Institute for Software Research

School of Computer Science

1-1-2007

Learning to Predict the Effects of Actions: Synergy Between Rules and Landmarks

Jonathan Mugan
Carnegie Mellon University, jmugan@andrew.cmu.edu

Benjamin Kuipers

Recommended Citation

Mugan, Jonathan and Kuipers, Benjamin, "Learning to Predict the Effects of Actions: Synergy Between Rules and Landmarks" (2007). *Institute for Software Research.* Paper 749.
http://repository.cmu.edu/isr/749

Learning to Predict the Effects of Actions: Synergy between Rules and Landmarks

Jonathan Mugan
Computer Science Department
University of Texas at Austin
Austin Texas, 78712 USA

Benjamin Kuipers
Computer Science Department
University of Texas at Austin
Austin Texas, 78712 USA

***Abstract*— A developing agent must learn the structure of its world, beginning with its sensorimotor world. It learns rules to predict how its motor signals change the sensory input it receives. It learns the limits to its motion. It learns which effects of its actions are unconditional and which effects are conditional, including what they depend on. We present preliminary results evaluating an implemented computational model of this important kind of foundational developmental learning. Our model demonstrates synergy between the learning of landmarks representing important qualitative distinctions, and the learning of rules that exploit those distinctions to make reliable predictions. These qualitative distinctions make it possible to define discrete events, and then to identify predictive rules describing regularities among events and the values of context variables. The attention of the learning agent is focused by a stratified model that structures the set of variables, and the structure of the stratified model is simultaneously created by the learning process.**

***Index Terms*— developmental learning, sensorimotor learning, qualitative abstraction, predictive rules, landmark values**

I. Introduction

A developing agent — human or robot — must learn the structure of its world, beginning with its own sensorimotor interaction. Even in an unknown environment, with uninterpreted sensors and effectors — what William James [1] described as “blooming, buzzing confusion” — it must learn to predict how its motor signals change the sensory input it receives. It learns rules to make predictions. It learns the limits to its motion. It learns which effects of its actions are unconditional and which effects are conditional, including what they depend on. And so on.

We present early results with an implemented computational model that shows how this learning process could be structured. Important qualitative distinctions are identified, and represented as *landmarks* in the ranges of continuous variables. The landmarks allow rules to be learned to capture significant regularities. Non-determinism in these rules allows attention to be focussed on regions where new landmarks can be learned to improve the predictions. And the new landmarks, in turn, allow yet more rules to be learned.

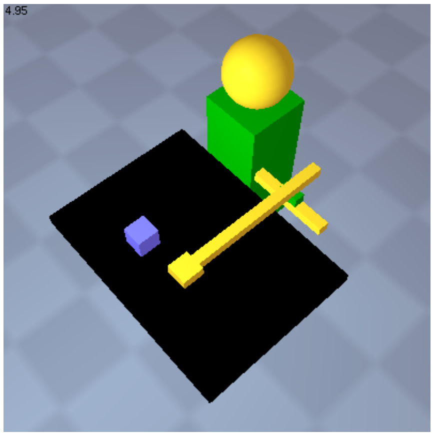

Fig. 1. A simulated “robot baby” is implemented in Breve [2]. It has a torso with a 2-dof arm and is sitting in front of a tray with a block. The robot has two motor variables $\tilde{u}_x$ and $\tilde{u}_y$ that move the hand in the x and y directions, respectively. The perceptual system creates variables for each of the two tracked objects in this environment: the hand and the block. The variables corresponding to the hand are $\tilde{h}_x(t)$, $\tilde{h}_y(t)$, and $h_a(t)$. The continuous variables $\tilde{h}_x(t)$, $\tilde{h}_y(t)$ represent the location of the hand in the x and y directions, respectively, and the Boolean variable $h_a(t)$ represents whether the hand is in view. The variables corresponding to the block are $\tilde{b}_x(t)$, $\tilde{b}_y(t)$, and $b_a(t)$ and they have the same respective meanings as the variables for the hand. When two objects are sufficiently close together the perceptual system also creates variables to represent their relationship. The relationship between the hand and the block is represented by the continuous variables $\tilde{c}_x(t)$, $\tilde{c}_y(t)$, and $\tilde{e}(t)$. The variables $\tilde{c}_x(t)$ and $\tilde{c}_y(t)$ represent the coordinates of the center of the hand in the frame of reference of the center of the block, and the variable $\tilde{e}(t)$ represents the distance between the hand and the block. The values of all variables are updated by perceptual trackers at each timestep as the object moves. If the block disappears from view by falling off the tray, then $b_a(t) = \mathsf{false}$.

