Learning approximate predictive models

COMP652 - Final project
Monica Dinculescu

January 7, 2009

Abstract. Learning the internal representation of partially observable environments has proven to be a difficult problem. State representations which rely on prior models, such as partially observable Markov decision processes (POMDPs) are computationally expensive and sensitive to the accuracy of the underlying model dynamics. Recent work by Hundt et al. proposes a duality construction that yields a minimal deterministic model which can make the same predictions about future observations as the original POMDP. We extend this theory to a class of history-based models, and consider situations in which an agent is only interested in a subset of the available observations. We show how to construct a history machine from any POMDP, and show that the double dual is well formed. Finally we propose an efficient algorithm that learns this structure from data, and is able to make predictions with the same accuracy as the original POMDP.

Introduction

Probabilistic models are necessary for decision making in complex realistic environments. Agents often cannot predict the exact outcome of their actions due to noisy sensors or incomplete knowledge of the world. Learning the internal representation of such partially observable environments has proven to be a difficult problem.

Consider an agent that tries to build a model of the world in order to answer questions about it. If the environment is too complex, the space of model parameters may be too large and the agent may be computationally incapable of updating them. One such example is a robot navigating in a room, looking for the door. Rather than representing the entire room, the robot can choose to only ask questions about its relative distance to a door. The agent can now construct an internal representation that can be used to make accurate predictions about certain features of interest in the world, and that is simpler than the full model of the world.

We are interested in constructing this internal representation such that it has maximal predictive power. That is, the internal representation of the system should be able to make good predictions about future observations, based on available data. There have been two predominant approaches in predicting a sequence of observations: partially observable Markov decision processes (POMDPs), and the recently proposed predictive state representations (PSRs)[LSS02].

Learning POMDPs from data is known to be very difficult. The most popular algorithm is an extension of the Baum-Welch algorithm, originally developed for hidden Markov models(HMMs)[C92]. The main disadvantage of this approach is that estimating the state of the system assumes perfect knowledge of the underlying model and is thus sensitive to the accuracy of the initial assumptions. In addition, belief state maintenance has, in the worst case, complexity equal to the size of the state space, and exponential in the number of variables. Predictive state representations (PSRs) are a recent approach that try to represent the state of a system as a set of predictions of observable outcomes of experiments that the agent can perform on the system. A reason for interest in PSRs is that the state representation is constructed only from actions and observations seen, thus resulting in a less restrictive model. However, empirical results show that learning the PSR model requires significant amounts of data, making learning algorithms both data and computationally expensive[SLJPS].

Recent work in duality theory[HPPP06], shows a duality construction for POMDPs which gives a minimal, deterministic representation of the original system. In this framework, states are thought of as predictions for sequences of action-observation pairs. The double-dual representation is of particular interest, because it has a deterministic transition structure, and no hidden state. In this paper we extend this theory to a class of history-based models. We show how any POMDP can be transformed into such a model, which we call a history machine, and show that both the dual and double dual machines are still well formed. We also describe an efficient algorithm that learns the double dual from data, and demonstrate its effectiveness on two examples.

POMDPs

A partially observable Markov decision process(POMDP) is is a general framework for decision making under uncertainty. Formally, a POMDP is a tuple

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{O}, \tau : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,1], \gamma : \mathcal{S} \times \mathcal{O} \to [0,1], b_0)$$

where $\mathcal{S}$ is a set of states, $\mathcal{A}$ is a set of actions, $\mathcal{O}$ is a set of observations, τ is a transition function, γ is an observation function and b_0 is the initial distribution over states. We will often write $\tau_a(s, s')$ for $\tau(s, a, s')$. Similarly, we will write $\gamma_a(s', \omega)$ for $\gamma(s', a, \omega)$. The conditional probability $P(s', \omega | s, a)$ is given by $\tau_a(s, s')\gamma_a(s', \omega)$. We are omitting the representation of rewards, usually found in the AI literature.

We call a sequence of action-observations pairs, $t = a_1 o_1 \ldots a_t o_t$ a **test**. If we think of predictions of future tests as "questions", then one way to model the behaviour of a system is by defining a mechanism to answer questions about the world. Littman et al. have shown that a model that can answer all such questions can make any conditional prediction of the future [LSS02].

