Representing systems with hidden state

Christopher Hundt
McGill University
Montreal, Canada
chundt@cs.mcgill.ca

Prakash Panagaden
McGill University
Montreal, Canada
prakash@cs.mcgill.ca

Joelle Pineau
McGill University
Montreal, Canada
jpineau@cs.mcgill.ca

Doina Precup
McGill University
Montreal, Canada
dprecup@cs.mcgill.ca

Abstract

We discuss the problem of finding a good state representation in stochastic systems with observations. We develop a duality theory that generalizes existing work in predictive state representations as well as automata theory. We discuss how this theoretical framework can be used to build learning algorithms, approximate planning algorithms as well as to deal with continuous observations.

Introduction

Learning and planning under uncertainty is a crucial focus of modern AI research. In this paper, we address the issue of sequential decision making. Much of the work in this field is based on the framework of Partially Observable Markov Decision Processes (POMDPs) (Kaelbling et al., 1998). In this framework, problems are modeled using discrete states and actions. Actions cause stochastic transitions between states. At each time step, a stochastic observation is also generated, based on the current state and the previous action. Much work has been devoted to planning in POMDPs when a model of the system (in terms of the stochastic transitions between states and the probability distributions over observations) is known. Unfortunately, learning POMDPs from data is a very difficult problem. One standard algorithmic solution is expectation maximization (EM) (Chrisman, 1992), but for POMDPs this approach is plagued by local minima (more so than for other probabilistic models) and works poorly in practice unless a good initial model of the system is used (Shatkay & Kaelbling, 1997). History-based methods (McCallum, 1995) often work better in practice but are less general. A lot of recent research has been devoted to finding alternative representations for such systems, e.g., diversity-based representation (Rivest & Schapire, 1994), predictive state representations (PSRs) (Littman et al., 2002) and TD-networks (Sutton & Tanner, 2005). These approaches aim to combine the generality of POMDPs with the ease of learning of history-based methods. The key idea underlying all of these approaches is that the state of the system is not considered as predefined; instead, it is viewed as a sufficient statistic for making (probabilistic) predictions about future trajectories. However, the models themselves are different and their relationships are only partially understood at the moment.

In this paper, we develop a duality theory for POMDPs, which unifies much of the existing work on predictive representations. We show how, for any POMDP, one can develop two alternative representations: a dual machine and a double-dual machine. The key idea in the development is that of making measurements on the system, which we call experiments. Experiments are sequences of actions interspersed with observations. They generalize previous notions of tests from the literature on predictive state representations. Both of the alternative representations that we present allow an accurate prediction of the probability of any experiment. The double-dual representation is of particular interest, because it has a *deterministic* transition structure, and no hidden state. Instead, its states can be thought of as "bundles" of predictions for experiments. As such, this representation holds the promise of much better planning and learning algorithms than those currently available. Our work also generalizes similar representations from automata theory (Brzozowski, 1962; Rivest & Schapire, 1994). We show how existing predictive representations can be viewed from the perspective of this framework. We also discuss the implications of these alternative representations for learning algorithms, approximate planning algorithms as well as working with continuous observations.

Background and definitions

A POMDP is a tuple $\mathcal{M} = \langle S, A, O, \delta_a : S \times S \longrightarrow [0,1], \gamma_a : S \times O \longrightarrow [0,1] \rangle$ where S is a finite set of states; A is a finite set of actions; O is a finite set of observations; $\delta_a(s,s') = Pr(s_{t+1} = s' | s_t = s, a_t = a), \forall a \in A$ is a stochastic transition function; and $\gamma_a(s,o) = Pr(o_{t+1} = o | a_t = a, s_{t+1} = s), \forall a \in A$ is an emission function, describing the probabilities of different observations. We are omitting here an explicit representation of rewards, which are used to determine the optimal strategy for choosing actions. Instead, one can think of rewards as part of the observation vector.

Our goal is to develop an alternative representation, in which the notion of state is redefined. We want to obtain a representation that is indistinguishable from the original system, in terms of the probability that it assigns to any trajectory. In order to formalize this goal, we define the notions of tests and experiments.

