, specific diagnostic information can enable practical measures for improving learning.

We also used this experiment to test whether the model could be applied to more complex hypothesis spaces. In the previous experiments, we assumed that for each button, the participant was certain about her belief that it functioned in a particular way. However, this may not always be the case. For example, a participant might believe that the *blue* button either moves the ship randomly or to the left, but be uncertain about which of these two movement patterns is correct. Such hypotheses may be likely to occur when a person has only limited information, and showing that the inverse planning can still diagnose uncertain beliefs is important for real-world applications. In this experiment, we extend the hypothesis space to include such uncertain beliefs and modified the flight planning phases to include multiple flight plans, which provide the opportunity for a participant to show evidence of such uncertainty.

7.1. Methods

7.1.1. Participants

A total of 60 participants were recruited from the University of California, Berkeley, and received course credit for their participation.

7.1.2. Stimuli

Participants used a modified version of the interactive flight planning environment described in Experiment 1. Each participant was randomly assigned to receive feedback based on the inverse planning model or feedback chosen randomly. Thirty participants received each type of feedback.

7.1.3. Procedure

Participants were told that they would be learning how to fly a particular type of alien spaceship. They were informed that they were part of the human resistance movement, and that as a pilot in this movement, they had two tasks: (a) learn how each button affected the movement of the spaceship, and (b) use this knowledge to create flight plans that would take ships back to Earth. The interface gave participants instructions about exploration phases and flight planning phases, just as in Experiment 1, and participants completed each type of phase six times. Exploration phases were identical to Experiment 1, except that participants were allowed six button presses rather than eight. The number of button presses per exploration phase was reduced in order to increase the number of phases required for participants to learn the meaning of all buttons. By increasing the

number of phases over which learning is likely to occur, the potential impact of feedback is increased for both conditions.

The flight planning phases were changed somewhat from Experiment 1. The first change was that there were four flight plans per phase, rather than only one. This was to allow participants the opportunity to show uncertainty about how a particular button worked, consistent with the extension of the model to a more complex hypothesis space.

The second change to the experiment was that some flight plans were for ships that had buttons that could not be used. This modification was to encourage participants to use a variety of different buttons and to learn how all of the eight buttons worked. Since one of our outcome measures was what proportion of buttons a participant had correct beliefs about, we wanted to lessen the probability that participants would learn about only four buttons, one for each direction, and then use only those buttons. The existence of ships with buttons that could not be used by participants was explained within the context of the cover story: Each ship in the phase contained human refugees who had managed to steal the ship from the aliens; in some cases, the aliens sabotaged some of the buttons as they were leaving. Participants entered one flight plan for each of the four ships in each flight planning phase. The starting locations of the ships were chosen as follows. The grid was broken into four quadrants (upper-left, upper-right, lower-left, and lower-right). Within each quadrant, one location was chosen uniformly at random from those squares that were at least two steps from Earth. These four starting locations were then placed in random order, and flight plans were entered sequentially. For each flight plan, participants were told that they were piloting a different ship, which worked the same way as the other ships. To implement the sabotaged buttons, 0–3 buttons were chosen uniformly at random for each ship; these buttons were marked as broken.

After each flight planning phase, participants were asked to check that the drop-down menu below each button showed their current beliefs. Then, they were told how many total points they had earned with their flight plans. In order to motivate participants, we modified the point structure from Experiment 1 such that participants received more points for getting closer to Earth, even if they did not actually reach Earth. Each plan was worth a maximum of 300 points. If the spaceship ended at Earth, the participant received all 300 points, while if the spaceship ended at a location that was further from Earth than it started, the participant received zero points. Otherwise, the participant received a fraction of the maximum points equal to the proportion of the final distance from Earth compared to the starting distance. More specifically, if the ship ended m squares from Earth (measured as the sum of the displacements in the x- and y-directions) and started out n squares from Earth, the participant received $300 \times \frac{n-m}{n}$ points. Participants were informed about how points were calculated and were told the total number of points that they earned from all four flight plans in the phase. This point structure encourages participants to attempt a flight plan even if they are missing crucial knowledge to complete the plan or if all buttons that would give the plan high probability of success are broken. For example, the ship might be up and to the right of Earth. If a participant believes that *red* moves the ship left but that the only button which moves the ship down

is broken, this point structure rewards her for at least beginning the plan by trying to move the ship left.

Once participants had been told how many points they had earned, they were given feedback about one of the buttons. Since the inverse planning model required some time for computation, we computed the model's choice of feedback for participants in both conditions in order to have similar timing of feedback across conditions (see below for details on the algorithm to compute feedback). This computation took 10–15 s, and computation began after all flight plans for the phase were entered. Thus, participants generally had very little waiting time, since feedback computations mainly occurred while they were checking the drop-down menus and learning how many points they had earned. If waiting was required, participants were told that they were waiting for intelligence via an on-screen message. Once the button had been chosen, a message was displayed informing the participant that another member of the resistance had learned that the ⟨*color*⟩ button moved the ship ⟨*direction*⟩; the information that was given to participants was always accurate, regardless of feedback condition. The only difference was that participants in the random condition received feedback about a randomly chosen button and participants in the model-based feedback condition received feedback about a button chosen by the inverse planning model. Participants were asked to change the drop-down menu for that button to reflect the new information and could not continue until they had done so.

7.2. Computing feedback

Feedback was provided six times during the experiment, once after each flight planning phase. Feedback was constrained to be chosen without replacement so that in each phase, participants received feedback for a button about which they had not previously been given feedback. For participants in the random condition, the button was chosen uniformly at random from the remaining buttons.