This learning process is one stage in a longer developmental learning sequence. We build on existing methods for learning the structure of the sensorimotor system by analyzing the statistical properties of disorganized elements (“pixels”) of the agent’s sensory and motor vectors [3]–[5]. We also assume that the agent has already learned to

This work has taken place in the Intelligent Robotics Lab at the Artificial Intelligence Laboratory, The University of Texas at Austin. Research of the Intelligent Robotics lab is supported in part by grants from the National Science Foundation (IIS-0413257 and IIS-0538927), from the National Institutes of Health (EY016089), and by an IBM Faculty Research Award. We would also like to acknowledge NSF grant EIA-0303609.

individuate, track, and describe perceived objects in its visual field, using the methods of Modayil and Kuipers [6], [7].

The agent's only task is to learn progressively more reliable rules to predict the effects of its actions on the objects in its environment. By observing the success or failure of these rules the unsupervised learning agent generates its own supervisory signal.

Learning identifies predictive rules and adds contextual conditions to increase reliability. In order to learn rules robustly, the agent builds a stratified model, where the variables it has characterized are organized into strata. It starts with the built-in motor variables in the lowest stratum, and then creates higher strata as determined by the rules it learns. The stratified model focuses the attention of the learning agent, so it can proceed greedily, building rules with antecedent events involving variables already in the model. Meanwhile, it infers a qualitative abstraction for the range of each continuous variable, describing values in the range in terms of landmark values, open intervals between landmark values, and directions of change. An important part of learning is to identify landmarks representing distinctions that make the rules more reliable.

We evaluate the learning algorithm with a simulated "robot baby" (Fig. 1). The agent's only task is to learn to predict future states in its environment. It learns how much force is necessary to move its hand. It learns the limits of its hand motion, and how to reliably move its hand within those limits. It also learns reliable rules that in some situations allow it to move its hand towards the block, and then to move the block. These are essential first steps toward grasping and manipulation of objects in its world.

The significance of this learning scenario is that the agent learns the qualitative distinctions that allow reliable and informative rules to be learned. The rules learned from these distinctions then serve as a supervisory signal because the agent can observe when they do and do not make correct predictions. The agent can then use this supervisory signal to make further distinctions, which in turn can lead to more predictive rules. All learning takes place within a stratified model that serves as a focus of attention and a mechanism to reduce the number of rules learned.

We will first introduce the theory behind our method. We then provide an evaluation, describing what the agent learns in the simulated environment described in Fig. 1. Finally, we will discuss related work and future directions of research.

II. Knowledge Representation and Learning

A cognitive learning agent (the "robot baby") has access to a number of continuous variables representing the interfaces to sensory and motor processes associated with its body. In our model, the learning agent can observe the continuous values of its variables, and can collect certain limited kinds of statistics, but it can only *represent* (i.e., store, match, and retrieve) and do logical and causal *inference* with discrete variables.

Therefore, a critical task for the "robot baby" is to learn appropriate abstractions from continuous to discrete variables. Initially, the values of the continuous variables are completely meaningless. Our goal is for the agent to learn, from its own undirected experience, to identify *landmark values* that make important qualitative distinctions for each variable. The importance of a qualitative distinction is estimated from the reliability of the rules that can be learned, given that distinction.

The qualitative representation is based on QSIM [8]. For each continuous variable $\tilde{x}(t)$ two discrete variables are created: a discrete variable $x(t)$ that represents the magnitude of $\tilde{x}(t)$, and a discrete variable $\dot{x}(t)$ that represents the direction of change of $\tilde{x}(t)$. (Non-zero directions of change that persist fewer than three time-steps are filtered out.)

A continuous variable $\tilde{x}(t)$ ranges over some subset of the real number line $(-\infty, +\infty)$. In QSIM, its magnitude is abstracted to a discrete variable $x(t)$ that ranges over a *quantity space* $Q(x)$ of qualitative values. $Q(x) = L(x) \cup I(x)$, where $L(x) = \{x_1, \cdots x_n\}$ is a totally ordered set of landmark values, and $I(x) = \{(-\infty, x_1), (x_1, x_2), \cdots (x_n, +\infty)\}$ is the set of mutually disjoint open intervals in the real number line that $L(x)$ defines. A discrete variable $\dot{x}(t)$ representing the direction of change of $\tilde{x}(t)$ has a single intrinsic landmark at 0, so its initial quantity space is $Q(\dot{x}) = \{(-\infty, 0), 0, (0, +\infty)\}$. (Note that, for most magnitude variables, zero is just another point on the number line, so those variables initially have no landmarks.)

