# Building a More Effective Teaching Robot Using Apprenticeship Learning

Paul Ruvolo, Jacob Whitehill, Marjo Virnes and Javier Movellan
Institute for Neural Computation
University of California San Diego
La Jolla, CA 92093-0445
{ 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 [1] 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 [2], 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 of a human expert.**

## I. Introduction

In the RUBI project at UCSD, we are exploring the potential of using interactive social robots as tools for assisting teachers in early childhood education environments [3]–[5]. 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 target skills that are critical for social robots to become effective teachers. One of the initial priorities in the project was to develop robust perceptual primitives for social interaction, including facial expression recognition, and auditory mood analysis (e.g., detecting cries [6]). As these systems are developed the challenge shifts to the problem of integrating them with the robot's actuators so as to produce effective teaching behavior. *Apprenticeship learning* is a potential framework for helping achieve this integration in a principled manner.

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 [1], 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 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 conducted a preliminary study to improve RUBI'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 [7] for an overview of approaches and challenges). In particular, Du Boulay and Luckin [8] 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 [9]). In our work we take a different tact: while much of pedagogical research tends to be theory driven (develop hypotheses and perform behavioral experiments to test their validity), 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 analyzing the models learned in this framework it may also be possible to develop new theories about the factors that define good teaching.

*Apprenticeship learning* is a method for solving control problems by incorporating demonstrations from an expert. Much of the work in this area has focused on having humans operate a device, such as a robotic arm or a helicopter [1], and recording states of the device along with actions that the human performed in these states. These demonstrations provide constraints on appropriate actions 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 [2], 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 observable features of the current teaching environment or partially-observable attributes of a child's cognitive and emotional state. Enumerating which of these many features are relevant for a teaching interaction is a very difficult problem.

Further complicating matters is the notion of dynamics: In the helicopter case, Newtonian mechanics provides a precise model of the system dynamics: An accurate model of the world can be learned by coupling classical physics with empirical determination of parameter values [10]. In contrast, in the teaching setting we don't have the social equivalent of Newtonian mechanics. As such probabilistic models of social dynamics need to be learned from the available data. Thus it is unclear a priori whether the apprenticeship methods that have worked well for controlling physical dynamical processes would also work well for controlling social processes.

## III. Learning to Teach

In this study, we explored two apprenticeship learning approaches to improve RUBI's teaching capabilities. The current version of RUBI (RUBI-4) consists of a touchscreen Tablet PC which she uses to play various educational games, a head with a pan servo, and two arms with 3 DOFs each. The head movement is guided by balancing exploring the world and locking onto faces. Arm movements are used as both an expression of RUBI's emotional state and to receive objects from children. In this work we focus solely on the algorithm used to coordinate RUBI's actions with the presentation of an educational game on RUBI's touchscreen.

In this work, instead of the human teacher providing 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, i.e., the robot, the human teacher, and the pre-school pupil form a *teaching triad* (see Figure 1). Apprenticeship learning approaches (defined in Section V) are then used to learn an improved robotic teaching policy. Two key assumptions are implicit in this formulation. The first is that the actions of RUBI will have a similar effect as the human teacher performing that same action. The second is that the human teacher acts to supplement the robots actions in order to implement a particular teaching strategy. This second assumption implies that we know that RUBI is teaching well (in other words, implementing a teaching strategy similar to the human demonstrator) 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.

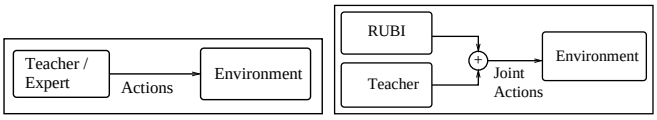


**Fig. 1. Left:** the conventional apprenticeship learning setting. **Right:** the teaching triad we use for our approach.