Because the system is partially observable, the agent never knows the true state of the world. Instead, a distribution, known as a belief state, is maintained over the set of states, S. Let $b_h(s)$ denote the probability of an agent being in state s after having observed the test h. The belief can be updated as follows:

$$b_{hao}(s') = \frac{\gamma(s', o) \sum_{s \in \mathcal{S}} b_h(s) \tau_a(s, s')}{\sum_{s' \in \mathcal{S}} \gamma(s', o) \sum_{s \in \mathcal{S}} b_h(s) \tau_a(s, s')}$$

We will now formally define predictions of tests as the probability of a set of observations occurring if a set of actions is executed. We use $\langle s_1 | t | s_2 \rangle$ to denote the probability that the system starts in state s_1, is subject to the test t and ends up in state s_2. This can be defined by induction on t as follows:

$$\begin{aligned} t = a\omega &: \\ \langle s_1 | a\omega | s_2 \rangle &= \tau_a(s_1, s_2)\gamma_a(s_2, \omega) \\ t = tt' &: \\ \langle s_1 | tt' | s_2 \rangle &= \sum_{s'} \langle s_1 | t | s' \rangle \langle s' | t' | s_2 \rangle \end{aligned}$$

We write $\langle s | t \rangle$ for $\sum_{s'} \langle s | t | s' \rangle$.

History machines

Because we would like to represent state as a set of predictions of observable outcomes of tests on the system, working with POMDPs directly is not advantageous. Rather, we would like to construct a history-based model, grounded in the original POMDP, where the state of the system is a history, and its observations are predictions of future tests.

Given a POMDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{O}, \tau : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,1], \gamma : \mathcal{S} \times \mathcal{O} \to [0,1], b_0)$ we construct a **history machine** as a tuple $\mathcal{H} = (H, \Sigma, T, \delta : H \to H, \varphi : H \times T \to [0,1])$, where

i. H is a set of action-observation sequences in $\mathcal{M}$, henceforth referred to as **histories**

ii. T is a set of action-observation sequences in $\mathcal{M}$, henceforth referred to as **tests**. The distinction is that "histories" are past behaviours of the machine, and "tests" are the subject of predictions of future behaviour, or "questions" about the world.

iii. $\Sigma = A \times O$ are the possible (action:observation) transitions between histories

iv. $\delta_{ao}(h) = hao$ is the deterministic transition function

v. $\varphi(h,t) = P(t|b_h) = \sum_{s \in S} b_h(s) \langle s|t \rangle$ is the stochastic observation emission function

Note that history machines are deterministic transition systems, due to the fact that states are tests on the original POMDP, which can be constructed from other tests by concatenation. This is also the reason why the transitions in this machine are labelled by action-observation pairs.

We now consider a specific way to limit the tests of interest. We will use a mechanism called a "probe" to filter the data from future tests. Intuitively, filtering the space of *future* tests collapses the space of observable behaviours to that in which the agent is interested. Further, filtering *histories* collapses the past by extracting predictive information from the time series.

A **probe** f on sequences of observations is a mapping $f : O^* \to \mathbb{R}$, with the following restriction: given observation sequences ω_1, ω_2, and an observation $o \in O$, then $f(\omega_1) = f(\omega_2) \Rightarrow f(\omega_1 o) = f(\omega_2 o)$. This restriction is necessary to ensure that histories in the same equivalence class with respect to the probe f will transition to the same equivalence class. For example, if $h_1, h_2 \in [h]$, then on an ao transition, the histories $h_1 o$ and $h_2 o$ should both belong in $[hao]$.

Similarly, we can define a probe g on *histories* as a mapping $g : \Sigma \times \mathcal{O}^* \to \mathbb{R}$. In this case we don't need the restriction above. Although many functions could be considered as histories probes, only those that consider the amount of information that a history has about the state of the world are actually useful.