Initially, when the agent knows of no meaningful qualitative distinctions among values for $\tilde{x}(t)$, we describe the quantity space as the empty list of landmarks, (). A quantity space with two landmarks might be described by (x_1, x_2), which implies five distinct qualitative values, $Q(x) = \{(-\infty, x_1), x_1, (x_1, x_2), x_2, (x_2, +\infty)\}$. Table I and Fig. 1 give examples of the variables the learning agent knows about, and their initial and final quantity spaces.

A. Events

An *event* $E(t)$ is a qualitative change in the state of a system that takes place at time-point t.

A *transition* $A_t \rightarrow a$ is an event defined by $A(t-1) \neq a$ and $A(t) = a$. That is, a discrete variable A changes to value a at time t, from some other value.

The only events we will consider here are transitions.

Our goal is to describe regularities in the occurrence of events. These regularities are expressed as *predictive rules*. There are two types of predictive rules: *causal rules* represent that one event occurs after another later in time, the linking of events appearing as causal to the agent; and *functional rules* represent that two events are linked by a function and so happen at the same time. These two rules differ only in the time component, so after initially discussing them separately, we will simply refer to both types as rules.

B. Causal Rules

A *causal rule* r has the form $\langle \mathcal{C} : E_1 \Rightarrow E_2 \rangle$, where $E_1(t)$ is one event, say $A_t \rightarrow a$, $E_2(t')$ is another event over a direction of change variable, say $B_{t'} \rightarrow b$, that takes place relatively soon after t, and the *context* $\mathcal{C}$ is a set that can consist of magnitude or Boolean variables.

We say that a causal rule $r = \langle \mathcal{C} : E_1 \Rightarrow E_2 \rangle$ *applies* at time t when $E_1(t)$ is true. That E_2 takes place "relatively soon after" $E_1(t)$ is formalized in terms of a time-delay k.

$$soon(t, E_2) \quad \equiv \quad \exists t' \, [t < t' < t + k \ \wedge \ E_2(t')] \qquad (1)$$

A rule $r = \langle \mathcal{C} : E_1 \Rightarrow E_2 \rangle$ *succeeds* when:

$$succeeds(r, t) \quad \equiv \quad E_1(t) \ \wedge \ soon(t, E_2) \qquad (2)$$

Associated with a causal rule $r = \langle \mathcal{C} : E_1 \Rightarrow E_2 \rangle$ is a probability distribution of the form

$$P(soon(t, E_2) | E_1(t) = \mathsf{true}, \mathcal{C}(t-1)) \qquad (3)$$

which is the conditional probability distribution over the binary random variable $soon(t, E_2)$, given that $E_1(t)$ is true, conditioned on the values of the variables in $\mathcal{C}$ at time $t-1$.

We define the *entropy* of a rule $r = \langle \mathcal{C} : E_1 \Rightarrow E_2 \rangle$ as the conditional entropy of $soon(t, E_2)$ given $\mathcal{C}(t-1)$, with the added restriction that event $E_1(t)$ occurs. In equation form it is given by

$$H(r) \quad = \quad H(soon(t, E_2) | E_1(t) = \mathsf{true}, \mathcal{C}(t-1)). \quad (4)$$

We are interested in learning rules that predict events, but rule r can have low entropy if it predicts that E_2 will almost never follow E_1, so a rule must have more than low entropy to be useful. We define a concept called *best reliability* represented as $brel(r)$. For rule r, $brel(r)$ is the highest probability of success for any value of $\mathcal{C}$. If $\mathcal{C} = \emptyset$ then $brel(r)$ is just the probability of success of r.

1) Learning a causal rule: The learner starts by searching for two events E_1 and E_2 such that observing event E_1 means that event E_2 is significantly more likely to occur than it would have been otherwise.

The learner asserts an initial rule $\langle \emptyset : E_1 \Rightarrow E_2 \rangle$ with empty context, when $P(soon(t, E_2) | E_1(t)) > 0.1$ and

$$\iota(P(soon(t, E_2) | E_1(t)), P(soon(t, E_2))) > \theta_a > 1 \qquad (5)$$

where the function on probabilities $\iota(p, q) = \frac{p}{q} \cdot \frac{1-q}{1-p}$ has been defined to have higher resolution near the extremes, and lower resolution near the center, over the interval $(0, 1)$ of probability values. The parameter $\theta_a = 2.5$ specifies how much more likely E_2 should be, after E_1 has been observed. (Here and elsewhere we require a minimum number of relevant observations so the probability will be reliable.)

