Building a More Effective Teaching Robot Using Apprenticeship Learning

MPLab TR 2008-2

Paul Ruvolo, Jacob Whitehill, Marjo Virnes and Javier Movellan
Institute for Neural Computation
University of California San Diego
San Diego, CA 92093-0445
Email: { paul, jake, marjo, movellan } @mplab.ucsd.edu

***Abstract*—What defines good teaching? While attributes such as timing, responsiveness to social cues, and pacing of material clearly play a role, it is difficult to create a comprehensive specification of what it means to be a good teacher. On the other hand, it is relatively easy to obtain *examples* of expert teaching behavior by observing a real teacher. With this inspiration as our guide, we investigated *apprenticeship learning* methods [?] that use data recorded from expert teachers as a means of improving the teaching abilities of RUBI, a social robot immersed in a classroom of 18-24 month old children. While this approach has achieved considerable success in mechanical control, such as automated helicopter flight [?], until now there has been little work on applying it to the field of social robotics. This paper explores two particular approaches to apprenticeship learning, and analyzes the models of teaching that each approach learns from the data of the human teacher. Empirical results indicate that the apprenticeship learning paradigm, though still nascent in its use in the social robotics field, holds promise, and that our proposed methods can already extract meaningful teaching models from demonstrations a from human expert.**

I. Introduction

In the RUBI project at UCSD, we are exploring the potential of using an interactive social robot as a tool for assisting teachers in early childhood education environments [**?**], [**?**], [**?**]. As part of this project, for the last three years we have conducted more than 1000 hours of field studies immersing social robots at UCSD's Early Childhood Education Center, and have identified a wide skill set that RUBI must demonstrate to be an effective teacher. Feedback signals such as recognizing children's facial expressions, analyzing auditory phenomena (for example, detecting cries [**?**]) are important building blocks. Determining the timing of social responses between pupil and teacher actions is also of great significance. What is needed is a method for integrating these components so that RUBI can act effectively in the social environment of a preschool. *Apprenticeship learning* is a potential framework for achieving this integration.

Recent work in the field of apprenticeship learning has shown the power of incorporating demonstrations from human experts in solving difficult control problems. Abbeel and Ng [?], for example, apply apprenticeship learning to the task of automatic helicopter control with impressive results. In fact, the helicopter trained using apprenticeship learning was able to perform complex acrobatic maneuvers such as flips and rolls at a time when the best autonomous helicopter systems were capable of doing little more than hovering in place. Although helicopter control is very different from robot teaching, the two domains also share key similarities: in each domain there is a penalty for failure (the helicopter crashes and is destroyed; the student is taught poorly and becomes turned-off to learning), and it is easier to obtain demonstrations of expert behavior (series of helicopter remote control signals; list of actions taken by the human teacher) than to specify the desired behavior explicitly. With these similarities in mind, we seek to improve RUBI the robot's teaching algorithm using data from an expert preschool teacher.

II. Previous Work

The idea of robots and intelligent agents that learn from people is not new (see [**?**] for an overview of approaches and challenges). In particular, Du Boulay and Luckin [**?**] suggest utilizing findings from pedagogical research to aid in designing a machine teaching agent. This approach has been applied often in the intelligent tutoring systems community (for example, see Burleson and Picard [**?**]). In our work we take a different tact: while much of pedagogical research tends to be theory driven (come up with a hypothesis, do experimental testing to see if it is true), in our work we take a data driven, machine learning approach. In particular we use data from a human expert and apply machine learning techniques to extract patterns and regularities that can be leveraged to create a complete specification of a teaching algorithm. By interpreting the models learned in this framework we may be able to develop new theories about what underlies good teaching.

Apprenticeship learning is a method for solving control problems by incorporating demonstrations from an expert. For example, to create a walking robot one might leverage data captured from 3-D motion sensors of a human walking. Much of the work in this area has focused on having humans operate a device, such as a robotic arm or a helicopter [?], and recording states of the device along with actions that the human performed in these states. These demonstrations provide important constraints about which actions are appropriate for a given state.

Most successful applications of apprenticeship learning have been in domains where there is both an intuitive notion of the state of the system as well as a precise mathematical model of the state dynamics and how they are affected by the various control signals at our disposal. For example, in the case of training a helicopter to perform acrobatic maneuvers autonomously [?], the helicopter's state can be described by its orientation, angular velocities, acceleration, and position relative to some fixed reference point. In the case of training a robot to teach children in a classroom, on the other hand, the state could consist of any number of cognitive and emotional components of the child's mind. How to represent the most important state compactly is an open question.