Definition 1 *A **test** t is a non-empty sequence of actions followed by an observation, i.e. $t = a_1 \cdots a_n o$, with $n \geq 1$.*

Definition 2 *An **experiment** is a non-empty sequence of tests $e = t_1 \cdots t_m$ with $m \geq 1$.*

Note that these definitions force one to take an action in order to observe anything; this is a consequence of the way observations are defined. If observations only depend on states, this restriction can be lifted. This notion of tests has been called *e-tests* in previous work by Ruddary & Singh (2004). The notion of *s-tests*, present in some of the literature on predictive state representations (Littman et al., 2002; Singh et al., 2004), corresponds to a special case of experiments in which all component tests contain just one action. However, we do not limit ourselves here to s-tests, as the algebraic structure they define is difficult to work with. Also, as we will see later, it is computationally advantageous to consider tests based on extended sequences of actions.

In order to proceed with the construction of the dual to a POMDP, we need to define a generalization of the transition function that works on sequences of actions.

Definition 3 *Given a POMDP, we define a transition function δ_α, where α is a sequence of actions, inductively: δ_a is as in the POMDP model, and:*

$$\delta_{a\alpha}(s,s') = \sum_{s'' \in S} \delta_a(s,s'')\delta_\alpha(s'',s'), \forall s,s' \in S. \quad (1)$$

Let $t = \alpha o$ be a test. We denote by $\langle s|t|s'\rangle$ the probability of the system arriving in state s' and emitting o if the action sequence α is executed starting from s. Given $\alpha = a_1 \cdots a_n \in$ and $o \in O$ we have

$$\langle s|\alpha o|s'\rangle = \delta_\alpha(s,s')\gamma_{a_n}(s',o). \quad (2)$$

Note the need to look at the last action in the sequence; this is due to the standard way in which POMDPs are defined, and explains why we must insist that experiments have at least one action. A similar notion for longer experiments can be constructed by induction as follows:

$$\langle s|te|s'\rangle = \sum_{s'' \in S} \langle s|t|s''\rangle\langle s''|e|s'\rangle. \quad (3)$$

One can define the notion of measurement for an experiment e from state s by just summing over the final states. We use the same angle bracket notation for this.

Definition 4 *The **prediction** of experiment e for state s, $\langle s|e\rangle$, is defined as:*

$$\langle s|e\rangle = \sum_{s' \in S} \langle s|e|s'\rangle. \quad (4)$$

Note that this is the same as the definition of predictions in the literature on predictive state representations: the prediction gives the probability of recording the specified sequence of observations, at the specified times, given that the specified sequence of actions is executed.

Our goal is to examine representations for POMDPs that capture behavior, rather than being specified a priori. One way to "probe" the behavior of the system described by $\mathcal{M}$ is to *run experiments*, starting at different hidden states, and record the results. Running an experiment $e = \alpha_1 o_1 \ldots \alpha_m o_m$ means that the sequence of actions $\alpha_1 \ldots \alpha_n$ corresponding to e is executed. The experiment "succeeds" if the corresponding sequence of observations $o_1 \ldots o_m$ is observed; otherwise, the experiment "fails". The success of an experiment can thus be considered a binomial random variable; the prediction for the experiment from a given state s defines the probability distribution of this variable at s. Hence, the prediction for the experiment can be estimated from data by counts. Of course, if two states cannot be distinguished by any experiments, they are redundant and one of them can be eliminated from the description of the system. Similarly, if two experiments always give the same results, they are redundant. We formalize these intuitions in the following definitions:

Definition 5 *Two experiments e_1 and e_2 are equivalent for $\mathcal{M}$, denoted $e_1 \sim_\mathcal{M} e_2$, if and only if $\langle s|e_1\rangle = \langle s|e_2\rangle$ for all $s \in S$. We denote by $[e]_\mathcal{M}$ the $\sim_\mathcal{M}$-equivalence class of e and by $\langle s|[e]_\mathcal{M}\rangle$ the prediction for any experiment in this class, when it is executed from state s.*

In other words, experiments are equivalent if their predictions are identical from any states in $\mathcal{M}$. We define an analogous equivalence relation over states.

Definition 6 *Two states $s_1, s_2 \in S$ are equivalent for $\mathcal{M}$, denoted $s_1 \sim_\mathcal{M} s_2$, if and only if they cannot be distinguished by any experiment, i.e., $\langle s_1|e\rangle = \langle s_2|e\rangle$ for all experiments e. We denote by $[s]_\mathcal{M}$ the $\sim_\mathcal{M}$-equivalence class of s and by $\langle [s]_\mathcal{M}|e\rangle$ the prediction for the success of experiment e when started from any state in the class $[s]_\mathcal{M}$.*

These are easily shown to be equivalence relations, and they are at the hinge of the duality theory.