2) Learning a context for a functional rule: Once the agent has learned a rule $r = \langle \emptyset : E_1 \Rightarrow E_2 \rangle$, it searches for a magnitude or Boolean variable v_1 such that if r is modified to be $r' = \langle \{v_1\} : E_1 \Rightarrow E_2 \rangle$ the variable v_1 provides sufficient information gain $H(r) - H(r') > \theta_{ig}$. The parameter $\theta_{ig} = 0.15$ determines how much information gain is required to augment the context. If there are multiple discrete variables that meet this criteria, then the one providing the largest information gain is chosen.

Once it has learned a rule $r' = \langle \{v_1\} : E_1 \Rightarrow E_2 \rangle$ the agent searches for another magnitude or Boolean variable v_2 such that if r' is modified to be $r'' = \langle \{v_1, v_2\} : E_1 \Rightarrow E_2 \rangle$ the variable v_2 provides sufficient information gain $H(r') - H(r'') > \theta_{ig}$. In principle, an arbitrarily large context can be learned, but in our current implementation, the size is limited to 2.

This approach is inspired by the concept of *marginal attribution* from Drescher's schema mechanism [9]. However, unlike Drescher and others, we reason with the entire set of possible values of the set $\mathcal{C}$ of context variables, rather than simply asserting a condition $V = v_j$ into the antecedent of a particular rule.

C. Functional Rules

A *functional rule* $r = \langle \mathcal{C} : E_1 \Rightarrow E_2 \rangle$ has the same form as a causal rule, and behaves in a similar way, with three exceptions. The first difference is in the timing of the events: the predicate $soon(t, E_2)$ is replaced with $E_2(t)$, which means that the events E_1 and E_2 must happen in the same timestep. The second difference is that functional rules are only learned on events over direction of change variables. And the third difference is because there is no time delay. If a functional rule $\langle \mathcal{C} : E_1 \Rightarrow E_2 \rangle$ is learned but its opposite $\langle \mathcal{C} : E_2 \Rightarrow E_1 \rangle$ has a significantly higher rate of success before the context is considered, then $\langle \mathcal{C} : E_1 \Rightarrow E_2 \rangle$ is replaced by $\langle \mathcal{C} : E_2 \Rightarrow E_1 \rangle$ (although $\langle \mathcal{C} : E_1 \Rightarrow E_2 \rangle$ is still used to build the stratification process as discussed in SectionII-E). We will refer to both types of rules as predictive rules or rules.

D. Landmarks for Predictive Rules

Inserting a new landmark x^* into an open interval (x_i, x_{i+1}) allows it to be replaced in $Q(x)$ by two intervals and the dividing landmark: (x_i, x^*), x^*, (x^*, x_{i+1}). Adding this new landmark into the quantity space $Q(x)$ allows a new distinction to be made that may transform a rule r into a new rule r'.

If a landmark candidate for a rule $r = \langle \mathcal{C} : A \rightarrow b \Rightarrow B \rightarrow b \rangle$ is on variable A, then to be adopted the landmark must increase the best reliability of r by transforming it into r' so that $\iota(brel(r'), brel(r)) > \theta_a$.

If the landmark candidate for r is on a variable other than A, it must improve the entropy of r to be adopted. To do this it must modify an existing variable in $\mathcal{C}$ or a variable outside of $\mathcal{C}$ that can then be added to $\mathcal{C}$ so that r becomes more deterministic by transforming it into r' where $H(r) - H(r') > \theta_{ig}$.

Landmark candidates are chosen considering the width of the interval and the highest gain [10] with respect to the success of the rule. Depending on the relative gains of nearby potential values for a new landmark x^*, this search can result in either a precise numerical value, or a range of possible values for x^* on different occasions: $range(x^*) = [lb, ub]$. Examples of both cases are shown in Table I.

E. Learning the Stratified Model

The learning process organizes the discrete variables into a stratified model, which serves as a focus of attention and helps contain the proliferation of rules.

The learning agent starts with a set of continuous variables. The motor variables $\{\tilde{u}_x, \tilde{u}_y\}$ control the hand position. Perceptual trackers deliver groups of continuous variables corresponding to the hand $\{\tilde{h}_x, \tilde{h}_y, h_a\}$, the block $\{\tilde{b}_x, \tilde{b}_y, b_a\}$, and the relation between the two $\{\tilde{c}_x, \tilde{c}_y, \tilde{e}\}$. (See Figure 1 for a description of these variables.)

A stratum $\mathcal{S}_i$ in the model at level i is a set of discrete variables. A variable may be in at most one stratum and a derivative variable $\dot{x}$ is always included in the same stratum as the variable x. The first stratum $\mathcal{S}_0$ is initialized with the qualitative motor variables $\{u_x, u_y\}$, which have initial intrinsic landmarks at 0. The remaining unstratified discrete variables are available for rule-building.