Further complicating matters is the notion of dynamics: In the helicopter case, classical physics gives precise constraints on the dynamics of the system. After learning certain parameter values empirically [**?**], an accurate model of the world dynamics can be learned. In contrast, in the social interaction setting of a pre-school classroom, there is no such exact science underlying its dynamics. While it is possible that apprenticeship learning does not work in social interaction domains due to the lack of a rigorous description of state and dynamics, it is also conceivable that such "ill-defined" domains can profit the most from expert demonstrations.

III. Learning to Teach

In this study, we explore two approaches to using apprenticeship learning to improve RUBI's teaching capabilities. The teaching setting is the "Name the Object Game" in which RUBI displays four different objects on her tablet PC (see figure 2) and asks the child to touch a specific object using an auditory prompt (e.g. "Show me the apple"). RUBI is equipped with a simple, deterministic teaching module that periodically reminds the child, at a fixed frequency, which object to touch. This simple "teaching" strategy was programmed by hand and can clearly be improved upon greatly. The frequency of the prompts could, using data from an expert human teacher, be made more "human-like"; auditory or visual hints regarding which fruit to select could be given to the child as necessary; and positive feedback could be issued to certain children upon their selecting the correct fruit object. In this study, we are interesting in learning automatically when to perform such teaching actions by observing expert human teachers. In this sense, we are interested in *learning to teach*.

In this work, instead of the human expert pre-school teacher giving examples of actions she would perform *alone* (as in the standard apprenticeship learning formulation), the teacher provides expert demonstrations by *augmenting* the pre-programmed behavior of RUBI. RUBI, the expert human teacher, and the pre-school pupil thus form a *teaching triad* (see Figure 1). Using these collected data of supplementary actions we will apply our apprenticeship learning approaches (defined in Section V) to learn an improved teaching policy. Two key assumptions are implicit in this formulation. The first is that the actions of RUBI will have an equivalent effect as the human teacher performing that same action. The second is that the human teacher acts because the robot has not performed an appropriate action. This second assumption implies that we know that RUBI is teaching well if the teacher is not intervening in the teaching session. Iteratively recording "apprenticeship" data from the teaching triads and re-training RUBI's teaching algorithm would hopefully converge so that the teacher would not feel the need to act at all to supplement RUBI. One disadvantage of this model is that there is no way for the human to tell RUBI that she is acting too much. While future research may address this problem, for the study presented in this paper we thought it best to allow the pre-school teacher to focus on teaching the children and not on actively "correcting the robot."

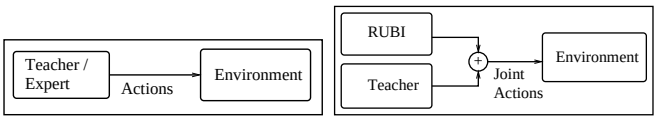

Fig. 1. Left: the conventional apprenticeship learning setting. Right: the teaching triad we use for our approach.

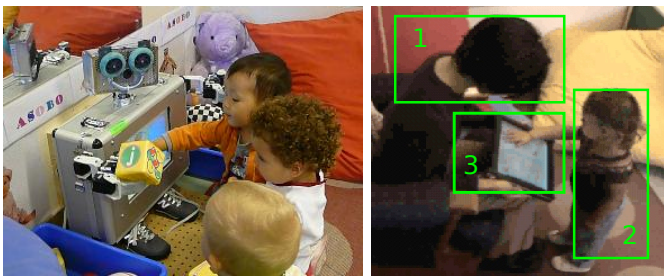

Fig. 2. Left: RUBI teaching children at ECEC. **Right:** A teaching triad consisting of a teacher from ECEC (1), a student (2), and a stripped-down version of RUBI consisting of a touch-screen tablet PC (3). Data from these interactions was coded by humans into 9 behavioral channels as shown in Figure 3

A. Dataset of teaching demonstrations

In order to gather data to inform RUBI's teaching, we recorded information from teaching triads (see Figure 2). These interactions take place between a child, a preschool teacher, and a stripped-down version of RUBI consisting of only her touch-screen tablet PC. The teacher positions the touchscreen on her belly, thus approximating the teaching setting used by RUBI, which has a touchscreen on her belly. In these sessions, the child was playing the "Name the Object Game". Only one child was allowed to interact with the teacher at any given time.

