2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Predictive State Representations for Grounding Human-Robot Communication

Eric Meisner[1], Sanmay Das[2], Volkan Isler[3], Jeff Trinkle[2], Selma Šabanović[4], and Linnda R. Caporael[5]

[1]Department of Computer Science, Johns Hopkins University, Baltimore MD, 21218
[2]Department of Computer Science, Rensselaer Polytechnic Institute Troy NY, 12180
[3]Department of Computer Science and Engineering, University of Minnesota, Minneapolis MN, 55455
[4]School of Informatics and Computing, University of Indiana Bloomington
[5]Department of Science and Technology Studies, Rensselaer Polytechnic Institute Troy NY, 12180

Abstract— **Allowing robots to communicate naturally with humans is an important goal for social robotics. Most approaches have focused on building high-level probabilistic cognitive models. However, research in cognitive science shows that people often build common ground for communication with each other by seeking and providing *evidence of understanding* through behaviors like mimicry. Predictive State Representations (PSRs) allow one to build explicit, low-level models of the expected outcomes of actions, and are therefore well-suited for tasks that require providing such evidence of understanding. Using human-robot shadow puppetry as a prototype interaction study, we show that PSRs can be used successfully to both model human interactions, and to allow a robot to learn on-line how to engage a human in an interesting interaction.**

I. INTRODUCTION

Just as computers started out as academic and industrial tools before becoming part of daily life, robots cut their teeth in factories but have now begun to enter the domestic domain. Robots that vacuum floors and mow lawns are commercially available, and in the coming years, robots will likely become increasingly common in homes. Robots have the potential to provide cognitive and physical assistance to an increasing elderly population which could alleviate the strain on the health care system and improve their quality of life by letting them remain independent for longer. However, in order for robots to be maximally helpful, they must not make their user's lives more difficult by being dangerous or unpleasant to deal with. One particularly challenging project is to make systems that communicate naturally with humans.

How can artificial agents, in particular embodied robots, participate meaningfully in interaction with humans? A social interaction with a human can be thought of as operating in a highly uncertain stochastic dynamic system. Traditionally, in AI, robotics, and control theory, an agent starts with a model of the world and that model dictates the way it behaves and learns. This approach has produced impressive results for certain types of difficult robotics problems, such as helicopter flight [1] and autonomous vehicle control [2]. However, adapting this approach to social robots leads to an array of problems. At the very least, a model of embodied interaction with humans would need to be orders of magnitude more complex than the kinds used to successfully perform typical probabilistic reasoning tasks [3]. A different approach to enabling human-robot interaction is suggested by 'emergent' theories of cognition and social cognition, which say that the understanding, meaning, and rules of social interaction are not a property of the world, but rather something that is agreed upon. Taking this view, the goal of a social robot is to create shared meaning with another agent. The nature of the problem is fundamentally different from navigation or manipulation. Existing frameworks for learning to control dynamical systems may, therefore, not be suitable for learning to participate in embodied discourse. However, it is still important to understand the stochastic dynamics underlying social interactions.

In this paper we focus on low level probabilistic models for understanding how others react to one's actions in discourse. Specifically, we propose the use of predictive state representations (PSRs [4], [5]) in order to learn expected responses to particular behaviors. This replicates the social process of *signal grounding*, in which participants in a dialog look for patterns in the exchange of social behaviors and learn to predict responses to their actions [6], [7]. Signal grounding is an essential component of a higher-level procedure called symbol grounding, which uses these learned responses to establish common meaning. If a robot is to learn to communicate with humans from the ground up, a first step would be to predict responses to particular actions or behaviors. We demonstrate the feasibility of this in an experimental human-robot interaction domain, shadow puppetry [8]. The limited range of actions combined with the ability to map sensory inputs to particular observations in the observation space with high fidelity make this an ideal domain for exploring reinforcement learning and control algorithms. At the same time, the limitation on the number of actions does not preclude participants from exhibiting complex low-level behaviors like imitation, anticipation, and coordination.

A. Contributions

We describe how to use PSRs to model interactions between humans and between humans and robots in the shadow puppetry domain. We show that we can model interaction as

a stochastic dynamical system by learning accurate PSRs with good predictive power on data from human-human shadow puppetry interactions (Section III). We propose a robot control algorithm based on the PSR representation that allows a robotic hand to learn on-line how to behave when interacting with a human controlled shadow puppet, and demonstrate in a small pilot user study the feasibility of (i) learning the intent of the human, and (ii) engaging humans in interesting social discourse (Section IV). Overall this work serves as both a demonstration that PSRs can be useful for real-world tasks, and as a proof-of-concept that allowing agents to build and manage their own representations can succeed in tasks where specifying a model of desired behavior is difficult.

II. BACKGROUND AND RELATED WORK

Control is essential to interaction [9]. Learning to evoke and predict responses from others is an important part of social learning. The natural approach to this problem therefore seems to be a decision-theoretic one that models a dynamical system that is not completely observable, such as a partially observable Markov decision process (POMDPs). The descriptive power of these frameworks is attractive for real-world robot tasks because it enables agents to reason about the result of future actions by interacting with the world and receiving feedback using imperfect perception. Typically, real-world successes of POMDPs have been in cases where the state and observation spaces of a system can be described in very few terms [10]. Several recent successful approaches that utilize POMDPs for assistive human-robot interaction have applied them in a top down fashion [11], [12], [13]. They use a representation with few states, providing a coarse description of the world. The actions for these POMDPs are complex tasks, such as getting on an elevator and escorting a human to a location using sensor feedback motions.