For example, assume that the probe f (on future tests) asks the question "have I observed the goal in this sequence of actions and observations?", and gives a binary result, depending on whether the goal was indeed observed. Clearly, the g probe (on histories) should capture information about the history that is consistent in answering the same set of questions as the probe f.

A good heuristic for history probes is based on the eligibility trace notion found in Reinforcement Learning. Eligibility traces are a means of temporarily assigning credit for the occurrence of an event (in our case, the occurrence or not of a goal). When such an event occurs, only the eligible states are assigned credit for it. One way to think of eligibility traces is as a short-term memory that decays over time. The magnitude of the trace determines how eligible that state is for a reward [SB98].

For example, consider the history $h = a_1 o_1 \ldots a_t o_t \ldots a_T o_T$, and the probe g that looks for the last occurrence of the goal. Let's assume that the observation o_t is in fact the goal. This observation will be given a large positive reward, which will then decay exponentially as time increases towards T. Clearly, time steps where the observation is not the goal will have a significantly smaller reward, as they will not have been marked as eligible.

With this in mind, we will use $\langle h|t\rangle_f$ to define the prediction of a probe f given a future test t, a history h and its corresponding belief state b_h. Formally,

$$\langle h|t\rangle_f = \sum_{s \in S} b_h(s)\langle s|t\rangle f(obs(t))$$

where $obs(t)$ is the sequence of observations of the test t.

In order to effectively collapse the space of observable behaviours, we define an equivalence relation on future tests: two tests t_1 and t_2 are equivalent ($\sim_f$) given a probe f if $\forall h \in H, \langle h|t_1\rangle_f = \langle h|t_2\rangle_f$. The idea here is that if two histories make the same predictions about all tests, then the two tests are indistinguishable, given the probe. We will use the term *f-equivalence classes* to refer to classes constructed using probes on tests.

Similarly, we can define an equivalence relation on histories: two histories h_1 and h_2 are equivalent ($\sim_g$) given a probe g if $g(h_1) = g(h_2)$. We will use the term *g-equivalence class* in these case.

Duality Framework

Duality is defined as a transformation applied to a structure that, when applied twice, yields either the original structure, or something isomorphic to it. The POMDP duality construction developed by Hundt et al. [HPPP06] uses a similar notion of equivalence classes of tests (without probes). In the paper, the dual representation is constructed by switching the role of states and observations in the original POMDP. Repeating the transformation provides the double dual machine, which is a deterministic machine whose states describe the behaviour of the original POMDP.

We now proceed to define an analogous construction on history machines, with respect to a set of probes. The dual representation can be defined as a tuple

$$\mathcal{H}' = (H', A, T', \delta', \varphi')$$

where

i. $H' = T/\sim_f$ are the equivalence classes of tests in the original machine

ii. $T' = H$, the histories in the original machine

iii. $\delta'_{ao}([t]_{\mathcal{H}}) = [aot]_{\mathcal{H}}$

iv. $\varphi'([t]_{\mathcal{H}}, h) = \varphi(h, t)$

It is important to notice that the dual machine is a transition system as well. The proof of the following theorem is deferred to the appendix.

Theorem 1. *The transition function in the dual, δ' is well defined. More specifically, $\forall t_1, t_2 \in H', t_1 \sim_f t_2 \Rightarrow \forall ao,\ aot_1 \sim_f aot_2$*

Switching the role of tests and histories again, we obtain the dual of the previous machine, or double dual of the original machine: a tuple

$$\mathcal{H}'' = (H'', A, T'', \delta'', \varphi'')$$

where

i. $H'' = \{T'/\sim_f\}/\sim_g = \{H/\sim_f\}/\sim_g$ are the equivalence classes of tests in the dual

ii. $T'' = H' = T/\sim_f$, are the equivalence classes of tests in the original machine

iii. $\delta''_{ao}([h]_{\mathcal{H}'}) = [hao]_{\mathcal{H}'}$

iv. $\varphi''([h]_{\mathcal{H}'}, [t]_{\mathcal{H}}) = \varphi'([t]_{\mathcal{H}}, h) = \varphi(h, t)$

Like in the case of the dual, we can show that the double dual is well defined.