A total of 8 preschool students participated in this study. Each child interacted one on one with the teacher for an average of XX minutes. The teaching sessions were contiguous with one teaching session per child. The coding of the teaching session into the set of features shown in Figure 3 was performed by an external human observer situated behind a one-way mirror.

1) Teacher repeats the computer sound i.e. the name of the object (e.g. “apple”)
2) Teacher asks a question, e.g. “Can you show me the apple?” / “Where is the apple?”
3) Teacher gives a hint, e.g. pointing the correct object, asking other objects before the correct object
4) Teacher gives child feedback, e.g. saying the “good job” or repeating the name of the object after a correct answer
5) Child touches the right object
6) Child touches the wrong object
7) Child touches the screen after giving the right answer
8) Child is far away out the reach of the computer
9) RUBI says the name of the object (e.g. “apple”)

Fig. 3: The nine actions that are recorded by a coder that observed interactions between RUBI, a teacher, and a student. Each of these actions is coded at 1 second granularity

IV. Mathematical Framework

The typical formalism for apprenticeship learning problem is the Markov Decision Process (MDP). Let $\Pi(Q)$ denote the set of all probability distributions over the set Q. An MDP is a tuple $(S, A, P, X_0, R, \gamma)$ where S is a set of states; A is a set of actions; $P : S \times A \rightarrow \Pi(S)$ describes the transition dynamics; $X_0 : \Pi(S)$ is a distribution over the initial state; $R : S \times A \rightarrow \Re$ is the reward, i.e., a notion of the desirability of performing a particular action in a particular state; and $\gamma \in [0, 1)$ is a discount factor that specifies how much to weight immediate versus future rewards.

A *policy* in the MDP setting is a mapping $\pi : S \rightarrow A$ from a given state to a particular action. The goal in the Markov Decision Process setting is to find a policy, $\pi^\star$, that maximizes some notion of desirability, e.g., the expected long-run discounted sum of rewards:

$$\pi^\star \quad = \quad \arg\max_{\pi} E_{s_0 \sim X_0} \left[\sum_{t=0}^{t=\infty} \gamma^t R(s_t, a_t) | \pi \right] \qquad (1)$$

Assuming complete knowledge of the parameters of an MDP there are numerous techniques available, one example being policy iteration, for computing $\pi^\star$ [**?**]. However, when one or more of the parameters of the MDP (e.g., the transition probabilities or reward function) is unknown, apprenticeship learning methods can prove to be useful tools.

We can characterize demonstrations from a human teacher as a series of action-state pairs. Sequences of these pairs are called *trajectories*. Let $u_i = (s_1, a_1), (s_2, a_2) \ldots (s_{T_i}, a_{T_i})$ denote the ith trajectory. We use the symbol $U = (u_1 \ldots u_m)$ to refer to set of m trajectories demonstrated by the expert.

Assuming that S and A (the states and actions of our MDP) are known, but that P and R (the transition dynamics and reward function) are unknown, how can the expert trajectories U be used to find a policy π that maximizes Equation 1? We explore two methods of achieving this goal, which we call the *direct approach* and the *indirect approach*.

The direct approach ignores P and R altogether and attempts instead to use the expert's state-action trajectories U as training data to a supervised learning algorithm. Thus, an explicit mapping from states to actions is constructed. This approach can be described as “mimic the expert.” The direct approach has been deployed widely in the field of robotics [**?**], [**?**]. The idea is that we assume the expert's behavior is approximately optimal; therefore, by mimicking his or her behavior, the policy that approximately maximizes Equation 1 can be computed.

The indirect approach attempts to construct a model of the transition dynamics, P, and reward structure, R, of the MDP. Once these parameters have been estimated, standard reinforcement learning techniques can be applied to obtain the optimal policy $\pi^\star$ [?]. The indirect approach benefits by learning about the relationship between state and reward. This approach may exhibit better generalization when the trajectories from the expert cover only a small portion of the state space. There are various algorithms for learning a policy in the indirect setting. Abbeel and Ng provide a framework that can guarantee performance similar to that of the expert under certain assumptions about the reward function [?]. Other approaches estimate R by choosing the reward function that makes the actions of the expert appear as good as possible w.r.t. R [**?**], [**?**]. In Section V-B we present our own algorithm for indirect apprenticeship learning which may be more appropriate for the “teaching triad” context in which expert trajectories are captured in tandem with an existing automatic teaching system.