The POMDP formulation is suitable for certain situations, especially if the structure of the model fits the particular application domain. For problems where a robot or agent must learn to interact socially with a human, it is tempting to model the human as a system that generates behaviors using one of these frameworks. The agent is left with the task of learning about the parameters of the system through its actions and observations. POMDPs closely resemble the "sense-think-act" model of cognitive architectures. States encode different "modes," and when an agent determines that a particular mode is active (with some probability), it is assumed to be responsible for the behavior of the system. This structure forces the designer to make many assumptions about the set of possible modes, the process of switching between modes and when a mode results in a particular behavior. This is problematic on many levels; for example, when the mode to which we attribute a behavior is incorrect or invalid. It is easier to get around these problems in physical domains like navigation than in domains like social interaction.

The theory described by Semin [7] suggests that language and other high level aspects of communication are based on synchronization or parity of behaviors. This idea proposes that mimicry, parity and correspondence allow humans to seek and provide evidence of understanding. The processes that generate these behaviors are non-cognitive and more strongly connected to physical experience than to high-level reasoning. This is incompatible with sense-think-act learning models, and suggests that we need to use models that more immediately couple actions and observations. Interestingly, one of the major thrusts in reinforcement learning in the last few years has been the development of predictive state representations, which seem to fit the need for such models perfectly.

A. Predictive State Representations

PSRs are a relatively new technique for modeling interactions between an agent and a system [14]. The general assumption is that this interaction is an nth-order Markov process, meaning that any future sequence of action-observation pairs is a function of some fixed part of the most recent history. The PSR learns a function from histories of actions and observations to future predictions. Predictions are sequences of actions and observations and are referred to as *tests*. The mapping function describes the probability that a given test will *succeed*, meaning that executing the sequence of actions in the test will cause its observation sequence to occur. In general, there is a minimal set of tests for any system that are sufficient to predict the outcome of all other tests. These are the *core tests*. The minimal state representation for the PSR is the set of predictions for each core test.

PSRs focus specifically on inputs and outputs without using coarse descriptors of state to model their cause. The PSR maintains a distribution over the set of possible future action-observation sequences. This distribution, and the parameters for updating the distribution, are learned from experience. PSRs may thus be able to accurately predict the results of future actions without making strong assumptions about, or partitioning, the unobservable system. They are also capable of effective off-policy learning. Our brief description of PSRs here is based on the more general treatments of Singh *et al.* [15] and McCracken *et al.* [5].

Formally, say an agent chooses actions a from a set A, and the system responds and generates an observation o from a set O. A PSR models the probability of seeing a particular sequence of action-observation pairs $q = a^{t+1}o^{t+1} \dots a^m o^m$ given some previous sequence of actions and observations $h = a^1 o^1 \dots a^t o^t$. The sequence q is referred to as a *test* and h is referred to as a *history*. The goal of a PSR is then, given a set of tests $Q = \{q_1 \dots q_n\}$, maintain a prediction vector containing the probability of each test conditioned on the current history, $\Pr(Q|h)$. The entries are $\Pr(Q|h) = [\Pr(q_1|h), \Pr(q_2|h) \dots \Pr(q_n|h)]$

Each entry gives the probability of a particular test succeeding, given a particular previous history. The set of tests in Q are the core tests. In order for the system to be a

PSR, the set of tests Q must have the property that, for any test $q \notin Q$, and history h, there exists a function $f_q(\cdot)$, s.t. $\Pr(q|h) = f_q(\Pr(Q|h))$. In the case of a linear PSR, the function is $\Pr(q|h) = f_q(\Pr(Q|h)) \equiv \Pr(Q|h)^T \mathbf{m_q}$, where the $\mathbf{m_q}$ are internal parameters. After observing a history h followed by an action observation pair ao, the entry for test q_i is (using a simple application of Bayes rule) updated as, $\Pr(q_i|h_t ao) = \frac{\Pr(aoq_i|h_t)}{\Pr(ao|h_t)} = \frac{\Pr(Q|h_t)^T \mathbf{m_{aoq_i}}}{\Pr(Q|h_t)^T \mathbf{m_{ao}}}$.

Note the implicit assumption that vectors $\mathbf{m_{aoq_i}}$ and $\mathbf{m_{ao}}$ are available. These vectors are the parameters of the PSR and for each test $q \in Q$, it is necessary to maintain a vector of all such "one step" extensions to q. This means that for every action-observation pair $ao, a \in A, o \in O$ and every test $q \in Q$, a projection vector $\mathbf{m_{aoq}}$ is maintained. This includes the zero length test ε, so that all $\mathbf{m_{ao}}$ are also maintained.

The need for $\mathbf{m_{ao}}$ and $\mathbf{m_{aoq_i}}$ parameters means that even a linear PSR suffers from the curse of dimensionality. The $\mathbf{m_{ao}}$ are the parameters of the PSR and can be learned using temporal difference methods and the gradient of the prediction error [15]. McCracken and Bowling [5] highlight the problem of enforcing that the state vector $\Pr(Q|h_t)$ contains valid probabilities.