To compute the feedback for the model-based feedback condition, we needed to calculate a posterior distribution over participant beliefs. As previously noted, we extended the hypothesis space in this experiment to a broader set of possible participant beliefs. The new hypothesis space included cases where a participant has uncertainty about exactly how a button works but places greater confidence on some possibilities than others. This hypothesis space was realized by assuming that participants might act as if they had a belief distribution over possible movement patterns for a button. The hypothesis T is now a collection of distributions $\{\theta^{(1)}, \ldots, \theta^{(8)}\}$. Each $\theta^{(i)}$ is a probability distribution over possible patterns: usually moves the ship left, usually moves the ship right, usually moves the ship up, usually moves the ship down, or moves the ship randomly. For instance, $\theta^{(i)} = [0.5, 0, 0, 0, 0.5]$ would correspond to uncertainty about whether the ith button usually moves the ship left or moves the ship randomly. After each flight plan, we inferred the participant's beliefs about all eight of the buttons; see Appendix for details. Based on the model's inferences, we chose to give feedback about button b^*, the button with the lowest probability that the participant had the correct understanding:

$$b^* = \underset{i}{\operatorname{argmin}}\, \theta^{(i)}_{h_i^*} \tag{5}$$

where h_i^* is the index of the true pattern for button i and $\theta_j^{(i)}$ is the jth element of the vector $\theta^{(i)}$. The minimum is constrained to be over only those buttons that have not previously been chosen, and if multiple buttons have the same minimal $\theta^{(i)}_{h_i^*}$, one of these buttons is chosen uniformly at random. Thus, this computation corresponds to choosing the button about which we believe the participant is most likely to have incorrect beliefs.

7.3. Results and discussion

As shown in Fig. 5, participants who received feedback based on the model were able to correctly identify more buttons at earlier phases than participants who received random feedback. The data were analyzed using a two-way repeated measures ANOVA in which the percentage of correctly identified buttons was predicted by feedback condition (between subjects) and phase number (within subjects), including an interaction term. There was a main effect of condition on the percentage of buttons correct ($F(1,290) = 5.53$, $MSE = 0.516$, $p < .025$). There was also a significant main effect of phase ($F(5,290) = 184.5$, $MSE = 2.32$, $p < .001$), and the interaction between these two effects was significant ($F(5,290) = 3.31$, $MSE = 0.0416$, $p < .01$). Note that as expected, the effect is the largest in the beginning and middle phases; if the number of phases was increased to eight, all participants in both groups would necessarily have all buttons correct as they receive information about one new button at each phase.

We also examined whether participants in the two conditions varied in how likely they were to correctly identify all buttons in a phase. To analyze this, we used GEEQBox, a Matlab toolbox, to perform a logistic regression analysis using generalized estimating equations to correct for repeated measures of the same participants (Ratcliffe & Shults,

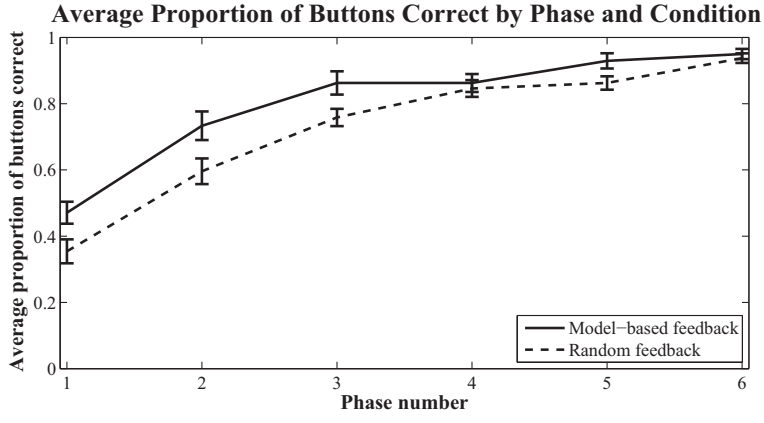

Fig. 5. Average proportion of buttons correct, by phase and condition. Participants who receive model-based feedback have more buttons correct at earlier phases. Error bars are equal to 1 *SE*.

2008). We regressed whether all buttons were correct on condition and phase, with no interaction term, and included a constant predictor in the model. The predictor for condition had value zero for participants in the model-based feedback condition and one for participants in the random feedback condition. An unstructured correlation matrix was assumed for the repeated measures correction. Both coefficients for condition (β_1) and phase (β_2) were significant ($\beta_1 = -0.98$, $p < .01$; $\beta_2 = 0.64$, $p < .001$); these values show that participants in the model-based feedback condition were more likely to have all buttons correct, and participants were more likely to have all buttons correct in later phases.

We hypothesized that the main reason that model-based feedback is more effective is that it is more likely to select buttons about which the participant has an incorrect belief. To test this, we compared how often each method selected such a button. We performed a similar logistic regression analysis as above, but with an outcome variable corresponding to whether the button selected for feedback had already been correctly identified by the participant. We measured participants' choices immediately before giving feedback. For this analysis, we exclude cases where the participant had all buttons correct, as there is no chance of any feedback method choosing a button about which the participant is incorrect. Both coefficients were again significant (condition: $\beta_1 = -1.17$, $p < .001$; phase: $\beta_2 = -0.52$, $p < .001$): Buttons about which the participant was incorrect were more likely to be selected for feedback in the model-based condition, and less likely to be selected in later phases. This supports our hypothesis that the reason model-based feedback results in faster learning is that it more frequently chooses buttons about which the participant has a misconception.

Overall, these results suggest that model-based feedback can speed learning. The model's inferences are accurate enough at recovering participants' beliefs to be informative, even though we know from the first two experiments that these inferences are not always correct. While the technique does require computational resources, the time to choose feedback was not prohibitive and generally took no more than 15 s, with participants rarely waiting for more than a few seconds. In this task, we gave relatively simple feedback. However, in more complex tasks where simply telling the learner about her misunderstanding may be less effective, the model could be used to personalize the examples and exercises that are presented to the learner, focusing on items that highlight the learner's misunderstanding, or to trigger remedial instruction on a particular concept. As Kluger and DeNisi (1996) discuss, there are many factors influencing whether a particular feedback intervention will be helpful to learners; our model provides an additional source of information that can be incorporated to construct effective feedback.