V. Methods

In applying both the direct and indirect apprenticeship learning approaches, we have to be careful in our definitions of states and actions to account for the fact that the data from the experts was obtained in conjunction with RUBI's existing teaching system. Our specific formulations of each approach is discussed in this section.

We can use the data collected from the teaching demonstrations to define the state and action parameters of a Markov Decision Process. At each time step there are five possible actions that can be performed by either the teacher or RUBI. These actions are: Repeat the name of the object; Ask a question; Give a hint; Give feedback; and Do nothing. These five actions comprise the set of actions, A, for an MDP. The crucial difference between the direct and indirect approach is both whether or not they learn a model of the dynamics of the world and the specific manner in which they define state.

A. Direct Approach

In the direct approach to apprenticeship learning, a mapping from states to actions is created using supervised learning techniques on data collected from expert demonstrations. The action space for this approach is defined as in Section V. How to define the state space is far less clear – which relevant variables exist in the world, and which of these could reasonably be captured by a robot, in particular RUBI? In the present study, we focus on learning the *timing* of teaching events – e.g., how often to tepeat the name of the fruit, how

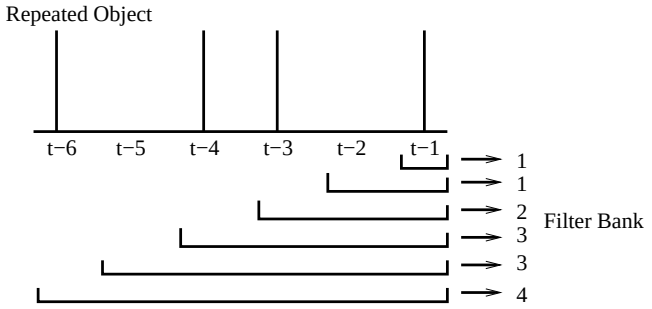

Fig. 4: *Top*: a stem plot of a the binary time series of the feature of whether the object name has been repeated. *Bottom*: the various temporal kernels that are extracted from this times series. These temporal kernels are used as input to supervised learning methods for predicting the next action.

long the child's last correct answer to offer a hint, etc. To enable such temporal relationships to be learned, we define the state space using the *history* of the recent actions of the child, and the recent joint actions of the teacher and the robot. We convert the recorded actions and observation histories into a series of features by applying temporal kernels of various lengths (e.g., how many times did the child get the right answer in the last 1 second, 2 seconds, 3 seconds, 4 seconds, etc.). The set of observations that RUBI makes of the child, either through her touchscreen or through her proximity sensor, is given by items 5 through 8 in Figure 3. Each of the child's four possible actions (excluding the action of doing nothing) forms a feature channel as well. We extract features from the history using kernels of sixteen different temporal scales. Each scale is one scale larger than the proceeding scale giving us temporal kernels of 1 second, 2 seconds, 3 seconds, . . ., and 16 seconds (see Figure 4). The combination of 16 temporal scales, 4 observations, 4 actions gives us a total of $16\times4+16\times4=128$ features to represent the state of the teaching session.

Given a definition of states and actions, we can apply standard supervised learning methods to learn a policy. The approach we employ in this work is multinomial ridge logistic regression [**?**]. Logistic regression outputs a matrix of weight values that express the log-probabilitys of choosing a particular action given the input features (history of actions and observations). These probabilities can be readiliy interpreted; we analyze the model of teaching that the direct approach learns, in Section VI-B.

B. Indirect Approach

The indirect approach is a two-stage process. In the first stage we learn a model of how the actions of a teaching agent affect the world (the states, and state transition probabilities). In the second stage we assign a notion of desirability (reward function) to each state in our system and then use reinforcement learning techniques to compute the optimal policy.

In order to learn both the state S and transition dynamics P of a teaching interaction, we use a Hidden Markov Model (HMM). HMMs use maximum likelihood estimation to derive an internal notion of states of the world that are responsible for generating a sequence of observations. In this work, the observations are features of the teaching interaction, such as whether or not the child pressed the right answer. In our case we used an extension of the standard HMM where the transition matrices are parameterized by an action. This allows us to learn a different set of dynamics for each possible action that the teacher can perform. The action space is defined identically to that in the direct approach. The observation space consists of all of the child's actions in Figure 3 (items 5 through 8).

After learning S and P using an HMM, we can convert the data of the teacher, robot, and student into state-action trajectories by computing the Viterbi path of each of episode of teaching between a particular student and a teacher, which is the most likely path of states that generated a particular set of observations under a specific HMM. We then pair each state along this path with the corresponding action from the coded data.