B. Discovery

In addition to the parameters $\mathbf{m_{ao}}$ and $\mathbf{m_{aoq_i}}$, the PSR maintains a representation of observed history in the form of a system dynamics matrix $D = \Pr(Q|H)$. Each row i of D represents the state of the system at time i so that for each test q_j, $D_{ij} = \Pr(q_j|h_i)$. Computing the linearly independent columns in D provides a means of automatically determining a minimal representation for a system by finding a small set of core tests that yields accurate predictions. There are several ways to accomplish this. McCracken and Bowling [5] suggest incrementally adding a test t to the set of core tests Q by computing the condition number of the matrix $\Pr(\{Q, t\}|H)$. If the condition number is less than 1, the matrix is well conditioned and t is added as a core test. Another method is to compute an SVD or QR decomposition and keep tests corresponding to columns in D that are approximately linearly independent. For the QR-decomposition $D = QR$ so that Q is an orthogonal matrix, $Q^TQ = I$, and R is upper triangular. If D has non-pivot columns, then R contains columns which express these non-pivot columns in terms of those columns in Q. The first nonzero (pivot) elements in each row of R determine the location of the linearly independent columns in D. In practice, we can determine columns in D that are approximately linearly independent according to the diagonal elements of R which are above a threshold. The linearly independent columns of D are kept as core tests. The orthogonality of Q can provide a measure of numerical error (for example, by comparing the Frobenius norm of Q^TQ to that of I).

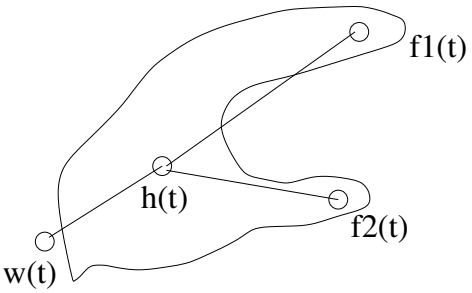
III. HUMAN-HUMAN INTERACTION MODELS

As mentioned above, in keeping with theories of cognition and social learning, we would like to minimize the imposition of unnecessary structure in modeling social robot learning. We use PSRs to study discourse by observing actual behavior without explicitly modeling purpose or intervening changes in modes. The discovery algorithms of PSRs are particularly interesting because they allow us to determine which sequences are most important in modeling the system, and this can be directly related back to the notion of signal grounding.

A. Shadow Puppetry

Shadow puppetry is a human-robot interaction domain that provides a means of observing an embodied discourse between two people or between a person and a robotic hand. The domain is expressive enough to support basic components of interaction and allows participants to convey and infer the meaning of emotive gestures [8]. At the same time, it limits the channels of communication sufficiently that we do not need to solve difficult perception or action-generation problems. It is feasible to capture and model signals in real-time using available computational and perception tools. Subjects are asked to participate in open-ended interactions. The behavior of each player is converted to a one dimensional signal using a behavior recognition system. Let $\Sigma = \{Nod, Talk, None\}$ denote the set of possible signals and $X, Y \in \Sigma^*$ denote the behavior sequence of players 1, and 2 respectively.

We use a perception system that recognizes the basic motions used in the shadow puppet game. Participants wear simple colored wrist markers that allow us to automatically infer wrist position, and from that to infer the locations of the hand center and fingertips. The hand contour is determined by searching the image for the skin colored blob nearest to the wrist marker. The wrist $w(t)$, hand center $h(t)$, and finger tip locations $f1(t), f2(t)$ as shown in figure 1(a) provide a rough kinematic model of the hand.



(a) Hand model parameters

(b) Gesture Labeling

Fig. 1. Automated gesture recognition system.

To automate gesture recognition, we record the parameters of the kinematic model in each frame. In order to identify

behaviors, we measure the statistical dispersion of these parameters over the most recent history of length $n = 7$. We calibrate our gesture recognition system for each user. During the training phase, the user performs each of the gestures several times. The vector of behavior parameters is computed and recorded for each example. We then fit a Gaussian distribution to each gesture class and use this mixture of Gaussians to classify gestures performed by the human. This system allows us to convert the behavior of each player to a one dimensional signal which codes his or her behavior (from the set of behaviors $\Sigma = \{Nod, Talk, None\}$) at each instant of time, in real-time.

B. Modeling

In order to make the problem of on-line learning of PSR parameters tractable, we have to reduce the space of possible actions. We focus on a property of interaction considered essential to social learning: imitation. We determine the actual behaviors X_t and Y_t of of players 1 and 2 at time t, and then code our data so that the behavior of a player is a 0 whenever it matches the previous behavior of his or her partner and a 1 otherwise. More formally we describe the imitation behavior at time t of players 1 and 2 respectively with variables A_t and O_t. $A_t = 0$ if $X_t = Y_{t-1}$, and 1 otherwise, and $O_t = 0$ if $Y_t = X_t$ and 1 otherwise. Then the PSR has two actions and two observations. The behaviors of the first player are the agent actions, and the behaviors of the second player are observations.