8. Applying the model to an educational game

Experiments 1 and 2 show that the inverse planning model can recover people's beliefs about the effects of their actions, at least within a simple planning environment, and that these inferences are about as accurate as those of human observers. Experiment 3 demonstrated that the inferences could be used to choose feedback to address gaps in a learner's

knowledge, resulting in faster learning than feedback that was not informed by an estimate of the learner's understanding. We now return to the initial motivation for this work: analyzing educational data and making inferences about students' misconceptions. We apply the model to data from a publicly available educational game (Red Hill Studios, 2009). This game provides a more ecologically valid setting than Experiments 1–3, which were conducted in a lab environment. Implementing the model in a real-world setting shows the feasibility of the approach in dealing with common complications such as extraneous actions, a larger state space, and variation in student facility with the game interface. While real-world applications do not provide self-reports of beliefs to which we can compare our inferences, we can instead examine whether the model's inferences are consistent with a more traditional assessment measure that was given after students played the game.

In this game, students learn about cell biology by playing the part of a microbe navigating through increasingly challenging environments. The student's goal in each of the 10 levels is to survive in the new environment, finding and touching the goal post without being eaten or running out of energy. To accomplish this, the student will need to selectively configure her microbe with organelles that enhance either movement or energy production.

At the beginning of the game, the student has a basic microbe with a nucleus and a flagella for movement, but little else. The first few levels of the game are tutorials, introducing the students to the game mechanics which include using the arrow keys to move the microbe around obstacles, running over smaller microbes to eat, and flushing waste products. For each level that the student wins she is awarded either five or ten game tokens. Once they progress past the first few levels they are introduced to the Micro-Mart where they can buy additional features to upgrade their microbe. These features include cilia to increase movement speed (two tokens), mitochondria to increase the energy extracted from food (two tokens each), and chloroplasts to convert sunlight into food (five tokens each). Students may buy multiple mitochondria and chloroplasts, limited only by their token supply. Note that each additional organelle also consumes more energy so having more than needed will decrease the microbe's ability to survive.

In this study, we model student play of the sixth level. As the student starts this level, she is told: "This water drop has nothing but water, carbon dioxide, and light. You'll need to make your own food. Check out the new options at the Mart." She is then taken to the Micro-Mart interface, where chloroplasts are available for purchase for the first time. The student can choose to buy chloroplasts and/or mitochondria or to buy nothing. When she is done with the Micro-Mart, she clicks the "play" button, which places her microbe in a dark section of the tank. To avoid running out of energy, she will need to seek out the patches of light so that her microbe's chloroplasts, if she purchased any, can generate food. The food will not produce enough energy to navigate the dark patches of the tank, however, unless she also has some mitochondria to increase the energy produced by the food. If she successfully navigates the tank and reaches the goal post, she is given 10 game tokens and moves on to the next level. If she fails, she is encouraged to try again, reconfiguring her microbe in the Micro-Mart if she wishes.

While students may consider the "play" phase of the game to be most challenging, as during this phase they are driving their microbe through the environment, avoiding predators and obstacles, it is actually the configuration phase that provides relevant information about student understanding of cellular function. When the students choose which organelles to add to their microbe, they are expressing a belief about what configuration of the microbe will be most successful in the next environment. As the educational content is primarily focused on understanding cellular energy production, the organelles of most interest are the mitochondria and the chloroplasts. For example, in level six, there is sunlight but no food. To survive this level, students will require chloroplasts to generate food from the sunlight, but they will also require mitochondria to generate maximal energy from the food. If students fail to buy either chloroplasts or mitochondria before playing level six, they are expressing a misunderstanding of the functions of these organelles. Traditional analysis of these data would look only at success information, marking a performance "correct" if the student won and "incorrect" if the student lost. By analyzing the sequences of actions the students take, we hope to infer finer-grained beliefs about the importance of both mitochondria and chloroplasts in a low-food but high-sunlight environment.

8.1. Game model

Modeling game play as a MDP requires defining the model elements $\langle S, A, T, R, \gamma \rangle$ and the noise parameter β. The state space S is defined by the configuration of the microbe, which consists of the counts of mitochondria and chloroplasts, each ranging from 0 to 15, giving 256 states. Additionally, there is a goal state *play-success* that is entered when the student wins the level.

The set of student actions A is defined as {*buy-mito*, *buy-chloro*, *play-level*}. We do not model the actions of the student within the tank, as their ability to drive the microbe around obstacles is irrelevant to inferring student understanding of cellular function. Instead, we treat the decision to play the level as an action which may or may not result in success. The two buy actions, meanwhile, directly affect the configuration of the microbe, incrementing the respective counts. The reward model reflects the desirability of achieving a *play-success* state by giving a positive reward of 10 for achieving this state; all other transitions incur a small cost, with the reward set to -0.5. The negative rewards reflect the fact that taking fewer actions to reach *play-success* is preferable to taking more actions.

The transition probabilities are deterministic and fixed for *buy-mito* and *buy-chloro*, as each action moves the student into the game state with the appropriate feature incremented by one. The result of *play-level*, on the other hand, is probabilistic, resulting either in success, in which case the state is changed to *play-success*, or failure, in which case the state does not change. The probability of achieving success is based in part on how well the microbe configuration is suited to the environment with randomness added by how well the student performs driving their microbe. When the microbe is optimally configured for this level by having a moderate amount of both mitochondria and

chloroplasts, then the probability of achieving the play-success state is highest. On the other hand, a microbe that lacks either mitochondria or chloroplasts will have no chance of a successful play. The students, however, may not know this.

Student understanding of the content material should be reflected in their hypotheses about which microbe configurations will maximize their chances of winning the level. As the game states reflect the microbe configuration, the transition probabilities $p(\textit{play-success}|s,a = \textit{play-level})$ over all $s \in S$ represent the student hypotheses. We parameterize the hypothesis space by modeling the probability of *play-success* as a truncated bivariate normal distribution centered on an "ideal" configuration of mitochondria and chloroplasts and truncated at the limits of the state space. Thus, we have 256 different transition models as hypotheses, each one representing a different "ideal" microbe configuration for this level.

Finally, the discount factor γ and the Boltzmann noise parameter β were both set to 0.9. Neither these parameters nor the other assumptions of the model were optimized to produce the best performance; we anticipate that similar settings of the parameters are likely to result in similar predictions.

Estimation of beliefs follows the same procedure as in Experiment 1, except that the calculation of the likelihood is simplified; see Appendix A for details.