The next step in the indirect approach is to define a reward function R. Thereafter, standard reinforcement learning algorithms can be employed to solve for the optimal policy. Several methods of choosing R are possible. Since the teacher's actions are assumed to be an error signal we defined R by assigning low rewards to states in which the teacher was likely to act and high rewards to states in which the teacher did nothing (action 5). Thus the robot was encouraged to learn a teaching policy that would minimize the need for the human teacher to intervene. We can quantify and compute this function precisely using the actions of the teacher Ω and the Viterbi path V. Recall that a_5 is the action in which the teacher did not correct the robot. δ represents the Kronecker delta function.

$$R(s) \quad = \quad 1 - \frac{\sum_{t=0}^{|V|} \delta\left(V_t, s\right) \times \delta\left(\Omega_t, a_5\right)}{\sum_{t=0}^{|V|} \delta\left(V_t, s\right)} \tag{2}$$

Given the reward function R, an optimal policy can now be learned using standard techniques from reinforcement learning using policy iteration.

VI. Results

In this section we compare the direct and indirect approaches on a variety of performance metrics. We also analyze qualitatively the models of teaching that each approach learned.

A. Predicting the Teacher

The coded data from the teaching triads yield a time series of actions and observations. The most direct way to measure whether or not our models have learned patterns from the human expert is to see if the models can predict the actions of the expert given the history of observations.

At each time step our models emit a probability distribution over actions, we measure the predictive accuracy by computing

TABLE I. Correlation coefficients between the direct approaches likelihood assigned to a particular action and a smoothed version of the actual actions coded.

Action	Correlation (direct)	Correlation (indirect)
Repeat object name	0.3118	0.0014
Ask a question	0.2259	−0.0014
Give a hint	0.2247	−0.0037
Give feedback	0.3463	−0.0343
No action	0.3472	0.0699
Average	0.2912	0.0201

the correlation coefficient between the probability of performing a particular action and a smoothed version of the actual actions of the robot and the teacher. A high correlation means that the model is likely to choose a particular action when the joint actions of the robot and teacher are also likely to choose that action. Temporal smoothing is performed using a Gaussian kernel of width 2 seconds. The smoothing is done to assign partial credit for predicting a specific action slightly before or after it actually occurred. Assigning partial credit for small temporal displacements is appropriate given the nature of the interactions between children and RUBI (e.g. reminding the child of the object a second late is not likely to matter too much). We train using teaching sessions from a subset of the children and test on those involving the remaining children.

The results of this analysis are given in Table I. The reported correlation coefficients are for validation data. Across the board the direct approach outperforms the indirect approach. One explanation is that only the direct approach has an explicit goal to reproduce the actions of the teacher. Another is that the underlying model that the indirect approach is based on is not very accurate. For the direct approach the easiest actions to predict are to give feedback and repeat the object name. The reason for the former is probably the large contingency between children getting the right answer and the teacher giving positive feedback. For the indirect approach the only action that has any correlation is the the action of not acting. This modest correlation might make sense since the goal of the indirect model was to act to steer out of states where the teacher is likely to act.

B. Analysis of models

We also analyzed which features were most predictive in the direct approach to apprenticeship learning. To determine the most important features we trained a sequential regression model to predict the action at the next time step using only a subset of the observed features: starting with an empty pool of features we add the feature that increases the performance of the model the most. The notion of performance is defined as in Section VI-A. In our analysis we select features as a group where the group is defined as a single attribute over all temporal scales (e.g. did the child get the right answer in the last 1 second, 2 seconds, 3 seconds, etc.). Once we have added the first group of features we then select the second group of features that when used in conjunction with the first gives the highest overall performance. The results of this analysis are given in Table II.

The features selected by sequential regression indicate that some actions are well predicted by an auto-regressive model. For example, predicting whether or not to repeat the object name is best accomplished using the history of whether or not the object name has been repeated in the recent past. In contrast, for some actions there is a strong contingency between the actions of the teacher and the actions of the child (e.g. between teacher giving positive feedback and the child getting the correct answer).