Duality for POMDPs

In mathematics, duality is usually associated with a general transformation applied to a mathematical object or structure. Applying the transformation once usually yields a different-looking structure. Applying the transformation twice yields the original object (or something indistinguishable from it). Perhaps the most popular notion of duality is the one used in mathematical programming, between the primal and dual representation of an optimization problem. However, many other dualities exist, some giving rise to very useful alternative representations. For example, Pontryagin duality gives rise to the notion of Fourier transform.

In this section, we will use the equivalence relations defined above in order to define the generic transformation needed for duality. We will first define the dual representation, using the notion of equivalence of experiments. This should be reminiscent of the work of Rivest & Schapire (1994) on diversity-based representations for automata. We will then define the double dual by using the notion of equivalence of states. Intuitively, it should be clear that the representation to be created will be identical to the original system, since only equivalences are used at every step. The generic transformation that will provide the dual representation relies on switching the roles of states and experiments.

In order to understand what the "dual" view of a POMDP might mean, consider for a moment the case of a *deterministic* POMDP, in which all transitions and all emissions have probability 0 or 1. One way to describe the system is to specify, for each state s, the collection of experiments that succeed when started in s. This is a "forward" view, useful to make predictions about the future, as well as for forward planning. A different approach is to specify for each experiment a *precondition*, i.e., the set of states from which the experiment will succeed. This is a "backward" view, used in classical AI by backward planning algorithms. We will call this the *dual representation*. Of course, the notion of precondition cannot be used directly in probabilistic systems. However, a dual representation based on this intuition can still be defined, as follows:

Definition 7 *We define the dual $\mathcal{M}'$ of a POMDP $\mathcal{M}$ as a tuple: $\mathcal{M}' = (S', A, O', \delta'_a : S' \longrightarrow S', \gamma' : S' \times O' \longrightarrow [0,1]$, where:*

- *the new set of states is the equivalence classes of tests from $\mathcal{M}$: $S' = \{[e]_{\mathcal{M}}\}$*
- *the new observations are the states of $\mathcal{M}$: $O' = S$*
- *the transition functions are defined as: $\delta'_a([e]_{\mathcal{M}}) = [ae]_{\mathcal{M}}, \forall a \in A$*
- *the emission function is $\gamma'([e]_{\mathcal{M}}, s) = \langle s|[e]_{\mathcal{M}}\rangle$*

Note that this is a *deterministic* transition system (see the definition of δ'_a), which is somewhat surprising given that we started with a stochastic system. This is due to the fact that the information is organized around experiments from the previous machine, and experiments are constructed from other experiments using concatenation. This is essentially the structure reflected in the transition functions δ'_a Note also that the emission function γ' is *not* a probability distribution anymore; rather, it specifies to what extent each state is a "precondition" for the experiments in each equivalence class, using the notion of conditional probability. Also, γ' does not depend on actions (like the emission function in the original system). This is due to the fact that actions are part of the experiments.

The dual machine is *not* a generative model, and its purpose is not to be executed. This is apparent from the fact that the emission function is not normalized. To understand this, note that if we consider a deterministic POMDP (like we did initially), the dual provides, for each experiment, exactly the set of states from which the experiment succeeds. This is a set, and cannot be converted into probabilities of the form $p(s|[e]_{\mathcal{M}})$ unless we have an initial state distribution. However, we want to stay away from initial distributions, because we are seeking a representation of the system that is independent of where it starts. Finally, note that our dual is a strict generalization of the update graph used by Rivest & Schapire (1994) for deterministic finite automata with observations.

The dual representation could potentially be quite useful for backward planning approaches. In MDPs, a typical example of such an approach is prioritized sweeping (Moore & Atkeson, 1993), which drastically reduces the time for computing a value function, by propagating information to predecessor states. A similar role is played by eligibility traces in reinforcement learning (Sutton & Barto, 1998). However, in the POMDP planning literature virtually all methods rely on forward planning. We will investigate the use of the dual representation for backward planning in the future.