8.2. Data

Data came from a pilot study of the educational game conducted in seven schools. Students played the game and then participated in a posttest to provide an external measure of content understanding. The posttest scores were analyzed using a standard Rasch item response theory model (Wilson, 2005), which yields an ability estimate for each student on a continuous logit scale. For level six, we have play records from 218 students, of which 127 have corresponding posttest scores. Play records varied in length from 1 to 22 actions, with a median record length of 4.

8.3. Results and discussion

The play records were analyzed using the inverse planning framework and full posterior probabilities over the hypothesis space were calculated for all students. To infer student beliefs, we first calculated the MAP estimates of the inverse planning model for each student based on his or her record of play. Because each hypothesis is characterized by the numbers of mitochondria and chloroplasts corresponding to the maximum probability of winning the level, we can consider the MAP estimates to be a representation of the student's concept of an "ideal" microbe configuration.

As we expect students with a better understanding of the content material to have more optimal beliefs about the ideal microbe configuration, we can compare the MAP estimates of their beliefs to their posttest ability estimates for evidence of validity. A misunderstanding about how the organelles function should be reflected both in that student's belief about what an ideal microbe configuration for level six would be and in their score

on the posttest. An analysis of variance on the MAP estimates shows that different beliefs in ideal mitochondria and chloroplasts were highly significant predictors for estimated ability scores on the posttest (mitochondria: $F(9,110) = 4.9$, $MSE = 57.2$, $p < .001$; chloroplasts: $F(7,110) = 2.9$, $MSE = 26.3$, $p < .01$). The relationship between average ability estimates of students and their MAP estimated ideal mitochondria and chloroplasts shows clear ability peaks at moderate levels of both features (Fig. 6a). This tracks well with the correct concept that level six requires both mitochondria and chloroplasts, but excessive amounts of either are counter-productive.

While the MAP estimates for student beliefs do provide evidence of validity, this method loses information contained in the full probability distribution. Many students had fairly flat posterior distributions, suggesting that there was insufficient observed data to make strong inferences about their beliefs. Other students had most of their probability mass shared among a few adjacent hypotheses, expressing some uncertainty but indicating a general locality of belief. To get a clearer picture of student thinking, expectations of ideal mitochondria and chloroplasts were taken over the posterior (posterior mean estimates) with the maximum probability used to scale our confidence in the estimate. Fig. 6b uses this information to capture the students' concepts of an ideally configured microbe. We note that student beliefs are distributed over a wider range of values for mitochondria than for chloroplasts. This may be due to the students' increased experience with mitochondria versus chloroplasts as previous game levels required mitochondria for survival.

Educationally, we would like to diagnose significant conceptual misunderstandings. In particular, students may not understand that chloroplasts generate food from sunlight, and

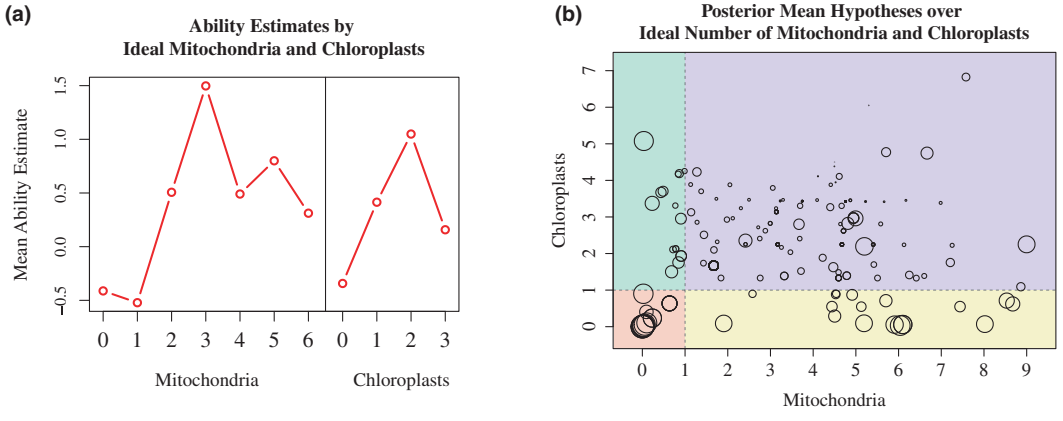

Fig. 6. Diagnosing students' understanding in the microbe game. (a) Mean ability estimates on the posttest as estimated by item response theory versus the maximum a posteriori estimates for ideal number of mitochondria and chloroplasts. Bins with fewer than 10 student estimates are not shown. (b) Posterior mean estimates for student beliefs about mitochondria and chloroplasts. Circle size indicates the magnitude of the highest probability in the hypothesis space, with larger circles reflecting higher confidence in the estimate. Background colors group the space into beliefs that no chloroplasts or mitochondria are needed, one but not the other is needed, or both are needed.

therefore believe that chloroplasts are unnecessary, or students may believe that chloroplasts generate energy directly from sunlight making mitochondria unnecessary. It is also possible that students do not understand the utility of either organelle making both seem unnecessary. By dividing the hypothesis space into coarse sections based on whether or not each feature is needed, we can examine four conceptually distinct groups (shown with different background colors in Fig. 6b). Based on their posterior mean estimates, students were classified as believing both mitochondria and chloroplasts are needed, only mitochondria are needed, only chloroplasts are needed, or neither is needed. As three of these groups represent fairly large misunderstandings, we would expect the posttest scores of these groups to be lower than the correct "both are needed" group. When we analyze the posttest ability estimates by groups, we find, as expected, that students who believed that both chloroplasts and mitochondria were needed demonstrated a greater understanding of the content overall (median posttest ability estimate of 0.33) compared to students who believed only mitochondria or only chloroplasts were needed (-0.36 and -0.54, respectively). Interestingly, students who appeared to believe that neither feature was needed have median ability score of -0.18, higher than the students who believed only one feature was necessary. This suggests that some of these students may be having trouble with the game interface rather than with the educational content. Note that traditional win/loss metrics would lump all of these students together, as none of them would be able to win the game having either zero mitochondria or zero chloroplasts. Thus, the model provides additional diagnostic information that could be used to guide individualized instruction.