Theorem 2. *The transition function in the double dual, δ'' is well defined. More specifically,* $\forall h_1, h_2 \in H, h_1 \sim_f h_2 \Rightarrow \forall ao,\ h_1ao \sim_f h_2ao$

The proof can be found in the appendix.

Note that the both the test and histories in the double dual are just equivalence classes of those in the primal. Moreover, the predictions of these equivalence classes matches those for members of the equivalence classes in the primal. This means that the double dual construction produces a machine which is a minimal version of the original machine (as the size of both the history as well as future predictions space has decreased). In addition, given the relationship between POMDPs and history machines, this shows that for any POMDP, there is a construction that produces a deterministic machine, with the same prediction capabilities as the original one, given a set of probes.

Although the result might seem similar to that by Hundt et al., it holds both theoretical and empirical differences. First, the use of probes allows us to filter the space of observations to a partition of interest. This is a significant improvement in the case where it is computationally impossible to process all the existent data. Secondly, because the double dual consists of equivalence classes of tests and histories of the original machine, this approach provides an algorithm to learn its structure directly from data. This option is not available in the original theory, where constructing the double dual requires encoding the space of tests and histories, of infinite size, which is not a feasible algorithmic solution.

Algorithm

The original duality framework was a purely conceptual one, that required both the dual and double dual machines to be constructed before obtaining the minimal POMDP [HPPP06]. This is only feasible for deterministic POMDPs, where the space of tests (and thus equivalence classes of tests) does not explode.

We now present an algorithm that constructs the double dual history machine from action-observation trajectories generated from a POMDP, given test and history probes.

The algorithm is based on two successive clustering processes. First, the original data is filtered using the test probe, and f-equivalence classes are formed. This are classes of histories that are indistinguishable to the agent, i.e. predict all tests with the same accuracy. The granularity of the clustering is variable, and allows for approximate constructions. This clustering gives the states and observations of the machine. Once these are formed, a second layer of clustering is performed, this time using the probe on histories here. The idea here is to further minimize the size of the final history machine, while maintaining the predictive power. It is important to note here that the choice of history probes greatly affects the correctness of predictions of the machine. Finally, transitions are added in a natural way, by concatenating all possible action-observation pairs.

For space purposes we only show the clustering part of the algorithm. To begin, we assume that the $\langle h|t\rangle$ $(\forall h, t)$ predictions (i.e. without the probes) have already been estimated from data. This can be done very easily through counting:

$$\langle h|t\rangle = \frac{\text{\# of times h as been followed by t}}{\text{\# of times h has been followed by the action sequence of t}}$$

Once clustering has been done, adding the transitions is a simple search algorithm over the contents of equivalence classes.

Algorithm 1 Cluster histories using the test/history probes

```
// Calculate predictions
for all h,t do
  $\langle h|t\rangle_f = \langle h|t\rangle \times f(obs(t))$
end for
S = $h_1$
O = all tests t
// Cluster histories based on the probe on tests (f)
for all $h_1 \in S, h_2 \in H$ do
  if $\forall t$ tests, $\langle h_1|t\rangle_f = \langle h_2|t\rangle_f$ then
    // Already in an equivalence class
  else
    // Form new equivalence class
    S = S $\cup h_2$
  end if
end for
// Cluster histories based on the probe on histories (g)
S' = $h_1$
for all $h_1 \in S', h_2 \in S$ do
  if $g(h_1) = g(h_2)$ then
    // Already in an equivalence class
  else
    // Form new equivalence class
    S' = S' $\cup h_2$
  end if
end for
```

It is also interesting to note that f and g can be applied to the space of histories in any order, with considerably different results. The probe on tests is very specific and easy to construct, if the set of observations of interest is known. In most AI environments, this could be the observation denoting a goal state, for example. However, the probe on histories is entirely given by heuristics and an insight in the domain. Thus, applying it before the probe on tests can significantly affect the correctness of the machine.

Empirical results

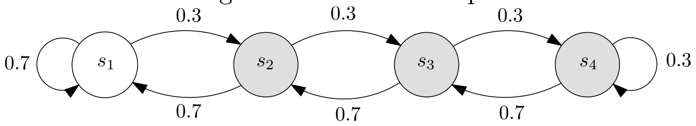
We present two examples to illustrate the effectiveness of the algorithm. The first is a small probabilistic domain, while the second is a significantly larger, deterministic domain.