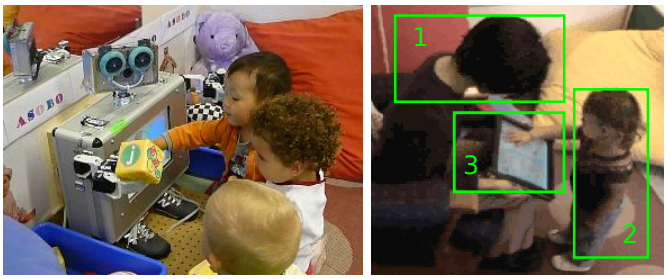


**Fig. 2. Left:** RUBI teaching children at ECEC. **Right:** A teaching triad consisting of a teacher from ECEC (1), a student (2), and a stripped-down version of RUBI consisting of a touch-screen tablet PC (3). Data from these interactions was coded by humans into $9$ behavioral channels as shown in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

### A. Dataset of teaching demonstrations

The original purpose of the study was to find methods to better assess the children vocabulary skills. To this effect we asked a human teacher to use a stripped-down version of RUBI consisting of only her touch-screen tablet PC. The teacher positioned the touchscreen on her belly, thus approximating the teaching setting used by RUBI (see Figure 1). In these sessions, the child was playing the "Name the Object Game" in which RUBI displays four different objects on her touchscreen (see Figure 2) and asks the child to touch a specific object using an auditory prompt (e.g. "Where is the apple?"). RUBI is equipped with a simple, teaching module that periodically reminds the child, at a fixed frequency, which object to touch. This simple "teaching" strategy was programmed by hand and could clearly be improved upon. The teacher was asked to try to engage the child as much as possible so as to to elicit his or her true knowledge of the vocabulary being assessed by the game.

A total of $8$ preschool students participated in this study. Each child interacted one-on-one with the teacher for an average of $4$ 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. The actions coded in this pilot study are not meant to be comprehensive. We discuss our plans to record additional information channels (such as facial expressions and auditory categories) in Section VII.

## IV. Mathematical Framework

A widely used [1], [11], [12] 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

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 "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

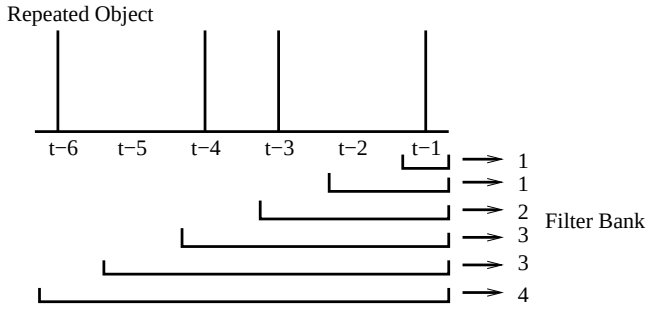


Fig. 4: *Top*: a stem plot of the binary time series of the feature of whether the object name has been repeated. *Bottom*: the various temporal kernels that are extracted from this time series. These temporal kernels are used as input to supervised learning methods for predicting the next action.

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 and arriving at a particular state; and $\gamma \in [0, 1)$ is a discount factor that specifies how much to weight immediate versus future rewards.

A *policy* is a mapping $\pi : S \rightarrow A$ from a state to an action. The goal is to find a policy, $\pi^\star$, that maximizes a notion of desirability, e.g., the expected 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 $\pi^\star$ [13]. 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 $i$th trajectory. We use the symbol $U = (u_1 \ldots u_m)$ to refer to a set of $m$ trajectories demonstrated by the expert.

Assuming that $S$ and $A$ (the states and actions of an MDP) are known, but that $P$ and $R$ (the transition dynamics and reward function) are unknown, the expert trajectories $U$ can be used to find a policy $\pi$ that maximizes Equation 1. We explore two methods for 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. The direct approach has been deployed widely in the field of robotics [14], [15]. The idea is that we assume the expert's behavior is approximately optimal; therefore, by mimicking his or her behavior, a policy that approximately maximizes Equation 1 can be implicitly learned.

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 dynamic programming techniques can be applied to obtain the optimal policy $\pi^\star$ [13]. 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 [1]. Other approaches estimate $R$ by choosing the reward function that makes the actions of the expert appear optimal with respect to $R$ [11], [12]. 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

We modeled the "teaching" situation as follows: 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 constitute the set of actions, $A$, for an MDP. The crucial difference between the direct and indirect approach are 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, a mapping from states to actions is created using supervised learning techniques on the data collected from the expert demonstrations (see Section III-A). The action space for this approach is defined in the preceding section. In the present study, we focused on learning the *timing* of teaching events – e.g., how often to repeat the name of the object, how long after the child's last correct answer to