From the current highest (or outermost) stratum $\mathcal{S}_i$ and each stratum below, the learning agent then repeatedly applies the predictive rule and landmark learning methods until no new rules or landmarks are generated. Once a new stratum $\mathcal{S}_{i+1}$ is defined, the set of rules is pruned, and the process repeats to build the next stratum.

Where $\mathcal{S}_i$ is the current highest stratum, the learning algorithm attempts to learn rules $r = \langle \mathcal{C} : A{\rightarrow}a \Rightarrow B{\rightarrow}b \rangle$ starting with antecedent variables A in $\mathcal{S}_j$ where $j \leq i$. When searching for a suitable result event $B \rightarrow b$, it restricts its attention to variables in $\mathcal{S}_j$, variables in $\mathcal{S}_{j+1}$, and variables so far unstratified. A previously unstratified variable that becomes the result variable B of a rule $r = \langle \mathcal{C} : A{\rightarrow}a \Rightarrow B{\rightarrow}b \rangle$ where $A \in \mathcal{S}_j$ is added to $\mathcal{S}_{j+1}$. And the other member of a variable-derivative pair, $\langle x, \dot{x} \rangle$ is also added to $\mathcal{S}_{j+1}$.

An exception is a functional rule $\langle \mathcal{C} : E_2 \Rightarrow E_1 \rangle$ that is learned because it has a higher rate of success than $\langle \mathcal{C} : E_1 \Rightarrow E_2 \rangle$. In this case for the purpose of stratification, the rule $\langle \mathcal{C} : E_2 \Rightarrow E_1 \rangle$ is treated as if it were of the form $\langle \mathcal{C} : E_1 \Rightarrow E_2 \rangle$.

After a stratum $\mathcal{S}_i$ is learned, the algorithm prunes the model. Rules that have not become sufficiently deterministic are pruned, where how deterministic a rule $r = \langle \mathcal{C} : A{\rightarrow}a \Rightarrow B{\rightarrow}b \rangle$ must be depends on the number n of variables in $\mathcal{C}$. Rule r is removed if $H(r) \geq 0.6 - (0.05n \times \theta_{ig})$. Rule r is also removed if $brel(r) \geq 0.6$. Of the remaining rules, if there exist rules $\langle E_1 \Rightarrow E_2 \rangle$, $\langle E_2 \Rightarrow E_3 \rangle$, and $\langle E_1 \Rightarrow E_3 \rangle$, then $\langle E_1 \Rightarrow E_3 \rangle$ is redundant and can be pruned.

TABLE I

VARIABLES, THEIR RANGES OF VALUES, AND INITIAL AND FINAL LANDMARKS

Var.	Range	Initial	Final	Landmarks
u_x	$[-500, 500]$	(0)	$(L_1, 0, L_2)$	$L_1 = -302.80$
u_y	$[-500, 500]$	(0)	$(L_3, 0, L_4)$	$L_2 = 302.23$
h_x	$[-2.0, 2.0]$	$()$	(L_5, L_6)	$L_3 = -305.10$
$\dot{h}_x$	$(-\infty, +\infty)$	(0)	(0)	$L_4 = 302.50$
h_y	$[-2.0, 2.0]$	$()$	(L_7, L_8)	$L_5 = [-2.03, -1.99]$
$\dot{h}_y$	$(-\infty, +\infty)$	(0)	(0)	$L_6 = [1.99, 2.04]$
h_a	$\{0, 1\}$	Binary	Binary	$L_7 = [-2.01, -1.99]$
b_x	$(-\infty, +\infty)$	$()$	$()$	$L_8 = [1.99, 2.01]$
$\dot{b}_x$	$(-\infty, +\infty)$	(0)	(0)	$L_9 = -2.16$
b_y	$(-\infty, +\infty)$	$()$	$()$	$L_{10} = 2.00$
$\dot{b}_y$	$(-\infty, +\infty)$	(0)	(0)	$L_{11} = -1.68$
b_a	$\{0, 1\}$	Binary	Binary	$L_{12} = 0.47$
c_x	$(-\infty, +\infty)$	$()$	(L_9, L_{10})	
$\dot{c}_x$	$(-\infty, +\infty)$	(0)	(0)	
c_y	$(-\infty, +\infty)$	$()$	(L_{11})	
$\dot{c}_y$	$(-\infty, +\infty)$	(0)	(0)	
e	$[0, +\infty)$	$()$	(L_{12})	
$\dot{e}$	$(-\infty, +\infty)$	(0)	(0)	

If the pruning operation eliminates all rules pointing to variables in some variable-derivative pair $\langle x, \dot{x} \rangle$ in $\mathcal{S}_i$, then those variables are removed from $\mathcal{S}_i$.