C. Human-Human Experiment

In this experiment, the data is formatted so that there is a 0 if a player matches the behavior of their partner and 1 otherwise. So, if at a given time step the action observation pair is [0,0], and player 2 changes behaviors in the next step, the next pair will be [0,1], and [1,1] will follow in the next step, unless player 1 changes to match player 2.

The data is primarily composed of repeating sequences of the action-observation pairs [0,0] or [1,1]. These sequences result from participants repeating their behaviors for short periods of time. When the behaviors match, we see a sequence of [0,0]. When they do not, we see [1,1]. The repeating sequences are terminated by a particular action-observation pair. For example, if a long sequence of [1,1] is followed by a [0,0], or [1,0], it marks the start of a repeating sequence of [0,0]. Rarely is a sequence of repeating [1,1] followed by a [0,1]. Similarly, a repeating sequence of [0,0] is often terminated by [1,1] or [0,1], but almost never followed by [1,0]. This pattern can be described using the first-order, 2-state POMDP in figure 2. The two states represent the repeating sequences [1,1] and [0,0]. The edges are marked with an action, an observation and a probability that the observation and transition will occur, given the action. The parameters of the POMDP are α, β, δ, and ϵ, which are $\ll 1$ and > 0.

We use a PSR representation to try and model actual sequences of behavior between several different pairs of human participants. There were 4 test subjects and each

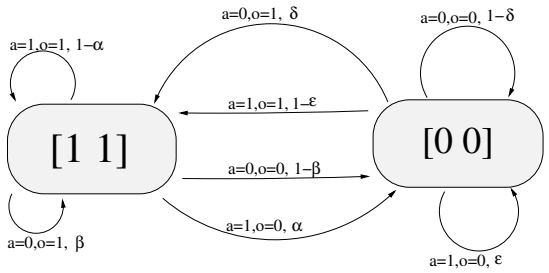


Fig. 2. A 2-state POMDP describing the pattern of interaction with parameters $0 < \alpha, \beta, \delta, \epsilon \ll 1$. The agent can remain in either of the two states [1,1] and [0,0], with high probability, by selecting 1 or 0 respectively. It can cause a transition out of [1,1] or [0,0], with high probability, by selecting 0 or 1 respectively.

sequence represents an interaction between a unique pair. Interactions took place for approximately 2 minutes. All sequences contain approximately 1000 action-observation pairs.

The performance of a PSR on modeling a particular human-human interaction can be measured by prediction error. Prediction error is 1 if $\Pr(o_t) \leq 0.5$, 0 otherwise. We simulate the on-line learning problem by processing actions and observations sequentially, without allowing the model access to future histories. For the PSR, all projection vectors and distributions are initialized to be uniform, meaning that each test contributes equally to the prediction of any other test. The gradient algorithm adjusts the weights of projection vectors depending on the frequency of correct guesses [14]. The model uses a decreasing learning rate $\alpha = \frac{10}{100+t}$, where t is the iteration of the learning algorithm. The α parameter dictates the step size used to adjust the projection vectors. In general, using a learning rate that is always less than the per-time-step prediction error prevents over correction. All probabilities are restricted to be in the interval $[1 \times 10^{-4}, 1]$.

In each example, the PSR learns using all of the core tests on the first half of the data. At the halfway point, the discovery algorithm is used to reduce the set of core tests, and the learning algorithm is continued using the reduced set. The results are compared to a simple predictor that ignores the actions of the agent, and always predicts that the next observation will be the same as the previous (this is the most predictive naive algorithm). The initial set of core tests contains all possible tests of length 2 or less. The plots in Figure 3 show the mis-prediction per time-step on each of 6 human-human interactions. The solid line represents the prediction error per time-step of the PSR predictor and the dashed line represents the prediction error per time-step of the simple predictor. The vertical line indicates the point in the sequence where the dimension reduction algorithm is executed. In all cases, the simple predictor has higher error per time-step than the PSR predictor. The PSR improves until eventually it predicts better than the one-step predictor.

Determining a reduced set of core tests reduces the dimension of the PSR, and allows subsequent learning to converge with fewer trials. Each trial starts with 20 core tests (4 tests of length 1 and 4^2 tests of length 2). After the discovery

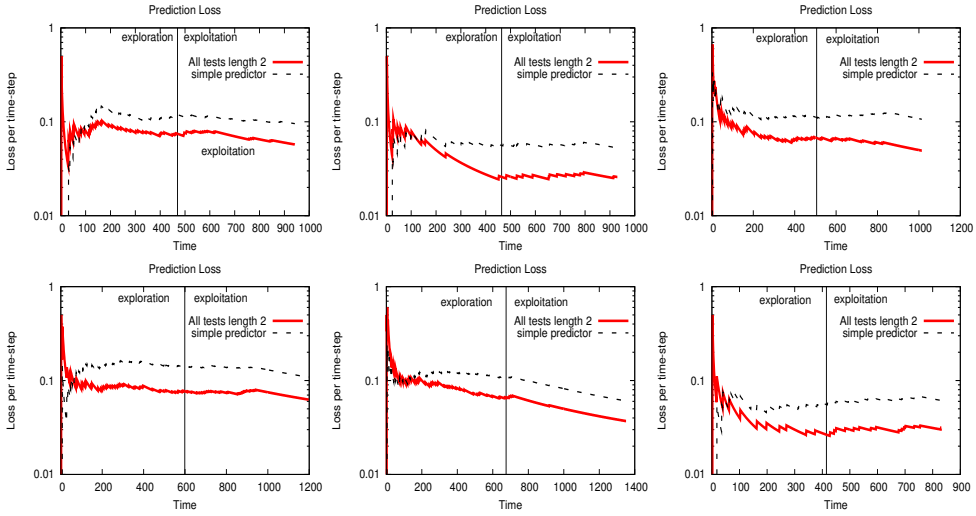


Fig. 3. Prediction rate of PSR on human-human interaction data. Prediction error is 1 if $\Pr(o_t) \leq 0.5$, 0 otherwise. The left and right sides of the vertical bar, respectively, represent the phase before and after the Discovery algorithm.