9. General discussion

People can make inferences about others' beliefs by observing their actions, and this ability has the potential to be very helpful in educational contexts. We have presented an inverse planning model for automatically making these inferences and shown that it can draw similar conclusions to people when given the same data. Our model relies on formulating action planning as an MDP, in which people may have misconceptions about what transition model is operating in the environment. We then use a variation of inverse reinforcement learning to infer these misconceptions. This model can be used to infer either continuous or discrete hypotheses, making it applicable in a wide variety of situations. We showed that this model can be applicable in educational settings in our final two experiments.

In the remainder of the paper, we discuss two additional issues. First, we consider the possibility of applying the model to a student who is learning or whose knowledge is unstable. Second, we consider the problem of defining an appropriate hypothesis space for a new application.

9.1. Unstable knowledge states

In this paper, we focused on situations in which student knowledge was fixed. While we could have used data from previous phases to infer people's knowledge in later flight

planning phases, we used only data from a single flight plan in order to assess how well the model could make inferences from limited data. However, there are many situations where someone might learn while completing a task or where we might observe the same student over a long period of time, during which her knowledge is presumably changing. One way of handling this issue is to incorporate a probabilistic model of learning, as in Rafferty, Zaharia, and Griffiths (2012). This model would specify how particular experiences (such as a particular state-action-next state tuple) are likely to affect the student's knowledge. Defining such models for a generic domain can be challenging, but student modeling has had many successes and is an active area of research in intelligent tutoring systems (e.g., Chang, Beck, Mostow, & Corbett, 2006; Conati & Muldner, 2007; Corbett & Anderson, 1995).

There has also been work modeling people's action choices in information-seeking tasks, in which a person needs to learn some information in order to be successful. For example, Fu and Pirolli (2007) examined how people navigate the Internet when trying to find some information or learn about some topic. The inverse planning framework can be used to calculate the value of information-seeking actions, although further experiments are needed to determine whether the noisily optimal policy is a good model of these actions or if an explicitly approximate method as in Fu and Gray (2006) is a better fit to human behavior. Overall, exploring the use of the inverse planning model to interpret data from an interactive environment where a student model has been defined is an important extension of the current project. If a detailed model of learning in a domain is not available, a second option would be to include a more generic model of learning such that with some probability the person's knowledge is the same as in the past and with some probability the knowledge has changed. Thus, while we have only considered using the model to assess static knowledge, it is possible to extend its application to cases where learning occurs over the course of the behaviors.

Even in the absence of learning, the knowledge states may appear to change from one task to another. Research into misconceptions in education has shown that some misconceptions are unstable: A student may exhibit the misconception at some times but not at others (Hatano, Amaiwa, & Inagaki, 1996; Hennessy, 1994; Taber, 2000). One reason might be that the situations in which the student exhibits the misconception are sufficiently different from those in which she does not as to make different predictions about what action the student will choose, even given the misconception. This would be represented in our model by having a representation of the state space that differentiated the two situations. However, it may also be that the student's knowledge appears to be truly probabilistic. This can be represented in our model by defining a hypothesis space over probabilistic beliefs, which will often be continuous. For example, in Experiment 3, we represented participants' knowledge using a distribution over possibilities for each button. Thus, half of the time someone might behave as if she believed the button moved the ship up, and half of the time the person might behave as if she believed the button moved the ship randomly. The potential for a continuous hypothesis space allows probabilistic or unstable knowledge states to be easily represented. In order to accurately infer such knowledge states, though, we may require more data, especially if a misconception is relatively improbable.

9.2. Defining the hypothesis space

The flight planning environment was relatively easy to represent as a MDP, and due to the instructions, the possible hypotheses were also reasonably well defined. However, there are many applications where this is not the case, and even within these experiments, we considered two different ways of defining the hypothesis space. There is not a generic way to take a task and output the possible transition structures that correspond to relevant misconceptions. Instead, one must consider what types of misconceptions are of interest and along what axes people are likely to vary in their knowledge in order to construct an appropriate space. Investigating whether this process can be simplified for applying the model to new domains is an important area for future research. One way of simplifying the process could be to try to induce the space of possible hypotheses automatically. This would require access to a large collection of existing data for the domain of interest, with actions from many different people. Such data would provide evidence for what types of knowledge variations occur in the domain.

For some specific domains, existing research has categorized either the space of dimensions on which people's knowledge varies or the specific misconceptions that people exhibit. For example, cognitive tutors cover topics in mathematics and use learner models in which knowledge is divided into a set of possible skills (Corbett & Anderson, 1995; Koedinger, Anderson, Hadley, & Mark, 1997). Such models often require significant time to construct, but recent work has addressed how to discover underlying skills automatically from students' interactions with a tutor or performance on a test (González-Brenes & Mostow, 2013; Waters, Lan, Studer, & Baraniuk, 2012). While these models do not generally identify non-normative rules that a student may believe, hypothesis spaces for our model could posit varying levels of competence in applying each skill. This would make our model applicable to a wide variety of domains where domain models of this form exist.

There have also been previous efforts to characterize specific misconceptions that students may have in particular domains, such as subtraction or algebra (e.g., Brown & VanLehn, 1980; Hatano et al., 1996; Payne & Squibb, 1990; Sleeman, 1984; Stacey & Steinle, 1999). This research identifies the space of misconceptions in these domains and thus could be used to create hypothesis spaces for applying the inverse planning framework to games or problem-solving tasks involving these domains. The inverse planning framework can leverage the previous research exploring both what misconceptions occur as well as how common these misconceptions are.

9.3. Conclusion

We have developed a model for making inferences about people's knowledge based on observing their actions. We take a generative approach in which Markov decisions processes are used to model human action planning, and we use a variation of inverse reinforcement learning to infer people's beliefs about the effects of their actions on the environment. Our model assumes that people are approximately rational actors who

choose actions that they believe will help them to accomplish their goals; the model allows us to infer what beliefs would be necessary for the observed actions to be (approximately) rational. We validated this model in the lab and applied it to providing feedback within a virtual environment and to interpreting data from an existing educational game. The model has the advantage of being flexible enough to be applied to data from a variety of domains and to accumulate evidence over multiple observations. This model also has a number of practical applications in education, where it has the potential to be used to interpret data from the growing number of interactive educational technologies. While there remains important future work in testing the model in additional educational environments, the existing results show its potential to accurately diagnose knowledge without requiring explicit questioning of the student.