We now show that the transition structure in the dual is well-defined. Essentially, the transitions in the dual are of the following form: an experiment e goes under an action a to an experiment ae. For this to be well-defined, we need to show that it does not matter which representative of the equivalence class of e is chosen. To do this, we first show a small helper lemma:

Lemma 8 *For any states $s_1, s_2 \in S$, action $a \in A$ and experiment e, $\langle s_1|ae|s_2\rangle = \sum_{s\in S} \delta_a(s_1,s)\langle s|e|s_2\rangle$ and $\langle s_1|ae\rangle = \sum_{s\in S}\delta_a(s_1,s)\langle s|e\rangle$*

Proof: We show this by induction on the length of e. For the base case, suppose that e consists of just one test $t = a_1 \ldots a_n o$. Then we have:

$$\begin{aligned}
\langle s_1|at|s_2\rangle &= \langle s_1|aa_1\ldots a_n o|s_2\rangle \\
&= \delta_{aa_1\ldots a_n}(s_1,s_2)\gamma_{a_n}(s_2,o) \text{ (from (2))} \\
&= \left(\sum_{s\in S}\delta_a(s_1,s)\delta_{a_1\ldots a_n}(s,s_2)\right)\gamma_{a_n}(s_2,o) \text{ (from (1))} \\
&= \sum_{s\in S}\delta_a(s_1,s)\langle s|t|s_2\rangle \text{ (by rearranging and (2))}
\end{aligned}$$

Now suppose the experiment is of the form te. Then we have:

$$\begin{aligned}
\langle s_1|ate|s_2\rangle &= \sum_{s'\in S}\langle s_1|at|s'\rangle\langle s'|e|s_2\rangle \text{ (from (3))} \\
&= \sum_{s'\in S}\left(\sum_{s\in S}\delta_a(s_1,s)\langle s|t|s'\rangle\right)\langle s'|e|s_2\rangle \text{ (base case)} \\
&= \sum_{s\in S}\delta_a(s_1,s)\sum_{s'\in S}\langle s|t|s'\rangle\langle s'|e|s_2\rangle \text{ (by rearranging)} \\
&= \sum_{s\in S}\delta_a(s_1,s)\langle s|te|s_2\rangle \text{ (from (3))}
\end{aligned}$$

Now for the second formula, we have:

$$\begin{aligned}
\langle s_1|ae\rangle &= \sum_{s_2\in S}\langle s_1|ae|s_2\rangle \\
&= \sum_{s\in S}\delta_a(s_1,s)\sum_{s_2\in S}\langle s|e|s_2\rangle \text{ (by rearranging)} \\
&= \sum_{s\in S}\delta_a(s_1,s)\langle s|e\rangle \text{ (from (4))}
\end{aligned}$$ ◇

The next lemma shows that the transitions in the dual machine are well defined.

Lemma 9 *If $e_1 \sim_{\mathcal{M}} e_2$ then $ae_1 \sim_{\mathcal{M}} ae_2, \forall a \in A$.*

Proof: Since $e_1 \sim_{\mathcal{M}} e_2$, we have $\langle s|e_1\rangle = \langle s|e_2\rangle$ for all $s \in S$. Then for any state s, we have:

$$\begin{aligned}
\langle s|ae_1\rangle &= \sum_{s'\in S}\delta_a(s,s')\langle s'|e_1\rangle \text{ (from Lemma 8)} \\
&= \sum_{s'\in S}\delta_a(s,s')\langle s'|e_2\rangle \text{ (because } e_1 \sim_{\mathcal{M}} e_2) \\
&= \langle s|ae_2\rangle \text{ (from Lemma 8)}
\end{aligned}$$ ◇

Now we proceed to define the double-dual representation. By analogy with the way we constructed the dual, we will consider tests on the dual machine $\mathcal{M}'$ and their equivalence classes. In order to see which tests are of interest in the dual, consider the semantics of the emission function γ'. It gives the measurement for an experiment given that the *starting* state was s. This means that considering sequential experiments on $\mathcal{M}'$ does not really make sense. Instead, we will only consider tests on the dual, of the form αs, where α is a sequence of actions.

Definition 10 *The prediction for a test on the dual given an experiment is defined recursively as:*

$$\begin{aligned} \langle [e]_{\mathcal{M}} | s \rangle &= \gamma'([e]_{\mathcal{M}}, s) = \langle s | [e]_{\mathcal{M}} \rangle \\ \langle [e]_{\mathcal{M}} | a\alpha s \rangle &= \langle [ae]_{\mathcal{M}} | \alpha s \rangle \end{aligned} \qquad \diamond$$

To understand what these predictions are, we will show a simple lemma:

Lemma 11 *The prediction for a test on the dual is* $\langle [e]_{\mathcal{M}} | \alpha s \rangle = \langle s | [\alpha^R e]_{\mathcal{M}} \rangle$*, where* α^R *denotes the reverse of* α*.*