III. EVALUATION

A. Simulation Scenario

The agent executed the learning algorithm on the simulation shown in Fig. 1. The agent's experience consisted of 405,000 timesteps where each timestep corresponded to 0.05 seconds. The agent alternated between gathering statistics while experiencing the world and learning new rules and landmarks. During each period of experience the agent began with its motor variables set to 0 and would then motor babble for 15,000 timesteps. During this time, if the block fell off of the tray it was immediately moved to a random position within reach of the agent. Since there are parts of the tray that cannot be reached by the agent, if at any point the block was not moved for 300 timesteps it was moved to a random location within reach of the agent.

As the agent experiences the world it uses its current set of landmarks to convert the continuous variables described in the caption of Fig. 1 into discrete variables. These variables include the variables u_x and u_y for controlling the hand, the variables h_x, h_y, $\dot{h}_x$, $\dot{h}_y$, and h_a that give the state of the hand, the variables b_x, b_y, $\dot{b}_x$, $\dot{b}_y$, and b_a that give the state of the block. And finally, the relation between the hand and the block is given by c_x, c_y, $\dot{c}_x$, $\dot{c}_y$, and e.

For each variable, Table I provides the physical range of values it can take on, the initial and final sets of landmarks, and the numerical value or range representing the agent's knowledge of the value of each landmark.

TABLE II

Subset of Learned Rules (T indicates either causal rule or functional rule)

Strata	T	Rule
$\mathcal{S}_0$= $\{u_x, u_y\}$	C	$\{h_x\} : u_x \rightarrow (-\infty, -302.80) \Rightarrow \dot{h}_x \rightarrow (-\infty, 0)$
	C	$\{h_x\} : u_x \rightarrow (302.23, +\infty) \Rightarrow \dot{h}_x \rightarrow (0, +\infty)$
	C	$\{h_y\} : u_y \rightarrow (-\infty, -305.10) \Rightarrow \dot{h}_y \rightarrow (-\infty, 0)$
	C	$\{h_y\} : u_y \rightarrow (302.50, +\infty) \Rightarrow \dot{h}_y \rightarrow (0, +\infty)$
$\mathcal{S}_1$= $\{h_x, \dot{h}_x, h_y, \dot{h}_y\}$	C	$\emptyset : h_x \rightarrow [-2.03, -1.99] \Rightarrow \dot{h}_x \rightarrow [0]$
	F	$\emptyset : \dot{h}_x \rightarrow (-\infty, 0) \Rightarrow \dot{c}_x \rightarrow (-\infty, 0)$
	F	$\emptyset : \dot{h}_y \rightarrow [0] \Rightarrow \dot{c}_y \rightarrow [0]$
	F	$\emptyset : \dot{h}_y \rightarrow (0, +\infty) \Rightarrow \dot{c}_y \rightarrow (0, +\infty)$
$\mathcal{S}_2$= $\{c_x, \dot{c}_x, c_y, \dot{c}_y, e, \dot{e}\}$	F	$\{c_x\} : \dot{c}_x \rightarrow (-\infty, 0) \Rightarrow \dot{e} \rightarrow (-\infty, 0)$
	F	$\{c_x\} : \dot{c}_x \rightarrow (0, +\infty) \Rightarrow \dot{e} \rightarrow (-\infty, 0)$
	F	$\{c_x, c_y\} : \dot{c}_y \rightarrow (-\infty, 0) \Rightarrow \dot{e} \rightarrow (0, +\infty)$
	C	$\{c_x\} : e \rightarrow (0, 0.47) \Rightarrow \dot{b}_x \rightarrow (-\infty, 0)$
$\mathcal{S}_3$= $\{b_x, \dot{b}_x\}$		empty

B. Learning Example

This section describes a particular case of rule and landmark learning that is typical of the learning agent's behavior.

(1) By noting that the event $\dot{h}_x \rightarrow (-\infty, 0)$ of the hand moving to the left is more likely to occur following the event $u_x \rightarrow (-\infty, 0)$ of force being applied to the left, the agent hypothesizes a causal rule linking $u_x \rightarrow (-\infty, 0)$ to $\dot{h}_x \rightarrow (-\infty, 0)$. It then creates the rule

$$r = \langle \emptyset : u_x \rightarrow (-\infty, 0) \Rightarrow \dot{h}_x \rightarrow (-\infty, 0) \rangle$$

with a best reliability $brel(r) = 0.36$ and an entropy $H(r) = 0.94$.