algorithm, the number of retained core tests for the 6 trials are 8, 11, 13, 14, 13, and 10 respectively. There are two tests that are present in all of the examples: [0 0] [1 0] and [0 0] [0 1]. The tests [1 0][0 0], [1 1][1 0] and [1 1][0 0] are present in all but one of the core sets. Of note is the fact that there is not a significant change in prediction error after the dimension reduction algorithm.

One concern might be that selecting the initial set of core tests to be all length 2 tests, ignores all effects that are greater than second order. It is possible that there are higher order effects contributing to when and why transitions happens. We have some evidence that this is not the case in the shadow puppetry domain. We have analyzed the cumulative distribution of the length of repeating sequences for the 6 simulation trials. These patterns all seem similar to Poisson distributions, which would imply that the probability of transition out of a state does not depend on the length of time spent in the state, consistent with a low-order Markov model (and in fact, with the POMDP model in Figure 2). In practice, adding the set of all length 3 tests did not significantly change the prediction error. When using the set of all length 4 tests, the discovery algorithm becomes intractable, due to the size of the system dynamics matrix (340×340).

IV. BUILDING INTERACTIVE ROBOT-PUPPETS

We have demonstrated that PSRs are capable of capturing patterns in human-human interaction. However, it is also possible to achieve good prediction errors in the task described in the previous section using models like POMDPs. The real reason to use PSRs is to avoid having to pre-program state representations and models, and instead allow robots to build interaction capabilities from basic competencies like action and perception. How can we use PSRs to achieve this goal? In particular, we need to address the problem of action selection.

Recall the theory that social learning occurs as a result of the desire to seek and provide evidence of understanding. This suggests that it may make sense for a robot learning very basic communication to *take actions that are very likely to yield predicted responses*. Given that there is a human-being in the loop who is trying to make herself understood to the robot, and will therefore change her pattern of behavior if she does not receive the responses she expects, this action selection strategy could lead to a successful attempt at establishing shared communication.

A. Controller Learning Algorithm

Algorithm 1 Exploration Schedule PSR Learning Algorithm(set <core tests> CT)

```
Initialize exploration schedule α = 1.0
while Change in One-step Prediction error is large do
    If( RAND(0:1) ≤ α ) EXPLORATION move
    Else EXPLOITATION move
    decay(α)
end while
DISCOVERY
Re-select core tests
loop
    If( RAND(0:1) ≤ α ) EXPLORATION move
    Else EXPLOITATION move
    decay(α)
end loop
```

We implement this idea in the following algorithm. First, the agent selects and executes the test sequences according to exploration utility [1]. The purpose of this stage is to sample the space of action-observation sequences to correctly estimate the system dynamics matrix, D. Once the change

[1]The exploration utility of a test is defined by the number of tests it contains as a sub-sequence. By executing a test, all sub-sequences are also executed, which allows off-policy learning of sub-sequence probabilites. The described measure of utility therefore accounts for the total experience gained by the agent from a executing particular test.

in error becomes small, the discovery algorithm is used to determine a minimal representation for the predicting action-observation pairs. This is done by determining the linearly independent columns in the system dynamics matrix. Finally, the agent continues the interaction by **selecting tests that are likely to succeed**. As described above, this is motivated by the notion that interaction is driven by the need of participants to seek and provide evidence of understanding. Another way to think of this is that certain patterns are established through repeated exchange of behaviors, and the two participants express mutual understanding by following these patterns.

Control algorithm 1 is broken into three stages: exploration, discovery, and exploitation. In the first stage, the learning agent samples the space of action-observation sequences, by repeatedly selecting and executing tests. When the change in prediction error is small, the discovery algorithm is used to reduce the parameter set of the predictor. Finally, the reduced representation is used to select tests for which the observations sequence will likely result from taking the action sequence. In control algorithm 1, the exploration and exploitation phase are mixed, using an exploration schedule. The learning agent selects an exploration move according to a decaying probability.

B. Experiment

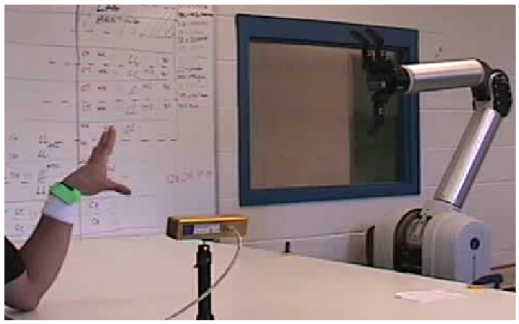


Fig. 4. Embodied interaction study: Subjects play the shadow puppet game with our 4 DOF Whole Arm Manipulator (WAM). A stereo camera is used to code the gestural language tokens of the human in real-time.