Proof: The base case is trivial based on the definition. For the induction step, using Definition 10 and the induction hypothesis, we have:

$$\langle [e]_{\mathcal{M}} | a\alpha s \rangle = \langle [ae]_{\mathcal{M}} | \alpha s \rangle = \langle s | [\alpha^R a e]_{\mathcal{M}} \rangle = \langle s | [(a\alpha)^R e]_{\mathcal{M}} \rangle \qquad \diamond$$

We now define an equivalence relation for these tests:

Definition 12 *Two tests* $t_1 = \alpha_1 s_1$ *and* $t_2 = \alpha_2 s_2$ *are* $\mathcal{M}'$*-equivalent if and only if* $\langle [e]_{\mathcal{M}} | t_1 \rangle = \langle [e]_{\mathcal{M}} | t_2 \rangle$*, for all experiments e. We denote by* $[t]_{\mathcal{M}'}$ *the equivalence class of* t*.*

With this notion, we are now ready to define the double-dual. Again, we will flip the role of the states and the tests on the dual.

Definition 13 *We define the double-dual as:* $\mathcal{M}'' = (S'', A, O'', \delta'', \gamma'')$*, where:*

$$\begin{aligned} S'' &= \{[t]_{\mathcal{M}'}\} & \delta''([t]_{\mathcal{M}'}, a) &= [at]_{\mathcal{M}'} \\ O'' &= S' = \{[e]_{\mathcal{M}}\} & \gamma''([t]_{\mathcal{M}'}, [e]_{\mathcal{M}}) &= \langle [e]_{\mathcal{M}} | t \rangle \end{aligned}$$

Like in the case of the dual, in order to show that the double-dual is well defined, we need the following lemma:

Lemma 14 *If* $t_1 \sim_{\mathcal{M}'} t_2$ *then* $at_1 \sim_{\mathcal{M}'} at_2$ *for any action* $a \in A$*.*

Proof: For any experiment e, we have:

$$\langle [e]_{\mathcal{M}} | at_1 \rangle = \langle [ae]_{\mathcal{M}} | t_1 \rangle = \langle [ae]_{\mathcal{M}} | t_2 \rangle = \langle [e]_{\mathcal{M}} | at_2 \rangle$$

where we used definition 10 in the first and third equalities and the fact that $t_1 \sim_{\mathcal{M}'} t_2$ in the second equality. $\diamond$

Now that we know that these constructions are well defined the fundamental fact of the duality is captured by the following theorem.

Theorem 15 *The prediction for an experiment e from a state s,* $\langle s | e \rangle$*, is given by* $\langle [s]_{\mathcal{M}'} | [e]_{\mathcal{M}} \rangle = \gamma''([s]_{\mathcal{M}'} | [e]_{\mathcal{M}} \rangle$*, where* $[s]_{\mathcal{M}'}$ *indicates the equivalence class of the test on the dual which has s as an observation and an empty sequence of actions.*

The proof is immediate from the construction.

Note that the double-dual has states corresponding not only to $[s]_{\mathcal{M}'}$, but also to extended tests $[\alpha s]_{\mathcal{M}'}$. The theorem above says that the equivalence classes of states are sufficient in order to describe the behavior of the system; this is true here because the emissions are defined in each double-dual state for *all* experiments. Hence, each state $[s]_{\mathcal{M}'}$ in the double-dual can be viewed as a "bundle" of predictions for all possible experiments, when starting in s. But these cannot really be directly measured, unless we have a way to reset the POMDP to a specific initial state (a very unlikely assumption). Instead, equivalence classes for tests of the form $[s\alpha]_{\mathcal{M}'}$ are of more interest, especially for action sequences α that are defined as *homing sequences*. A homing sequence is a sequence of actions that puts a dynamical system in a known state. Such sequences were defined by Rivest & Schapire (1993) in the context of learning deterministic automata. Evan-Dar, Kakade & Mansour (2005) recently showed that in a POMDP, there exist homing strategies, which put the system in a known belief state. The equivalence classes for tests containing such strategies are the best candidates for estimation from data in the dual machine. A similar notion is that of *controllability* from discrete-time dynamical systems (Santina et al., 1996). A system is called controllable if it can be set to a known state with a finite sequence of actions. We note that duality notions exist in the control literature, and the relationship with the duality that we put forth here will need to be explored further.