1. Tunnel world

Consider the system below, with 4 states and one action, that floats from state i to state $i-1$ with probability p, and to state $i+1$ probability $1-p$. There are two deterministic observations: dark (D) and light (L): light is emitted by the leftmost state, while all other states emit dark. Finally, the starting state is always s_4.

The implementation details of the probes are as follows:

Figure 1: Tunnel Example

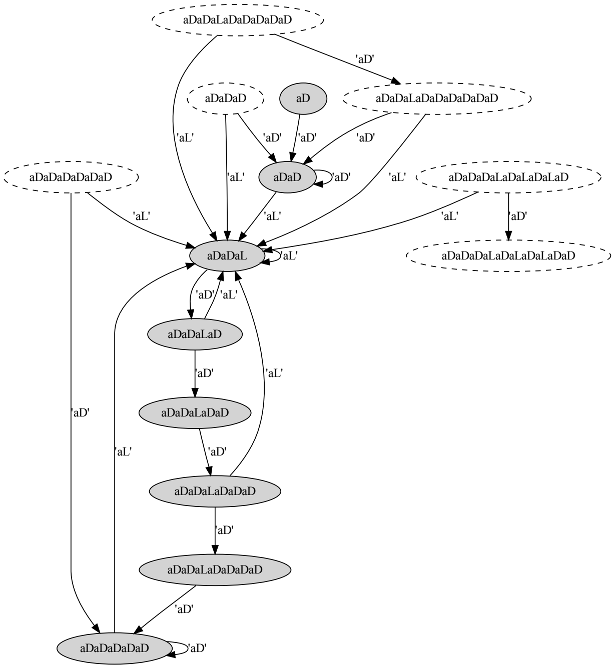


1. For a given test t, $f(obs(t)) = 1$ if t contains at least one “L” observation, and 0 otherwise.
2. The history probe behaves like a complicated eligibility trace:
   - i. If h ends in L, then $g(h) = R_1$
   - ii. Otherwise, let the length of h be n, and let j be the index (from the end) when L was last observed. $g(h) = \sum_{i=j+1}^{n} \gamma^i \times R_2$
   - iii. R_1 is a large reward, R_2 is a significantly smaller one, and γ is a discount factor.

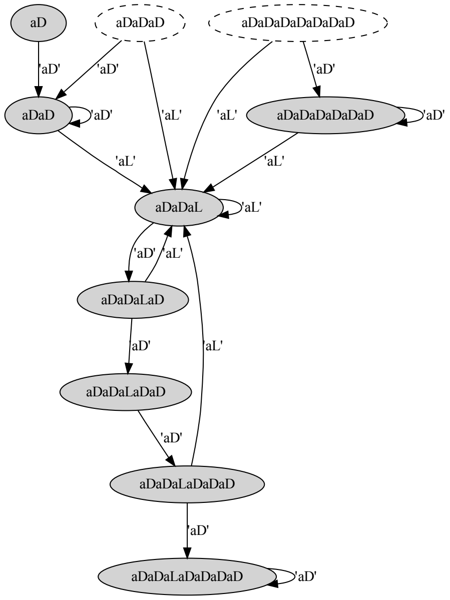
First, we will show a result illustrating the structure of the final machine, followed by a smaller example demonstrating the predictions of future tests. Figure 2. shows the machine constructed by the algorithm when the data consists of history trajectories of length 10, and tests of length 4. The states shaded in gray represent the final states of the machine, after histories were probed, while the dotted states represent states that were present after applying the probe on tests, but that were merged when histories were probed. The figure does not show dead states (i.e. states without incoming or outgoing links). Figure 3. shows the final machine constructed from data containing history trajectories of length 8, and tests of length 2. Finally, Table 1. shows the observation probabilities for each of the final states in this machine. Recall that in this model, if a test t does not contain an observation of L, then $\forall h, \langle h|t\rangle_f = 0$.