offer a hint, etc. To enable such temporal relationships to be learned, we defined 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 then applied a bank of filters to convert the recorded actions and observation histories into a series of features. The set of observations of the child (items 5 through 8 in Figure 3) can be detected by RUBI in autonomous mode (i.e. not stripped-down) either through her touchscreen or through her proximity sensor. Each of RUBI's four possible actions (excluding the action of doing nothing) forms a feature channel as well. We extract features from the history using filter kernels of sixteen different temporal scales: 1 seconds, 2 seconds, ..., 16 seconds (see Figure 4). The combination of 16 temporal scales, 4 observations, 4 actions gives us a total of $16 \times 4 + 16 \times 4 = 128$ 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. In this work we used multinomial ridge logistic regression [16]. Logistic regression outputs a matrix of weight values that express the log-probabilities of choosing a particular action given the input features (history of actions and observations). These probabilities can be readily interpreted; we analyze the model of teaching that the direct approach learns in Section VI-B.

### B. Indirect Approach

The indirect approach involves learning a model of how the actions of a teaching agent affect the world (the states, and state transition probabilities), assigning a notion of desirability (reward function) to each state in our system, and then using 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 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. We use an extension of the standard HMM that learns 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, which is the most likely path of states that generated a particular set of observations under a specific HMM, of each of episode of teaching. 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 $R$ using the training data. 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). The robot was encouraged to learn a policy that would minimize the need for teacher intervention. We define this reward using the actions of the teacher $\Omega$ and the Viterbi path $V$. Recall that $a_5$ is the action in which the teacher did not correct the robot. $\delta$ represents the Kronecker delta function.

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

Given the reward function $R$, an optimal policy is learned using policy iteration.

## VI. Results

In this section we compare the direct and indirect approaches on a variety of performance metrics. We also provide quantitative analysis of the models learned.

### A. Predicting the Teacher

The most direct way to measure whether 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 that our models emit a probability distribution over actions, we measure the predictive accuracy by computing the correlation coefficient between the probability of the learned policy selecting a particular action and a smoothed version of the actual actions of the teacher. A high correlation means that the model is likely to choose a particular action when the teacher is also likely to choose that action. Temporal smoothing is performed using a Gaussian kernel of width 2 seconds. The smoothing assigns partial credit for predicting an action slightly before or after it actually occurred.

The results of this analysis using a leave-one-child-out cross-validation scheme are given in Table I. 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 on which the indirect approach is based 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, correlation values are very low. This suggests that the HMM learned for the indirect approach is not adequately characterizing the temporal properties of the data from the teaching triads.

### B. Analysis of models

We analyzed which features were most useful for prediction in the direct approach. To determine the most important features we trained models to predict a particular action versus all of the others using only a subset of the observed features: starting with an empty pool of features we add the

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

| **Action to Predict** | 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 |

feature that increases the performance the most. The notion of performance is defined as in Section VI-A. We select features as a group which includes a single variable 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 in conjunction with the first gives the highest overall performance. The results of this analysis are displayed in Table II.

The features selected by sequential regression indicate that some actions are 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 child and the actions of the teacher (e.g. between the child getting the correct answer and the teacher giving positive feedback).

Figure 5 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 immediately. Intuitively this makes sense as a reward mechanism for desirable behavior. Another trend learned by the model was that the longer it has been since a child has given a correct answer, the more likely it should be to give 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 actions recommended by the trained models. The Fourier power spectrum was computed using non-overlapping two-minute windows. The cosine between the two spectra was used as a similarity measure. The result is a similarity of 0.9687 for the direct model and 0.9420 for the indirect model. 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

The pilot study presented here serves as a first step to illustrate the potential apprenticeship learning for building controllers for social robots. We compare the utility of two approaches in predicting the actions of a human teacher. Analysis of the features most relevant for predicting each action suggests that the approaches learned important aspects about the timing of behaviors during teaching.

In this pilot, the direct approach to apprenticeship learning outperformed the indirect approach. One reason for this may be that in social situations indirect approaches cannot capitalize on precise prior models of the system dynamics. 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.

## References


[1] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning."

[2] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng, "An application of reinforcement learning to aerobatic helicopter flight," in *Advances in Neural Information Processing Systems 19*. Morgan Kaufmann, 2006.

[3] T. F., C. A., and M. J. R., "Socialization between toddlers and robots at an early childhood education center," *Proceedings of the National Academy of Sciences*, 2007.

[4] M. J. R. T. F., T. C., R. P., and E. M., "The rubi project: A progress report," in *Proceedings of the 2nd ACM/IEEE international conference on human-robot interaction*, 2007.