Acknowledgments

Parts of this work were previously presented at the 5th International Conference on Educational Data Mining in 2012 (Rafferty, LaMar, & Griffiths, 2012). We thank research assistants Benjamin Shapiro, HyeYoung Shin, and Christina Vu for their help with the laboratory experiments. This work was funded by a Department of Defense NDSEG Fellowship to Anna N. Rafferty, NSF grant number IIS-0845410 to Thomas L. Griffiths, and NSF grant number DRL-0816359 to Robert Hone for Michelle M. LaMar's work.

Note

1. We could only compare model results to the participant's beliefs in 88 of the 101 plans because in 13 of the plans, the planner marked all buttons used in the plan as "do not know."

References

Abbeel, P., & Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. In R. Greiner, and D. Schuurmans (Eds.), *Proceedings of the Twenty-First International Conference on Machine Learning* (pp. 1–8). New York: ACM Press.

Amir, O., & Gal, Y. (2011). Plan recognition in virtual laboratories. In T. Walsh (Ed.), *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 2392–2397). Menlo Park, CA: AAAI Press/International Joint Conferences on Artificial Intelligence.

Baker, C. L., Tenenbaum, J. B., & Saxe, R. R. (2006). Bayesian models of human action understanding. Y. Weiss, B. Schölkopf and J. C. Platt (Eds.), *Advances in neural information processing systems (NIPS)* (pp. 99–106). Cambridge, MA: MIT Press.

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

Baker, C. L., Saxe, R. R., & Tenenbaum, J. B. (2011). Bayesian theory of mind: Modeling joint belief-desire attribution. In L. Carlson, C. Hoelscher, & T. F. Shipley (Eds.), *Proceedings of the 33rd*

Annual Conference of the Cognitive Science Society (pp. 2469–2474). Austin, TX: Cognitive Science Society.

Barnes, T., & Stamper, J. (2008). Toward automatic hint generation for logic proof tutoring using historical student data. In B. P. Woolf et al. (Eds.), *Proceedings of the 9th International Conference on Intelligent Tutoring Systems* (pp. 373–382). Berlin: Springer.

Becchio, C., Manera, V., Sartori, L., Cavallo, A., & Castiello, U. (2012). Grasping intentions: From thought experiments to empirical evidence. *Frontiers in Human Neuroscience*, *6*.

Bellman, R. E. (1957). *Dynamic programming*. Princeton, NJ: Princeton University Press.

Bertsekas, D. P., & Tsitsiklis, J. N. (1996). *Neuro-dynamic programming*. Belmont, MA: Athena Scientific.

Brown, J., & VanLehn, K. (1980). Repair theory: A generative theory of bugs in procedural skills. *Cognitive Science*, *4*(4), 379–426.

Chang, K., Beck, J., Mostow, J., & Corbett, A. (2006). A Bayes net toolkit for student modeling in intelligent tutoring systems. In M. Ikeda et al. Kevin Ashlay, and Tak-Wai Chan (Eds.), *Proceedings of the 8th International Conference on Intelligent Tutoring Systems* (pp. 104–113). Berlin: Springer-Verlag.

Chi, M., Jordan, P., VanLehn, K., & Hall, M. (2008). Reinforcement learning-based feature selection for developing pedagogically effective tutorial dialogue tactics. In pedagogically effective tutorial dialogue tactics (Eds.), *Proceedings of the 1st International Conference on Educational Data Mining* (pp. 258–265). Montreal: Educational Data Mining Society.

Conati, C., & Muldner, K. (2007). Evaluating a decision-theoretic approach to tailored example selection. In R. Sangal et al. (Eds.), *Proceedings of the 20th International Joint Conference on Artificial Intelligence* (pp. 483–488). Menlo Park, CA: AAAI Press.

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

Davis, E. A., Linn, M. C., & Clancy, M. (1995). Learning to use parentheses and quotes in LISP. *Computer Science Education*, *6*(1), 15–31.

Feinberg, E., Shwartz, A., & Altman, E. (2002). *Handbook of Markov decision processes: Methods and applications*. Boston, MA: Kluwer Academic Publishers.

Feng, M., Heffernan, N., & Koedinger, K. R. (2009). Addressing the assessment challenge with an online system that tutors as it assesses. *User Modeling and User-Adapted Interaction*, *19*, 243–266.

Fu, W.-T., & Gray, W. D. (2006). Suboptimal tradeoffs in information seeking. *Cognitive Psychology*, *52*(3), 195–242.

Fu, W.-T., & Pirolli, P. (2007). Snif-act: A cognitive model of user navigation on the world wide web. *Human–Computer Interaction*, *22*(4), 355–412.

Gal, Y., Yamangil, E., Shieber, S., Rubin, A., & Grosz, B. (2008). Towards collaborative intelligent tutors: Automated recognition of users strategies. In B. Woolf, E. Ameur, R. Nkambou, & S. Lajoie (Eds.), *Intelligent tutoring systems* (Vol. 5091, pp. 162–172). Berlin/Heidelberg: Springer.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *6*, 721–741.

González-Brenes, J. P., & Mostow, J. (2013). What and when do students learn? Fully data-driven joint estimation of cognitive and student models. In S. K. D'Mello et al. (Eds.), *Proceedings of the 6th International Conference on Educational Data Mining* (pp. 236–239). Memphis, TN: Educational Data Mining Society.

Goodman, N. D., Baker, C. L., & Tenenbaum, J. B. (2009). Cause and intent: Social reasoning in causal learning. In N. Taatgen, & H. van Rijn (Eds.), *Proceedings of the 31st Annual Conference of the Cognitive Science Society* (pp. 2759–2764). Austin, TX: Cognitive Science Society.

Grèzes, J., Frith, C., & Passingham, R. E. (2004). Inferring false beliefs from the actions of oneself and others: An fmri study. *Neuroimage*, *21*(2), 744–750.