A reassuring result is that as the length of the history trajectories grows, only the number of ”dotted states” increases, whereas the effective size of the final machine remains the same. This means that the size of the learnt machine does not explode as the size of the data increases, as the tunnel world can be approximated by a finite number of states.

Intuitively, the double dual looks like chain of histories, where at each step more information is gained about the possible state of the world. It is normal that all states have a path towards the ”aDaDaL” state, as any history ending in the ”Light” observation has determined precisely the state of the world.



(a) Figure 2: Double dual, $|histories| = 10, |tests| = 4$



(b) Figure 3: Double dual, $|histories| = 8, |tests| = 2$

Table 1: Double dual observations, $|histories| = 8, |tests| = 2$

h	$\langle h\|aD\rangle_f$	$\langle h\|aL\rangle_f$	$\langle h\|aDaD\rangle_f$	$\langle h\|aDaL\rangle_f$	$\langle h\|aLaD\rangle_f$	$\langle h\|aLaL\rangle_f$
aD	0	0	0	0.344	0	0
aDaD aDaDaD	0	0.256	0	0.184	0.077	0.179
aDaDaL	0	0.697	0	0.211	0.210	0.489
aDaDaLaD	0	0.157	0	0.266	0.047	0.109
aDaDaLaDaD	0	0.695	0	0.208	0.209	0.494
aDaDaDaDaDaD =aDaDaDaDaDaDaD	0	0.699	0	0.211	0.209	0.490
aDaDaLaDaDaD	0	0	0	0.405	0	0
aDaDaLaDaDaDaD aDaDaLaDaDaDaDaD	0	0.350	0	0.355	0.105	0.245

2. Coloured walls

The second example we will consider is the deterministic grid world below. There are 36×4 states (position and orientation), three actions, forward (F), turn left(L), turn right(R), and 5 observations (wall colours). The probe on tests checks whether the green” wall has been seen. The history probe is similar to the one described in the tunnel world example, however checking for the green” observation instead of “Light”. Finally, the starting state is always the one shown.

Figure 2: Coloured Walls

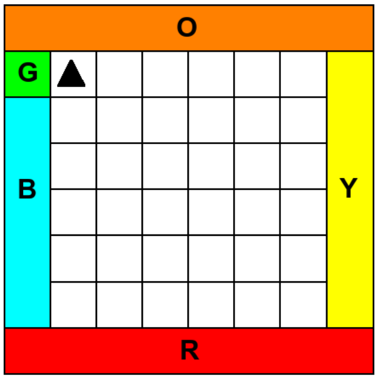
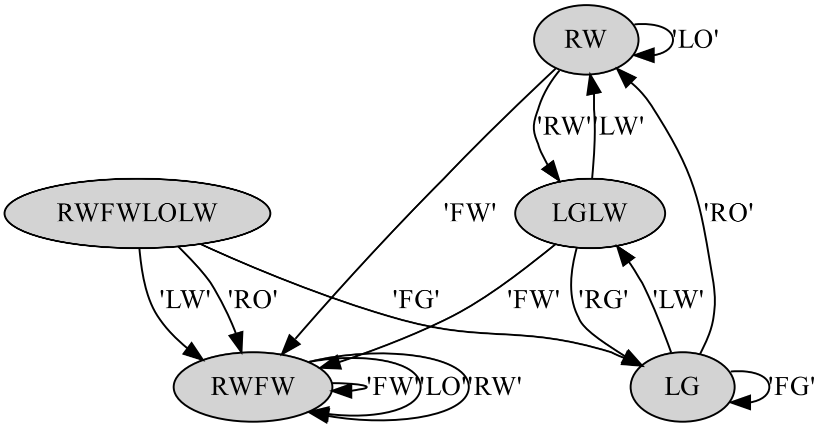


Figure 6. shows the double dual constructed for data consisting of histories of length 6, and the length of predicted tests is 2. To minimize cluttering, we will not show the intermediate model constructed (i.e. the ”dotted” states).