In the experiment, human test subjects are stationed across a table from the robot, as in Figure 4. Subjects wear a colored wrist marker which is used to infer the position of the hand and fingertips. The robot can recognize and perform three actions. The gesture classifier is as described above. In order to automatically code the human behaviors, a calibration phase is performed for each individual. The actions of the human are converted into the 0-1 imitation/non-imitation model described previously. Subjects are asked to interact with the robot in two five-minute sessions. Subjects are given a goal and asked to communicate that goal to the robot through their actions. In the first phase, they are asked to make the robot imitate their actions and in the non-imitation phase, they are asked to do the opposite. Each phase starts with the same initial conditions and uses control learning algorithm 1. The robot observes the human gesture, and converts it to either imitate or non-imitate. This provides the observations for algorithm 1. The experiment is designed to evaluate the following hypothesis: **As a result of the control learning algorithm, (1) The robot will perform in accordance with the human's intent. (2) The tests in the predictive state representation will encode the intent of the human.**

Statistic	Phase 1 Q1	Phase 1 Q2	Phase 2 Q1	Phase 2 Q2
Average	4.57	4.86	4.43	3.86
Variance	0.67	0.40	3.29	3.48
Stdev	0.82	0.63	1.81	1.86

TABLE I
SURVEY RESPONSE STATISTICS

At the end of each phase, the subject is asked to fill in a short survey with the following questions: (1) The robot could recognize my actions. (2) The robot could understand my intent. (3) Briefly describe your strategy for this phase. The first and second questions are rated on a scale of 1 to 6, with 1 being not true and 6 being very true. Question 3 is open-ended and designed to control for the fact that subjects' interpretations of instructions may vary. Table I gives the results of the first and second survey questions. On average, the test subjects gave highest scores to the robot's performance in the imitation phase. The results for this phase also have relatively small variance. For the non-imitation phase, the average ratings are considerably lower and the variance is high.

C. Results

We evaluate the algorithm in several ways. First, we look at the parameters of the final representations learned by the robot in interaction with humans. If the human communicated their intent successfully to the robot, there should be evidence in the final representation. In the PSR, the learned linear prediction function is a generative model of human and robot behavior frequencies. This function is encoded in the projection vectors $\mathbf{m_{aoq_i}}$ and $\mathbf{m_{ao}}$, which are used to map state $\Pr(Q|h)$ to $\Pr(q_i|hao)$. The probability of test q_i succeeding, conditioned on a previous action-observation pair ao, is given by the pseudo-probability $\Pr(q_i|h_t ao) = \frac{\Pr(aoq_i|h)}{\Pr(ao|h)} = \frac{\Pr(Q|h)^T \mathbf{m_{aoq_i}}}{\Pr(Q|h)^T \mathbf{m_{ao}}}$.

These vectors may contain arbitrarily large positive and negative values, so in order to evaluate the tendency of a robot to execute a test in the predictive representation, we can set $h = \emptyset$ and determine the conditional probabilities for each test q_i using $\Pr(q_i|ao) = \frac{\Pr(aoq_i)}{\Pr(ao)} = \frac{\sum_j \mathbf{m_{aoq_i}(j)}}{\sum_j \mathbf{m_{ao}(j)}}$. This pseudo-probability provides a measure of likelihood for the success of each test, q_i as well as inclination of the robot to select it.

Recall that action-observation pair [0,0] occurs when the robot and human perform the same action, and [1,1] occurs when they do not. Also recall that sequences [1 0] and [0 1] are considered transitional in our model of imitation. Depending on how imitation is interpreted by the test subject,

Subject	Extension Prefix	Phase 1 test	Phase 2 test
1	[0 0]	**[0 0]**	[0 0]
1	[0 1]	[0 0]	[1 1]
1	[1 0]	[0 0]	[0 0]
1	[1 1]	[1 1]	**[1 1]**
2	[0 0]	**[0 0][0 0]**	[0 0]
2	[0 1]	[0 0]	[1 1]
2	[1 0]	[1 0]	[0 0]
2	[1 1]	[1 1]	**[1 1]**
3	[0 0]	**[0 0]**	**[0 0]**
3	[0 1]	[0 0]	[0 0]
3	[1 0]	[0 0][0 0]	[0 0][0 0]
3	[1 1]	[1 1]	[1 1]
4	[0 0]	**[0 0]**	[1 1]
4	[0 1]	[0 0]	[1 1]
4	[1 0]	[0 0][0 0]	[1 0]
4	[1 1]	[1 1]	**[1 1]**
5	[0 0]	**[0 0]**	[1 1]
5	[0 1]	[0 0]	[0 1][0 0]
5	[1 0]	[0 0]	[1 1]
5	[1 1]	[1 1]	**[1 1]**
6	[0 0]	**[0 0]**	**[1 0][1 1]**
6	[0 1]	[0 0]	[1 0]
6	[1 0]	[0 0]	[1 1]
6	[1 1]	[0 0]	[1 1]
7	[0 0]	**[1 1]**	**[0 0]**
7	[0 1]	[0 0]	[0 0]
7	[1 0]	[0 0]	[0 0]
7	[1 1]	[1 1]	[1 1]