Hatano, G., Amaiwa, S., & Inagaki, K. (1996). "Buggy algorithms" as attractive variants. *The Journal of Mathematical Behavior*, *15*(3), 285–302.

Hennessy, S. (1994). The stability of children's mathematical behavior: When is a bug really a bug? *Learning and Instruction, 3*(4), 315–338.

Kautz, H., & Allen, J. F. (1986). Generalized plan recognition. In T. Kehler et al. (Eds.), *Proceedings of the Fifth National Conference on Artificial Intelligence* (pp. 32–37). Menlo Park, CA: AAAI Press.

Kluger, A., & DeNisi, A. (1996). The effects of feedback interventions on performance: A historical review, a meta-analysis, and a preliminary feedback intervention theory. *Psychological Bulletin*, *119*(2), 254–284.

Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, *8*(1), 30–43.

Lan, A. S., Waters, A. E., Studer, C., & Baraniuk, R. G. (2014). Sparse factor analysis for learning and content analytics. *Journal of Machine Learning Research*, *15*, 1959–2008.

Lesh, N., Rich, C., & Sidner, C. L. (1999). Using plan recognition in human-computer collaboration. In J. Kay (Ed.), *Proceedings of the Seventh International Conference on User Modeling* (pp. 23–32). New York: Springer.

Liu, T.-C., Lin, Y.-C., & Kinshuk (2010). The application of Simulation-Assisted Learning Statistics (SALS) for correcting misconceptions and improving understanding of correlation. *Journal of Computer Assisted Learning*, *26*(2), 143–158.

Ng, A. Y., & Russell, S. (2000). Algorithms for inverse reinforcement learning. In P. Langley (Ed.), *Proceedings of the Seventeenth International Conference on Machine Learning* (pp. 663–670). San Francisco: Morgan Kaufmann.

Patel, D., Fleming, S., & Kilner, J. (2012). Inferring subjective states through the observation of actions. *Proceedings of the Royal Society B: Biological Sciences*, *279*(1748), 4853–4860.

Payne, S., & Squibb, H. (1990). Algebra mal-rules and cognitive accounts of error. *Cognitive Science*, *14*(3), 445–481.

Puterman, M. L. (2005). *Markov decision processes: Discrete stochastic dynamic programming*. Hoboken, NJ: Wiley.

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.

Rafferty, A. N., LaMar, M. M., & Griffiths, T. L. (2012). Inferring learners' knowledge from observed actions. In K. Yacef et al. (Eds.), *Proceedings of the 5th International Conference on Educational Data Mining* (pp. 226–227). Chania, Greece.

Ramachandran, D., & Amir, E. (2007). Bayesian inverse reinforcement learning. In Rajeev Sangal, Harish Mehta, and R. K. Bagga (Eds.), *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 2586–2591). Menlo Park, CA: AAAI Press.

Ratcliffe, S. J., & Shults, J. (2008). GEEQBOX: A MATLAB toolbox for generalized estimating equations and quasi-least squares. *Journal of Statistical Software*, *25*(14), 1–14.

Razzaq, L., Feng, M., Nuzzo-Jones, G., Heffernan, N. T., Koedinger, K. R., Junker, B., . . . Rasmussen, K. (2005). The assistment project: Blending assessment and assisting. In C.K. Looi et al. (Eds.), *Proceedings of the 12th International Conference on Artificial Intelligence in Education* (pp. 555–562). C. K. Looi et al.

Red Hill Studios. (2009). *Lifeboat to Mars*. Available at http://www.pbskids.org/lifeboat. Accessed 5 May 2012.

Ross, S. (1983). *Introduction to stochastic dynamic programming*. San Diego, CA: Academic Press.

Russell, S. (1998). Learning agents for uncertain environments (extended abstract). In P. L. Bartlett, and T. Mansour (Eds.), *Proceedings of the Eleventh Annual Conference on Computational Learning Theory* (pp. 101–103). Madison, WI: ACM Press.

Shute, V. (2008). Focus on formative feedback. *Review of educational research*, *78*(1), 153–189.

Shute, V. (2011). Stealth assessment in computer-based games to support learning. In S. Tobias and J. D. Fletcher (Eds.), *Computer games and instruction* Charlotte, NC: Information Age Publishers.

Sleeman, D. (1984). An attempt to understand students' understanding of basic algebra. *Cognitive Science*, *8* (4), 387–412.

Stacey, K., & Steinle, V. (1999). Understanding decimals: The path to expertise. In J. M. Truran, & K. M. Truran (Eds.), *Making the difference: Proceedings of the 22nd Annual Conference of the Mathematics Education Research Group of Australasia Incorporated* (pp. 446–453). Sydney: MERGA.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning*. Cambridge, MA: MIT Press.

Taber, K. (2000). Multiple frameworks? Evidence of manifold conceptions in individual cognitive structure. *International Journal of Science Education*, *22*(4), 399–417.

Tauber, S., & Steyvers, M. (2011). Using inverse planning and theory of mind for social goal inference. In L. Carlson et al. (Eds.), *Proceedings of the 33rd Annual Conference of the Cognitive Science Society* (pp. 2480–2485). Austin, TX: Cognitive Science Society.

Ullman, T. D., Baker, C. L., Macindoe, O., Evans, O., Goodman, N., & Tenenbaum, J. B. (2010). Help or hinder: Bayesian models of social goal inference. *Advances in Neural Information Processing Systems (NIPS)*, *22*, 1874–1882.

Wilson, M. (2005). *Constructing measures: An item response modeling approach*. Mahwah, NJ: Lawrence Erlbaum.

Appendix A: Calculating the likelihood in the inverse planning model

As described in the section on inferring learners' beliefs, using the inverse planning model to diagnose beliefs requires calculating the likelihood of a set of actions given that the person has a particular set of beliefs about how their actions affect the environment. We want to calculate $p(\mathbf{a}|s_1, T, R, \gamma)$: the probability of taking the sequence of actions a given that the initial state of the environment was s_1. By conditioning only on the initial state of the environment, we are focusing on situations where the person knows this initial state but does not know later states with certainty. This occurs when the outcome of each action cannot be observed; this is the case for the first three experiments we present, but we will modify this assumption for the final application of the model.