(2) To try to make this rule more reliable, the agent looks at the value of u_x each time event $u_x \rightarrow (-\infty, 0)$ occurs. It finds that the best reliability of r can be increased by creating a landmark $L_1 = -302.80$ (in the simulator it takes a force of 300 to move the hand). It then updates r to be

$$r' = \langle \emptyset : u_x \rightarrow (-\infty, L_1) \Rightarrow \dot{h}_x \rightarrow (-\infty, 0) \rangle$$

with best reliability $brel(r') = 0.74$ and an entropy $H(r') = 0.58$.

(3) The agent cannot move the hand farther to the left if it is already at its leftmost point. The agent finds that if it adds a landmark $L_5 = [-2.12, -1.95]$ for h_x that it can add h_x to the context of r' creating a new rule

$$r'' = \langle \{h_x\} : u_x \rightarrow (-\infty, L_1) \Rightarrow \dot{h}_x \rightarrow (-\infty, 0) \rangle$$

a best reliability of $brel(r'') = 0.90$ with entropy $H(r'') = 0.39$. The rule r'' is now deterministic enough to make useful predictions. The initial failure rate of 0.10 comes from having experienced relatively few timesteps and the stochastic property of motor babbling. When $u_x = (-\infty, L_1)$ is selected for a small number of timesteps and it is a value close to L_1, then it may not be enough to elicit the event $\dot{h}_x \rightarrow (-\infty, 0)$. But even in this noisy environment, the rule can be learned.

(4) Landmark $L_6 = [1.99, 2.04]$ is learned in the same way using movement in the positive x direction. With the addition of L_6, r'' is the first rule seen in Table II. If we let $E_1 = u_x(t) \rightarrow (-\infty, L_1)$ and $E_2 = \dot{h}_x(t) \rightarrow (-\infty, 0)$ then at the end of learning the rule r'' represents the set of probabilities

$$\begin{aligned} P(soon(t, E_2) \mid E_1, h_x(t) = L_5) &= 0.01 \\ P(soon(t, E_2) \mid E_1, h_x(t) = (L_5, L_6)) &= 0.93 \\ P(soon(t, E_2) \mid E_1, h_x(t) = L_6) &= 0.95 \end{aligned}$$

(5) Landmark L_5 also allows the agent to create a new rule. Since L_5 represents the limit of movement in the negative x direction, it can be used to predict when the hand will stop using the rule $\langle \emptyset : h_x \rightarrow L_7 \Rightarrow \dot{h}_x \rightarrow 0 \rangle$, with best reliability 0.96 and entropy 0.23.

C. What the Agent Has Learned

The agent learns a set of landmarks and a stratified structure on the set of discrete variables (Table I). It also learns a total of 33 rules: 14 causal rules and 19 functional rules (a sample of these appears in Table II). The numbers of rules at levels $\mathcal{S}_0$, $\mathcal{S}_1$, $\mathcal{S}_2$, and $\mathcal{S}_3$, is 5, 15, 13, and 0, respectively.

Recall that, initially, only the motor and direction-of-change variables included an initial landmark at 0, and that magnitude variables had no landmarks at all, so that any qualitative distinctions among values of those variables had to be learned by the agent from its own experience.

By using the learning algorithm, the agent learns that the motor variables directly control the motion of the hand, and that a motor value greater than about 300 is necessary to produce motion (4 rules in $\mathcal{S}_0$). It learns that motion of the hand directly affects the relational variables between the hand and the block (5 rules in $\mathcal{S}_1$).

It learns that each hand position variable is restricted to the approximate range of $[-2.0, +2.0]$ in both the x and the y direction. It also learns that it cannot move the hand further in any direction if it is at the boundary in that direction (4 rules in $\mathcal{S}_1$).

It learns a large variety of rules (in $\mathcal{S}_2$ and $\mathcal{S}_3$) describing the relationships among (a) the position of the hand in the frame of reference of the block, (b) the distance between the hand and the block, and (c) the motion of the block. It learns a landmark on the distance e between the hand and the block and then learns that if the distance between the hand and the block reduces to less than 0.47 that the block may move.

The agent also learns important landmarks on the variables c_x and c_y. The hand and the block each have a diameter of 2.0, so if they are next to each other $\tilde{c}_x$ will have a value of 2.0 or -2.0 depending on which side the block is on, and this is represented by landmarks L_9 and L_{10}. When the hand is under and touching the block $\tilde{c}_y = -2.0$, and this value is approximated by landmark L_{11}.

Not every learned rule makes sense from the point of view of a human observer. Nor is every rule learned that a human observer would consider justified. In some cases, deeper analysis shows that the learning agent is, in fact, behaving sensibly. In other cases, additional learning from additional experience will resolve problems due to statistical anomalies. Further analysis and experimentation will bring deeper insights into this learning process.