We want to emphasize that the double-dual is a *conceptual* construction, not an algorithmic solution. A different way of viewing it is as an encoding of the *state-test prediction matrix* (Littman et al., 2002). This matrix contains predictions (or measurements) for s-tests, which are a special class of experiments. The double-dual essentially holds the distinct rows of this matrix, each associated with the equivalence class of a state. The duplicate columns have been eliminated. Of course, there are more compact ways of encoding this matrix. The PSR approach is based on the fact that the rank of the matrix is upper bounded by the number of states of the original system. Hence, as shown in (Littman et al., 2002), the matrix can actually be described by a number of linearly independent columns (called core tests) and by a finite number of parameter vectors (one for each action-observation pair, and one for each extension of a core test by an action-observation pair). The same observations related to the rank hold here as well. We also observe that the predictions for s-tests are always sufficient to reconstruct predictions for experiments. However, considering experiments is advantageous both theoretically (no duality construction is possible without them) and empirically, as we discuss further below.

An example

We present a simple example to illustrate our duality construction on a POMDP system. The domain is depicted in Figure 1 and the corresponding POMDP is in Figure 2.

Table 1 shows the state-experiment predictions for several tests. Clearly, some of the experiments belong to the same

Table 1: State-experiment prediction matrix

	N*R*	N*B*	S*R*	S*B*	E*R*	E*B*	W*R*	W*B*	NN*R*	...	N*RNR*	...
s1	0	1	0	1	0.5	0.5	0	1	0		0	
s2	0.5	0.5	0.5	0.5	0.5	0.5	0	1	0.5		0.25	
s3	0	1	0	1	0.5	0.5	0	1	0		0	
s4	0.5	0.5	0.5	0.5	0.5	0.5	0	1	0.5		0.25	

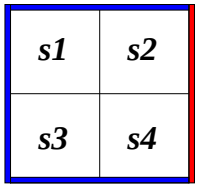

Figure 1: A four-state navigation domain. The top, bottom and left walls are painted Blue(=*B*), the right wall is painted Red(=*R*). The robot can take actions: North(=N), South(=S), East(=E), West(=W), which have the expected (deterministic) effects. In any state, the robot observes the color of one of the two adjacent walls (randomly chosen with equal probability).

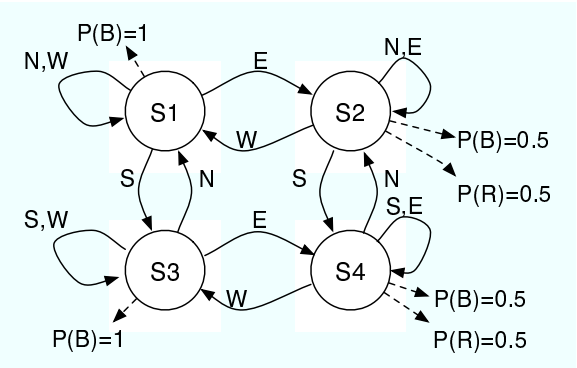

Figure 2: POMDP (original) representation of the domain. Solid arcs denote transitions and dashed arrows denote emissions.

equivalence class, for example, N$R \sim_{\mathcal{M}}$ SR. In fact, we can identify five equivalence classes of tests with one observation: $e1 = [\mathrm{N}R]_{\mathcal{M}}$, $e2 = [\mathrm{N}B]_{\mathcal{M}}$, $e3 = [\mathrm{E}R]_{\mathcal{M}}$, $e4 = [\mathrm{W}B]_{\mathcal{M}}$, $t5 = [\mathrm{W}R]_{\mathcal{M}}$. The fragment of the dual automaton containing these classes is presented in Figure 3. Similarly, we can identify equivalence classes with two observations (e.g., *NRNR*, which is equivalent to *SRSR*, *NRSR*, etc), three observations etc. Each of these categories will form a separate fragment, or connected component, of the dual automaton, and transitions will be only within that component. In general, if the observations are stochastic, an infinite number of such components will exist, and obviously we do not want an explicit representation for all of them. Analyzing the structure in terms of the components points us to the following observation. Within each component, the predictions for experiments can be defined in terms of predictions of *other* experiments. For instance, the prediction of whether *NER* will succeed is the same are the prediction that *ER* will succeed after an *N* action. These are *temporal-difference* relationships, of the sort explicitly captured in TD-networks (**?**). However, no such relationships exist for tests in different components.