TABLE II

MAXIMUM PROBABILITY CONDITIONAL TEST SEQUENCES. THIS TABLE SHOWS THE MOST LIKELY SEQUENCE FOLLOWING EACH EXTENSION PREFIX FOR EACH SUBJECT. THE BOLD ENTRIES (WITH CORRESPONDING PREFIX) REPRESENT THE MOST LIKELY SEQUENCE FOR EACH PHASE.

we should expect to see differences in the frequency of [0,0] and [1,1] during the first and non-imitation phase of the experiment. Table II shows the sequence with highest conditional probability for each possible prefix and each subject in each phase. In accordance with the interaction model (Figure 2), no transitional sequence ever has highest probability. Results for subject 1,2,4 and 5 are ideal in both phases. For these cases, the most likely sequences in phases 1 and 2 are sequences of [0,0] and [1,1] respectively. For subjects 3 and 6, the most likely sequence in phase 1 is [0,0][0,0], which means that the control algorithm will frequently choose (correctly) to imitate the human. In phase 2, the highest probability sequence for subject 6 is [0,0][1,0][1,1], which indicates a transition to non-imitative behavior if the robot ever observes itself performing the same action as the human. For subject 3 in phase 2 and subject 7 in phases 1 and 2, the maximum probability test is the opposite of what was expected and desired. In order to understand the reason for this abnormality, it is appropriate to examine the responses to third survey question. Subject 7 provided the following response to question 3 in phase 2 of the experiment:

"I tried to do the opposite motion than the robot at the start, but find myself being in consistent [sic] after trying to mimic it correctly in phase 1. overall I feel as though I had more control over it's actions this time, meaning that when I nodded it nodded but I wanted it to not do my action, so for this phase the robot did not understand my intent."

This response from subject 7 indicates a breakdown in the performance of the human in the signal grounding process. This may be due in part to habituation from the behavior pattern in the imitation phase, or to an innate tendency to mimic behavior.

It is also worth noting that while some of the most likely tests for a given history are the same for some subjects across the two phases, the most likely overall tests (bolded) are more important for the interaction, because some of the histories may be unlikely to occur given the intent of the human. For example, in the non-imitation condition $[0, 0]$ is unlikely to occur, because the human is not likely to respond to imitation with further imitation. In this case, $[1, 1]$ or $[0, 1]$ would be more likely prefixes, upon which the the robot would have to select future tests.

D. Discussion

The human-robot interactions show some interesting results. First it is clear that the imitation cases were very successful in creating a rewarding experience for humans interacting with the robot (people like it when they are successful at their task, which was to make the robot understand them). The outcome of the non-imitation case is not as clear, but in both cases the robot learned significantly different representations, and generated significantly different behavior, and for the most part this appears to have been successful at engaging human partners. Of course this is a small, proof-of-concept study with few participants, but the results are encouraging for the development of social robotics algorithms that attempt to ground communication in basic learning of the expected outcomes of actions.

One of the reasons for the success of the "exploitation" action selection policy of selecting actions for which the robot has high certainty of the response is undoubtedly because a human is attempting to have an interesting interaction with the robot. Consider the imitation and non-imitation cases in a little more detail. In the imitation case, if the robot learns that imitating will lead to further imitation, this reinforces the behavior, leading to a "good" equilibrium where the robot has understood the human's intent and can act in accordance with that intent. In the non-imitation case, if the robot starts to imitate the human's actions, the human will learn to change her actions because they are not eliciting the desired behavior. This will prevent the imitation equilibrium from emerging. This is confirmed by the evidence in Table II – while the most likely tests for some histories are the same across the two phases for many of the subjects, the probability of that history actually occurring (which is dependent on the human's actions) will be different in the two cases, which is why (1) the robot manages in general to perform in accord with the human's intent, and (2) the single most likely test to succeed is very different in the two phases. In some ways the algorithm is taking advantage of the fact that the human takes initiative in the interaction in order to

learn appropriate behavior. This is an interesting example of parent-child or master-apprentice learning.

V. CONCLUSIONS

This paper marries ideas from emergent theories of cognition with the recently developed predictive state representation (PSR) framework for reinforcement learning in designing an effective algorithm for social human-robot interaction in a prototype shadow puppetry domain. We demonstrate that PSRs can accurately capture the dynamics of human-human shadow puppet interaction, and then combine the PSR representation with an action selection strategy based on the idea that social interactions develop when participants are able to seek and convey understanding. The resulting algorithm is successful in two ways: (1) when we give a human a particular task ("get the robot to imitate / not imitate you"), the PSR representation learns an appropriate encoding of the human's intent in an on-line fashion, and (2) the robot in general generates behaviors that the human thinks are appropriate responses in the social interaction.

An important aspect of this work is that the algorithm learns on-line. The ability to examine data beforehand and learn about a human is a convenience that a situated agent might not have. For practitioners looking to use PSRs for different tasks, the scope of the initial representation (i.e. the initial set of core tests) is particularly important. The trade-off between the robustness of a large set of core tests and the efficiency of a small set is apparent. Another issue is the problem of enabling the agent to deal more directly with raw signals, rather than providing it with a predefined set of gestural primitives.