MDPs encode certain independence assumptions which we can take advantage of to calculate the likelihood. In particular, future states of an MDP are conditionally independent of past states given the current state; this is known as the Markov property. This means that the likelihood can be calculated using the following recursion:

$$p(\mathbf{a}|s_1, T, R, \gamma) = p(a_1|s_1, T, R, \gamma) \sum_{s' \in S} p(s'|a_1, s_1, T) p(a_2, \ldots, a_n|s', T, R, \gamma), \qquad (6)$$

where $p(a_1|s_1, T, R, \gamma)$ is defined by the person's *policy* for choosing actions given the current state.

In the final application of the model, students see the effect of each action before choosing their next action. To model this, we condition each action choice on the current state. This simplifies the calculation in Equation 6

$$p(\mathbf{a}|s_1, T, R, \gamma) = \prod_{i=1}^{n} p(a_i|s_i, T, R, \gamma). \tag{7}$$

Appendix B: Inferring beliefs in a continuous hypothesis space

In Experiment 3, we assumed that participants might have uncertainty about how each button worked. This was represented as having a transition model $T = \{\theta^{(1)}, \ldots, \theta^{(8)}\}$, where beliefs about the ith button are represented as the distribution $\theta^{(i)}$. Each $\theta^{(i)}$ is a five-dimensional multinomial, with the jth component of the multinomial corresponding to the probability that the button followed the jth movement pattern. The five possible movement patterns were usually moves the ship left, usually moves the ship right, usually moves the ship up, usually moves the ship down, or moves the ship randomly. The hypothesis space is thus continuous: Each hypothesis is a collection of eight vectors, with each vector corresponding to the probability distribution for one button.

Since participants know that each button does operate consistently, although they may not be certain which consistent pattern it follows, we use the following generative model of action planning: At the start of a flight plan f, participants choose one pattern for each button; let $b_f^{(i)}$ be the participant's belief about how button i is working in plan f. The probability of choosing pattern j for button i is $\theta_j^{(i)}$. The participant then chooses the sequence of buttons to guide the ship from its current location to Earth; planning occurs in the same manner as in Experiment 1. This process is repeated for each of the four flight plans in the phase.

Due to the discrete nature of the hypothesis space, the posterior distribution over hypotheses could be calculated exactly for Experiment 1. For Experiment 3, we instead calculate an approximate posterior distribution over the continuous hypothesis space using Gibbs sampling (Geman & Geman, 1984). We sample the variables $b_f^{(i)}$ corresponding to how each button works in each flight plan; at each iteration, we choose an i and f randomly, and sample a new value for $b_f^{(i)}$ given the current values of each other latent variable. To sample this value, we compute:

$$\begin{aligned} p(b_i^{(f)}|b_{-i}^{(1:F)}, b_i^{(-f)}, \mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(F)}) &\propto p(\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(F)}|b_i^{(f)}, b_{-i}^{(1:F)}, b_i^{(-f)}) p(b_i^{(f)}|b_{-i}^{(1:F)}, b_i^{(-f)}) \\ &= p(b_i^{(f)}|b_{-i}^{(1:F)}, b_i^{(-f)}) \prod_{j=1}^{F} p(\mathbf{a}^{(j)}|b_i^{(f)}, b_i^{(-f)}, b_{-i}^{(1:F)}) \\ &= p(b_i^{(f)}|b_i^{(-f)}) \prod_{j=1}^{F} p(\mathbf{a}^{(j)}|b_i^{(j)}, b_{-i}^{(j)}) \\ &\propto p(b_i^{(f)}|b_i^{(-f)}) p(\mathbf{a}^{(f)}|b_i^{(f)}, b_{-i}^{(f)}), \end{aligned} \tag{8}$$

where $\mathbf{a}^{(f)}$ are the observed actions for flight plan n, $b_{-i}^{(f)}$ is the assignment of patterns to buttons other than i for plan f, and $b_i^{(-f)}$) is the assignment of patterns for button i in

plans other than f. We now consider how to sample each of the two terms in the final equation.

To calculate $p(b_i^{(f)}|b_i^{(-f)})$, we place a Dirichlet prior on each θ_i. This then corresponds to a Dirichlet-multinomial model:

$$\begin{aligned} p(b_i^{(f)} = k|b_i^{(-f)}) &= \int_\theta p(b_i^{(f)}|\theta)p(\theta|b_i^{(-f)})d\theta \\ &= \frac{\text{count}(b_i^{(-f)} = k) + \alpha_k}{F - 1 + \sum_j \alpha_j} \end{aligned} \tag{9}$$

where $\boldsymbol{\alpha}$ are the parameters of the prior on θ_i and F is the number of flight plans.

The second term required to sample $b_i^{(f)}$ is the probability $p(\mathbf{a}^{(f)}|b_i^{(f)}, b_{-i}^{(f)})$. To compute the probability, we follow a similar recursive pattern to that in Experiment 1. However, sometimes ships in Experiment 3 had broken buttons. In those cases, we restricted the space of possible actions to those that were available, and calculated the policy given that restricted space.

The above Gibbs sampling procedure allows us to compute a series of samples for how the buttons work in each flight plan. For the feedback, we choose the button b^* that the participant is least likely to know the right pattern for:

$$b^* = \underset{i}{\text{argmin}}\, \bar{\theta}_{h_i^*}^{(i)} \tag{10}$$

where h_i^* is the true movement pattern of button i and $\bar{\theta}$ is our empirical estimate of θ. Buttons about which feedback has already been given are excluded. We calculate the empirical estimate $\bar{\theta}$ from the samples:

$$\bar{\theta}_k = \frac{\sum_{f=1}^{F} \text{count}(b_i^{(f)} = k) + \alpha_k}{\text{num flight plans} + \sum_j \alpha_j}, \tag{11}$$

again relying on the fact that this is a Dirichlet-multinomial model. To calculate feedback in the experiment, we set all $\alpha_j = 1$, corresponding to a uniform prior. We generated 10,100 samples, removing the first 100 samples for burn-in. This produced similar results to using more samples or a longer burn-in period.