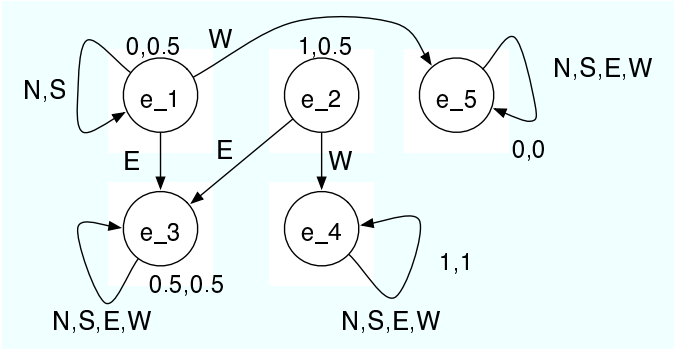

Figure 3: The automaton corresponding to the dual. Note that the emission functions are not normalized.

Observe that s-tests end up, in our example here, in different components. This reflects the fact that s-test are not *compositional*: the prediction of observing $a_1o_1a_2o_2$ cannot be obtained, in general, from the prediction of a_2o_2. This is due to the fact that making observation o_1 carries information, which influences the probability of subsequent experiments. However, experiments that allow for sequences of actions are compositional; this is clearly shown by the fact that components consist of multiple related tests and experiments.

Similarly we can construct the double-dual, and the most relevant part of it is depicted in Figure 4. This contains the equivalence classes for states s_1 and s_2 from the original machine. In this example, because the POMDP has deterministic transitions, there are actually two very simple homing sequences: action E will ensure that the system is in s_2, while action W will ensure that the system is in s_1. However, such exact reseting will not be possible in POMDPs with stochastic transitions.

Relationship to automata theory

An important special case of this theory is that of deterministic systems, i.e. automata with deterministic observations. In automata terminology, actions are alphabet symbols, labels or inputs. Most of the work is carried out in systems that

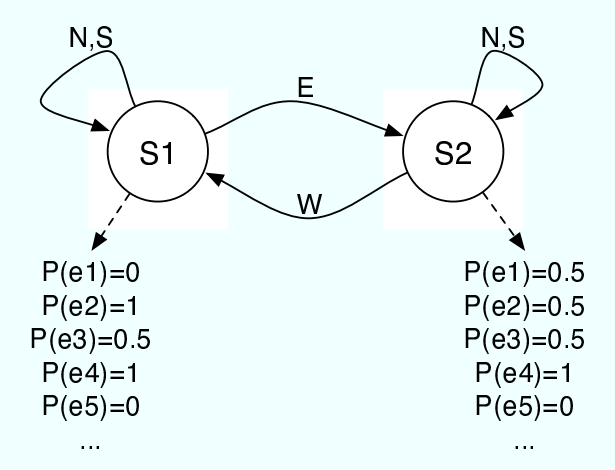

Figure 4: The automaton corresponding to the double dual.

can just accept or reject, i.e., which have only two observations. The case of deterministic automata with multiple observations has been studied by Rivest and Schapire (1994), who define an *update graph* associated with an automaton. The structure of this update graph is exactly the same as the transition structure of the dual machine we defined. However, their work does not describe this as an automaton with observations.

For deterministic automata, we can show that the double-dual machine is actually the *minimal automaton* that is *equivalent* (isomorphic) to the original machine (the proof is omitted here for lack of space but is presented in a forthcoming paper). As a matter of fact, a similar construction for minimizing traditional finite state automata was used by Brzozowski in 1962. His intriguing algorithm is as follows: take the transitions of the automaton, reverse them and flip the accepting and non-accepting states. Of course, the resulting machine is not deterministic, so it is determinized in the usual way. Then reverse the result, flip the accepting and non-accepting states and determinize again. It turns out that the result of each of these operations are the dual, and double-dual respectively (for the special case when we only have two observations).

A similar duality construction can also be done for the case of nondeterministic automata. Remarkable, the double-dual of a nondeterministic automaton is the minimal deterministic automaton that produces the same sequences of observations. The details go beyond this paper and are presented in a forthcoming extended version.

Discussion and extensions

The representations we introduce relate in interesting ways to both PSRs and TD-networks. As seen above, PSRs are a finite encoding of the double-dual machine, which exploit linear independence assumptions. However, to date, all of the work has been done on *exact* (lossless) representations of the original system; the number of parameters needed (in addition to the core tests) depends on the number of actions and observations in the system. This becomes a problem if the number of observations is large.