In generalizing this approach, these problems are not insurmountable. First, there are many possible granularities between this and a completely out-of-the-box PSR model. The level of expert design used in the system can be though of as a sliding scale. Second, the off-policy nature of PSRs means that they can be run multiply and in parallel. A PSR need not select an action to learn from its outcome. Third, actions must be recognized and selected in real-time, but reasoning about history can be done in background. This means that it may be possible for an agent to start with simple, decoupled models and incrementally combine or extend them. For example, the agent may start with a small set of short tests, and extend the important ones.

In retrospect, for the particular task we focus on in this paper, interaction with humans based on patterns of imitative / non-imitative behavior, an internal model based on Markovian assumptions (like a POMDP or HMM) could have performed well in terms of prediction accuracy on the human-human tests, and provided a basis for a control algorithm for the robot. However, the goal of this project was to provide an algorithm that did not require pre-specifying a type of model and a state space. The Markov model of Figure 2 is based on an after-the-fact human analysis of data from the human-human experiments. Whether an automated learning process could have learned that these were the relevant states and extracted appropriate transition probabilities from the data is an interesting question, but it is not our focus in this work. Instead, we demonstrate that it is possible to learn how to interact successfully without the need to rely on the existence and learnability of state space models (although we reiterate that the possible "true" model of interaction ends up being simple in this case, and much more research is needed to determine if PSRs will be successful when the underlying model is more complex).

Another important aspect of this work is that the PSRs can be learned online. The research reported here is one of the first real-world successes of a PSR algorithm that we are aware of. We believe PSRs are a promising approach to several problems for which specifying a useful model of the world is hard. While the deployment described here did involve significant engineering in choosing exploration schedules, deciding when to perform discovery, and so on, it is clear that these algorithms can be made to work on real robots who have to act and learn in an uncertain world.

REFERENCES

[1] A. Y. Ng, H. J. Kim, M. I. Jordan, and S. Sastry, "Inverted autonomous helicopter flight via reinforcement learning," in *Int. Symp. Experimental Robotics*. MIT Press, 2004.

[2] S. Thrun and et al., "Stanley: The robot that won the darpa grand challenge: Research articles," *J. Robot. Syst.*, vol. 23, no. 9, pp. 661–692, 2006.

[3] A. Tapus, M. J. Matarìc, and B. Scassellati, "The grand challenges in socially assistive robotics," *IEEE RAM, Special Issue on Grand Challenges in Robotics*, vol. 14, no. 1, 2007.

[4] M. L. Littman, R. S. Sutton, and S. Singh, "Predictive representations of state," in *NIPS 14*. MIT Press, 2002, pp. 1555–1561.

[5] P. McCracken and M. H. Bowling, "Online discovery and learning of predictive state representations," in *NIPS*, 2005.

[6] F. J. Bernieri, J. S. Reznick, and R. Rosenthal, "Synchrony, pseudosynchrony, and dissynchrony: Measuring the entrainment process in mother-infant interactions," *J. Personality and Social Psychology*, vol. 54, no. 2, pp. 243–253, 1988.

[7] G. Semin, "Grounding communication: Synchrony," *Social Psychology: Handbook of Basic Principles 2nd Edition*, pp. 630–649, 2007.

[8] E. Meisner, S. Sabanovic, V. Isler, L. R. Caporael, and J. Trinkle, "Shadowplay: A generative model for nonverbal human-robot interaction," in *4th ACM/IEEE Int. Conf. on Human-Robot Interaction*, 2009.

[9] E. Goffman, *The Presentation of Self in Everyday Life*. Anchor Books, 1959.

[10] K. Hsiao, L. Kaelbling, and T. Lozano-Perez, "Grasping pomdps," in *IEEE ICRA*, April 2007, pp. 4685–4692.

[11] J. Pineau, M. Montemerlo, M. Pollack, N. Roy, and S. Thrun, "Probabilistic control of human robot interaction: Experiments with a robotic assistant for nursing homes," in *Joint Workshop on Technical Challenges for Robots in Human Environments*, October 2002.

[12] F. Broz, I. R. Nourbakhsh, and R. G. Simmons, "Planning for human-robot interaction using time-state aggregated pomdps." in *AAAI*, D. Fox and C. P. Gomes, Eds. AAAI Press, 2008, pp. 1339–1344.

[13] J. Boger, J. Hoey, P. Poupart, C. Boutilier, G. Fernie, and A. Mihailidis, "A planning system based on markov decision processes to guide people with dementia through activities of daily living," *IEEE Transactions on Information Technology and Biomedicine*, vol. 10, no. 2, pp. 323–333, April 2006.

[14] S. Singh, M. R. James, and M. R. Rudary, "Predictive state representations: a new theory for modeling dynamical systems," in *AUAI '04: Proceedings of the 20th conference on Uncertainty in artificial intelligence*. Arlington, Virginia, United States: AUAI Press, 2004, pp. 512–519.

[15] S. Singh, M. L. Littman, N. K. Jong, D. Pardoe, and P. Stone, "Learning predictive state representations," in *In Proceedings of the Twentieth International Conference on Machine Learning*, 2003, pp. 712–719.