[5] J. R. Movellan, F. Tanaka, B. Fortenberry, and K. Aisaka, "The RUBI project: Origins, principles and first steps," in *Proceedings of the International Conference on Development and Learning (ICDL05)*, Osaka, Japan, 2005.

[6] P. Ruvolo, I. Fasel, and J. Movellan, "Auditory mood detection for social and educational robotics," in *International Conference on Robotics and Automation*, 2008.

[7] C. Breazeal and B. Scassellati, "Challenges in building robots that imitate people," 2001.

[8] B. du Boulay, "W1 - modeling human teaching tactics and strategies," in *ITS '00: Proceedings of the 5th International Conference on Intelligent Tutoring Systems*. London, UK: Springer-Verlag, 2000, p. 662.

[9] W. Burleson and R. Picard, "Affective agents: Sustaining motivation to learn through failure and a state of stuck," in *ITS '04: Proceedings of the 7th International Conference on Intelligent Tutoring Systems*, 2004.

[10] P. Abbeel, V. Ganapathi, and A. Y. Ng, "Learning vehicular dynamics, with application to modeling helicopters." in *NIPS*, 2005.

[11] A. Y. Ng and S. Russell, "Algorithms for inverse reinforcement learning," in *Proc. 17th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 2000, pp. 663–670.

[12] G. Neu and C. Szepesvri, "Apprenticeship learning using inverse reinforcement learning and gradient methods," in *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2007, pp. 295–302.

[13] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

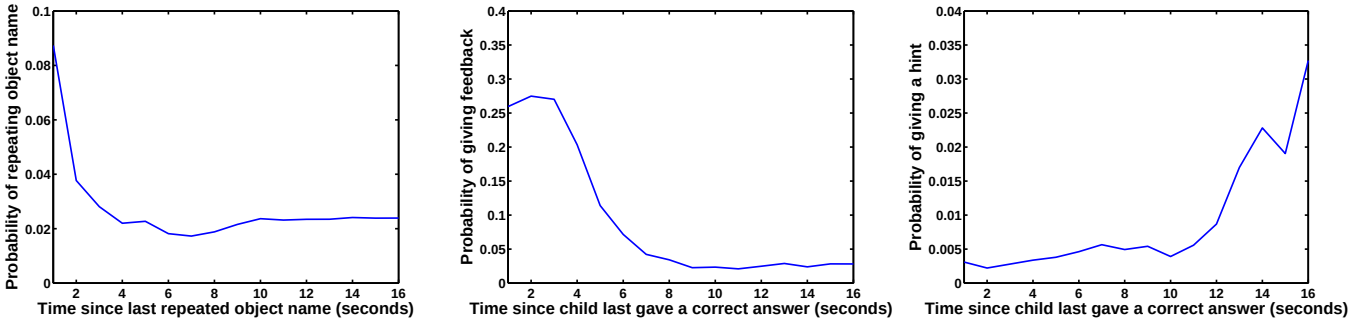


Fig. 5: The relationships learned using the direct approach between three predicted actions and a particularly salient feature for each action. *Left*: the model learns to occasionally repeat the name of the object twice in a row. *Center*: the model learns to give feedback immediately following a correct 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 by sequential regression using the direct approach for the binary task of predicting a particular action versus all the other actions. In sequential regression, at each iteration the feature that improves the performance the most is selected. In this work a particular feature includes all of the temporal kernels associated with that feature (e.g. the history of a particular feature over a 16 second window.

| | **Features Selected For Prediction** | |
|---|---|---|
| **Action to Predict** | First feature selected | Second feature selected |
| Teacher says object name | teacher or robot said object name | child gave incorrect answer |
| Teacher asks a question | teacher or robot asked a question | teacher or robot said the object name |
| Teacher gives a hint | teacher or robot gave a hint | teacher or robot gave feedback |
| Teacher gives feedback | child gave correct answer | child gave wrong answer |
| No Action | teacher or robot said object name | child gave correct answer |

[14] D. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," in *Advances in Neural Information Processing Systems 1*. Morgan Kaufmann, 1989.

[15] C. Sammut, S. Hurst, D. Kedzier, and D. Michie, "Learning to fly," in *Proceedings of the Ninth International Conference on Machine Learning*. Aberdeen: Morgan Kaufmann, 1992.

[16] M. J. R., "Tutorial on multinomial logistic regression," *MPLab Tutorials. http://mplab.ucsd.edu*, 2006.