Figure 3: Double dual, $|histories| = 6, |tests| = 2$



One way to read the results of the dual is as follows: given that we know the starting state (top left corner), seeing the history ”RW” (for example) will mean that we are facing right, on the top row. From here, taking a left will automatically mean we have seen the orange wall, and will leave us in a state identical to the previous one, from the point of view of the green wall. In other words, the top row consists of three different states: states that observe ”Green” after turning left (i.e. the top left corner), and the rest of the states that observe an arbitrary colour. Because both probes are only interested in the ”Green” observation, the agent cannot distinguish the other colours from each other.

Conclusion

We have introduced a representation that relates POMDPs to predictive models, based on a duality framework. By allowing the agent to only distinguish between certain features of observations, we have simplified the modeling task, and constructed an internal representation of a system that retains maximal predictive information given available data. The experimental results illustrate that the double dual constructed can be used to make predictions with similar accuracy as the original model. However, the choice of probes, more specifically of history probes, can greatly affect the correctness of its structure. We plan to investigate this further in future work. Additionally, we will also consider the planning problem, and whether a policy learned on the double dual model can be transferred to the original POMDP.

Appendix

Proof of Theorem 1.

$$
\begin{aligned}
\langle h|aot_1\rangle_f &= \sum_{s\in S} b_h(s)\langle s|aot_1\rangle f(\omega((aot_1)) \\
&= \sum_{s\in S} b_h(s) \sum_{s'\in S} \tau_a(s,s')\gamma(s',o)\langle s'|t_1\rangle f(\omega(t_1)) \\
&= C \times \sum_{s'\in S} b_{hao}(s')\langle s'|t_1\rangle f(\omega(t_1)) \\
&= C \times \langle hao|t_1\rangle_f \\
&= C \times \langle hao|t_2\rangle_f && t_1 \sim_f t_2 \\
&= \sum_{s\in S} b_h(s)\langle s|aot_2\rangle f(\omega(aot_2)) \\
&= \langle h|aot_2\rangle_f
\end{aligned}
$$

where C is a normalization constant. ■

Proof of Theorem 2.

$$
\begin{aligned}
\langle h_1ao|t\rangle_f &= \sum_{s\in S} b_{h_1ao}(s)\langle s|t\rangle f(\omega(t)) \\
&= C \times \sum_{s\in S} b_{h_1}(s) \sum_{s'\in S} \tau_a(s,s')\gamma(s',o)\langle s'|t\rangle f(\omega(t)) \\
&= C \times \sum_{s\in S} b_{h_1}(s)\langle s|aot\rangle f(\omega(aot)) \\
&= C \times \sum_{s\in S} b_{h_1}(s)\langle h_1|aot\rangle_f \\
&= C \times \sum_{s\in S} b_{h_2}(s)\langle h_2|aot\rangle_f && h_1 \sim_f h_2 \\
&= \sum_{s\in S} b_{h_2ao}(s)\langle s|t\rangle f(\omega(t)) && \text{as above} \\
&= \langle h_2ao|t\rangle_f
\end{aligned}
$$

where C is a normalization constant. ■

References

[HPPP06] C. Hundt, P. Panangaden, J. Pineau, and D. Precup. Representing systems with hidden state. In *The Twenty-First National Conference on Artificial Intelligence(AAAI)*, 2006.

[C92] L. Chrisman. Reinforcement learning with perceptual aliasing: The perceptual distinctions approach. *Proceedings of the Tenth National Conference on Artificial Intelligence (pp. 183-188)*, 1992.

[LSS02] M. Littman, R. Sutton, and S. Singh. Predictive representations of state. In *Advances in Neural Information Processing Systems (pp. 1551-1561)*, 2002.

[SB98] R. Sutton, A.G. Barto. *Reinforcement learning: An introduction.* MIT Press, 1998.

[SJR04] S. Singh, M. James, M.R. Rudary. Predictive state representations: A new theory for modeling dynamical systems. In *Uncertainty in Artificial Intelligence (pp. 512-519)*, 2004.

[SLJPS] S. Singh, M. Littman, N. Jong, S. Pardoe, P. Stone. Learning Predictive State Representations. In *The Twentieth International Conference on Machine Learning (ICML)*, 2003.