Now let us consider the case in which observations are continuous. This case is very important for practical applications, yet has received very little attention, with a few exceptions (Hoey & Poupart, 2005). The definition of a POMDP can be generalized to this case, e.g., using emission functions that are probability density functions. The state-experiment matrix will now become an operator. However, operators behave much like matrices, and have a notion of rank. It is not hard to see that the rank of the operator will be limited by the number of states in the original POMDP. To do this, consider increasingly fine discretizations of the observation space. For each such discretization, we have a corresponding finite POMDP. Each such POMDP has a state-experiment matrix whose rank is limited by the number of states of the original system. In the limit, as the size of a discretization cell goes to 0, the rank will still be the same, an limited by the number of states. However, for each of these systems, the number of parameters needed for a PSR depends linearly on the number of observations. Hence, exact PSR representations will not be feasible in general. This motivates the need for *approximate methods*. Of course, in POMDPs planning is always performed to a finite horizon, so this immediately restricts the length of the experiments of interest. However, more aggressive methods will be necessary in general.

One suggestion that comes from the double-dual representation is to replace equivalence by an ϵ-equivalence, where ϵ is a parameter that controls the precision of the representation. This would allow for a much coarser double-dual, on which PSR representations can be developed. All learning and planning algorithms presented so far would transfer immediately.

Another class of approximations restricts attention to predictions of certain observations. This is the case in TD-nets, where predictions of interest are specified in the question network, as well as in the work of Poupart & Boutilier (2003) on lossy compression methods for POMDPs, where the goal is to predict rewards. Point-based planning methods for POMDPs can also be viewed as working with an approximation of the double-dual, in which only certain parts of the automaton are represented (which ones depends on the heuristics of the different algorihtms).

We plan to explore these issues further in future work. We will also investigate the connections of predictive representations in AI to notions of duality established for automata (Arbib & Manes, 1974) and control theory (Santina et al., 1996).

References

Arbib, M. A., & Manes, E. G. (1974). Machines in a category: An expository introduction. *SIAM Review*, *16*, 163–192.

Brzozowski, J. A. (1962). Canonical regular expressions and minimal state graphs for definite events. *Symposium on Mathematical Theory of Automata* (pp. 529–561).

Chrisman, L. (1992). Reinforcement learning with perceptual aliasing: The perceptual distinctions approach. *Proceedings of the Tenth National Conference on Artificial Intelligence* (pp. 183–188).

Even-Dar, E., Kakade, S. M., & Mansour, Y. (2005). Reinforcement learning in POMDPs without resets. *Proceedings of IJCAI* (pp. 690–695).

Hoey, J., & Poupart, P. (2005). Solving POMDPs with continuous or large discrete observation spaces. *Proceedings of IJCAI* (pp. 1332–1338).

Kaelbling, L. P., Littman, M. L., & Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, *101*.

Littman, M., Sutton, R. S., & Singh, S. (2002). Predictive representations of state. *Advances in Neural Information Processing Systems* (pp. 1551–1561).

McCallum, A. (1995). *Reinforcement learning with selective perception and hidden state*. Doctoral dissertation, University of Rochester.

Moore, A. W., & Atkeson, C. G. (1993). Prioritized sweeping: Reinforcement learning with less data and less real time. *Machine Learning*, *13*, 103–130.

Poupart, P., & Boutilier, C. (2003). Value-directed compression of POMDPs. *Advances in Neural Information Processing Systems* (pp. 1547–1554).

Rivest, R. L., & Schapire, R. E. (1993). Inference of finite automata using homing sequences. *Information and Computation*, *103*, 299–347.

Rivest, R. L., & Schapire, R. E. (1994). Diversity-based inference of finite automata. *Journal of the ACM*, *41*, 555–589.

Rudary, M., & Singh, S. (2004). A nonlinear predictive state representation. *Advances in Neural Information Processing Systems* (pp. 855–862).

Santina, M. S., Stubberud, A. R., & Hostetter, G. H. (1996). Discrete-time systems. *The control handbook* (pp. 239–252). CRC Press.

Shatkay, H., & Kaelbling, L. P. (1997). Learning topological maps with weak local odometric information. *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence* (pp. 920–929).

Singh, S., James, M. R., & Rudary, M. R. (2004). Predictive state representations: A new theory for modeling dynamical systems. *Uncertainty in Artificial Intelligence* (pp. 512–519).

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

Sutton, R. S., & Tanner, B. (2005). Temporal-difference networks. *Advances in Neural Information Processing Systems*.