IV. Related Work

Drescher [9] created the *schema mechanism*, by which an agent progressively constructed knowledge of the world in the form of schemas that linked contexts, actions, and results, all consisting of Boolean variables. The schema mechanism suffered from an intractable proliferation of rules, partly (we believe) because it lacked the stratified model that imposes a hierarchical structure on the set of variables in our approach.

Cohen et al. [11] created an agent called Neo that, like our agent, learned by looking for events that tend to co-occur, however the input into the learning algorithm was discrete. Cohen et al. [12] have also done work with abstracting sensor input using clustering of time series data.

Das et al. [13] also cluster time series data. Their algorithm discretizes the data by clustering subsequences using a sliding window. They then learn rules from this discretized data, but the rules do not in turn influence the discretization.

To learn both a Bayesian network and the discretizations of the variables for that network, Friedman and Goldszmidt [14] iterate between the two. They learn new thresholds for variables in an unsupervised setting by seeking to maximize the mutual information between each variable and its parents. Their work addresses the problem of building Bayesian networks from continuous data. The purpose of a Bayesian network is to compactly represent a full joint probability distribution. Our method is designed, like Drescher's, for environments that cannot be completely characterized.

V. Conclusions and Future Work

We have presented preliminary results from an implemented computational model of an important kind of foundational developmental learning. Our model demonstrates an important synergy between the learning of landmarks representing important qualitative distinctions, and the learning of rules that exploit those distinctions to make reliable predictions. These qualitative distinctions make it possible to define discrete events, and then to identify predictive rules describing regularities among events and the values of context variables.

We have demonstrated these only with the very first steps in learning to control the hand of our "baby robot" and to interact with a single object. There is much more to learn, and the learning mechanism will undoubtedly evolve in response to more extensive and more quantitative evaluation.

The next major step will be to incorporate active learning. The current learning process uses sensorimotor experience resulting from random "motor babbling" behavior. Once the earliest levels of representation have been learned, new landmarks to resolve non-determinism in the rules can only be learned by actively pursuing experience in the non-deterministic region. The rules learned in this paper encode the knowledge necessary for the creation and execution of simple plans needed to pursue such experience. The next step in our research is to develop and evaluate the mechanisms for applying that knowledge.

References

[1] W. James, *The Principles of Psychology*, 1890.

[2] J. Klein, "Breve: a 3d environment for the simulation of decentralized systems and artificial life," in *ICAL 2003: Proceedings of the eighth international conference on Artificial life*. Cambridge, MA, USA: MIT Press, 2003, pp. 329–334.

[3] D. M. Pierce and B. J. Kuipers, "Map learning with uninterpreted sensors and effectors." *Artificial Intelligence*, vol. 92, pp. 169–227, 1997.

[4] B. Kuipers, P. Beeson, J. Modayil, and J. Provost, "Bootstrap learning of foundational representations," *Connection Science*, vol. 18, no. 2, pp. 145–158, 2006.

[5] L. A. Olsson, C. L. Nehaniv, and D. Polani, "From unknown sensors and actuators to actions grounded in sensorimotor perceptions," *Connection Science*, vol. 18, no. 2, pp. 121–144, 2006.

[6] J. Modayil and B. Kuipers, "Bootstrap learning for object discovery," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2004.

[7] ——, "Autonomous shape model learning for object localization and recognition," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2006.

[8] B. Kuipers, *Qualitative Reasoning*. Cambridge, Massachusetts: The MIT Press, 1994.

[9] G. L. Drescher, *Made-Up Minds: A Constructivist Approach to Artificial Intelligence*. Cambridge, MA: MIT Press, 1991.

[10] U. M. Fayyad and K. B. Irani, "Multi-interval discretization of continuousvalued attributes for classification learning," in *Thirteenth International Joint Conference on Articial Intelligence*, vol. 2. Morgan Kaufmann Publishers, 1993, pp. 1022–1027.

[11] P. R. Cohen, M. S. Atkin, T. Oates, and C. R. Beal, "Neo: Learning conceptual knowledge by sensorimotor interaction with an environment," in *Agents '97*. Marina del Rey, CA: ACM, 1997.

[12] P. R. Cohen, T. Oates, C. R. Beal, and N. Adams, "Contentful mental states for robot baby," in *Proc. 18th National Conf. on Artificial Intelligence (AAAI-2002)*. AAAI/MIT Press, 2002.

[13] G. Das, K.-I. Lin, H. Mannila, G. Renganathan, and P. Smyth, "Rule discovery from time series," in *Knowledge Discovery and Data Mining*, 1998, pp. 16–22.

[14] N. Friedman and M. Goldszmidt, "Discretizing continuous attributes while learning bayesian networks," in *International Conference on Machine Learning*, 1996, pp. 157–165.