Figure 5 shows shows the probability of the direct model choosing a particular action as a function of a particularly salient feature. Several interesting phenomena can be seen. First, our system learns to occasionally repeat the object name in quick succession. This trend also emerged in the expert data and might serve the purpose of placing emphasis on particular utterances. A second trend that emerged was a strong contingency between the child answering a question correctly and the model recommending that positive feedback be given immeditately. Intuitively this makes sense as a reward mechanism for desirable behavior. The third trend that emerged was that the longer it has been since a child has gotten a correct answer, the more likely the model is to recommend giving the child a hint. The underlying intuition could be that children who are having a difficult time need extra guidance.

C. The Rhythm of Teaching

From observing the videos of the interactions of RUBI, the teacher, and the child we hypothesized that the rhythm of interactions between the teacher and the child may play an important role in teaching. To see if our models learned any such rhythm we computed the power spectrum of the binary sequence of action / no action from the demonstrations and compared it to the same sequence generated from the predictions of our models. To compute the power spectrum we divide the teaching episodes into two-minute chunks and extract the power spectrum using the Fourier Transform. As a measure of closeness between the two spectra we use the cosine of the angle between the two power spectra (treating each as a vector in a high-dimensional space). The result is a similarity of 0.9687 for the direct model. The resulting similarity for the indirect model is 0.9420. The baseline performance for RUBI's original teaching module is 0.9115. While the importance of rhythm in teaching is an open question, this result suggests that fundamentally these models are capable of learning such a rhythm.

VII. Conclusion

This pilot work serves as a first step to illustrate how to use apprenticeship learning for building controllers for social robots. We demonstrate the utility of various models in predicting the actions of a human teacher. Analysis of the features most relevant for predicting each action suggests that our approaches have learned intuitive models of teaching.

In this pilot, the direct approach outperformed the indirect approach. This may be due to the fact that in social situations

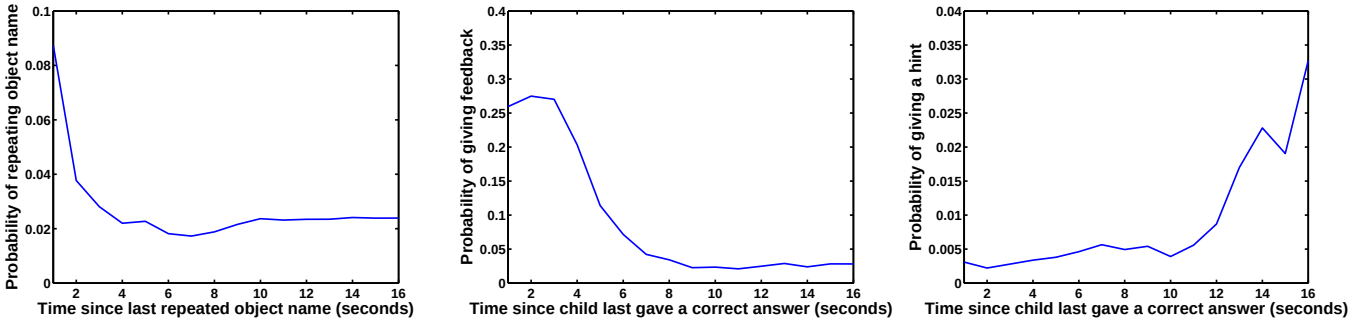

Fig. 5: The relationships learned using the direct approach between three predicted actions and a particularly salient feature for each action. *Left*: the model learns to occasionally repeat the name of the object twice in a row. *Center*: the model learns to give feedback immediately following a correct answer from the child. *Right*: the model learns to give hints to children that have not answered correctly in a while. These trends can be seen in the expert's data as well.

TABLE II. The first two features selected by the direct approach in the task of predicting a particular action. Each model was trained using sequential regression to predict one action versus the others. In sequential regression, at each iteration we add the one group of features that improves our performance the most.

Action	First feature selected	Second feature selected
Say object name	said object name	incorrect answer
Ask a question	asked a question	said the object name
Give hint	gave a hint	gave feedback
Give feedback	correct answer	wrong answer
No Action	object name	correct answer

precise models of the system dynamics are not available and thus they need to be learned from the data. It appears than in such situations it may be more efficient to use the data to directly learn a controller than to learn a model of social dynamics. The situation may change, however, as better models of social dynamics are developed. Just as control theory in the domain of mechanical tasks was greatly spurred by the development of realistic physical models, so can the control theory of social domains be enhanced by a greater understanding of the laws and regularities of interactions between social entities. Recording more relevant perceptual data that RUBI already has the capabilities to extract, such as facial expression recognition and auditory scene analysis, might allow for a better model of the social world and thus more intelligent reactions to rapidly changing conditions.