State Space Compression with Predictive Representations

Abdeslam Boularias
Laval University
Quebec G1K 7P4, Canada

Masoumeh Izadi
McGill University
Montreal H3A 1A3, Canada

Brahim Chaib-draa
Laval University
Quebec G1K 7P4, Canada

Abstract

Current studies have demonstrated that the representational power of predictive state representations (PSRs) is at least equal to the one of partially observable Markov decision processes (POMDPs). This is while early steps in planning and generalization with PSRs suggest substantial improvements compared to POMDPs. However, lack of practical algorithms for learning these representations severely restricts their applicability. The computational inefficiency of exact PSR learning methods naturally leads to the exploration of various approximation methods that can provide a good set of core tests through less computational effort. In this paper, we address this problem in an optimization framework. In particular, our approach aims to minimize the potential error that may be caused by missing a number of core tests. We provide analysis of the error caused by this compression and present an empirical evaluation illustrating the performance of this approach.

Introduction

A number of representations for dynamical systems have been proposed during the past two decades beside POMDPs. However, they either impose restrictions to the underlying environment (i.e., they are not as general as POMDPs), or they do not seem to provide any advantages over POMDPs. Among these representations, PSRs (6) seem appealing and more powerful for several main reasons. First, PSRs are grounded in the sequence of actions and observations of the agent, and hence relate the state representation directly to the agent's experience. Second, PSRs offer a representation for dynamical systems which is as general as POMDPs, and can be potentially more compact than POMDPs (5). Third reason is related to the generalization problem. Indeed, the predictive representation does not rely on a specific physical layout of an unobservable environment, so it has the potential of being useful for fast adaptation to a new similar environment.

An important issue for automated agents concerns learning the model. Learning models of dynamical systems under uncertainty has been the focus of huge research effort for different frameworks. There are two major parts to PSR model learning: finding the set of core tests Q (known as the discovery problem), and learning the weight vectors, or projection vectors m_{ao} and m_{aoq_i} (known as the parameter learning problem). The set Q of core tests can be found by searching for the maximum number of linearly independent tests. The system dynamics matrix (SDM) is a mathematical construct that explains the predictive state representation purely based on observable data. The system dynamics matrix forms the basis for PSR learning algorithms presented in (4; 7; 9; 10; 5). Solving the discovery problem together with learning parameters has been attempted by James and Singh (4). The core tests are found by searching for the linearly independent columns of the SDM. Wolfe et al. (10) presented a modified algorithm for learning and discovery for PSRs in systems without reset, called the suffix-history method. In this approach, the histories with identical suffixes are grouped together for counting. All PSR discovery methods suffer from the fact that generating core tests is very much related to computing the SDM. Estimating the prediction probabilities for each entry of the SDM usually requires a large number of test/history samples. Moreover, computing SDM entries approximately makes the computation of the rank of this matrix numerically unstable.

The method presented in (6), incrementally finds all core tests, given the POMDP model of the environment. Although this method is not data driven, it still requires rank computations for huge matrixes in order to find the exact PSR model (complete set of core tests) and is practical only for small domains. From the perspective of planning and generalizing across tasks, learning and using PSRs can be quite advantageous even when the POMDP model is already available. In light of the difficulty of the learning problem for PSRs, it is natural to look for approximation algorithms which generate a subset of core tests more appropriate for planning purposes. The intractability of planning in partially observable systems is due in part to the size of the domain. The number of variables needed to represent the state space, action space, and observation space has a great impact on the efficiency of planning. A typical means of overcoming such situations is to make a smaller and more efficient approximate models. On the other hand, we are interested in reducing the error in these approximations as much as possible. These two goals involve a trade-off between the size of the model and the accuracy of the solution that can be

obtained. The reduced PSR model, which contains only a subset of core-tests, conveys this trade-off. In this paper, we address the issue of dynamically generating approximate PSR models. We follow the underlying assumption in (6) to have access to the POMDP model. However, we are interested in a lossy compressed PSR with k number of core tests (fewer than the number of POMDP states). We formulate this problem as an optimization problem that minimizes the loss function related to prediction of observations and rewards and show that the approximation errors with respect to the value function solution and belief estimation are bounded. Our experimental results with this approach are encouraging. In particular, our method can take advantage of the POMDP domains which contain structures in the underlying states, transitions, and observations to represent a very compact model.

Background

In this section we briefly describe POMDPs and PSRs.

Formally, a POMDP is defined by the following components: a finite set of hidden states S; a finite set of actions A; a finite set of observations Z; a transition function $T : S \times A \times S \rightarrow [0,1]$, such that $T(s,a,s')$ is the probability that the agent will end up in state s' after taking action a in state s; an observation function $O : A \times S \times Z \rightarrow [0,1]$, such that $O(a,s',z)$ gives the probability that the agent receives observation z after taking action a and getting to state s'; an initial belief state b_0, which is a probability distribution over the set of hidden states S; and a reward function $R : A \times S \rightarrow \Re$, such that $R(a,s)$ is the immediate reward received when the agent takes action a in hidden state s. Additionally, there can be a discount factor, $\gamma \in (0,1)$, which is used to weigh less rewards received farther into the future.

The sufficient statistic in a POMDP is the belief state b, which is a vector of length $|S|$ specifying a probability distribution over hidden states. The elements of this vector, $b(i)$, specify the conditional probability of the agent being in state s_i, given the initial belief b_0 and the history (sequence of actions and observations) experienced so far.

After taking action a and receiving observation z, the agent updates its belief state using Bayes' Rule:

$$b'_{bao}(s') = P(s'|b,a,o) = \frac{O(a,s',o)\sum_{s\in S} b(s)T(s,a,s')}{P(o|a,b)} \quad (1)$$

where denominator is a normalizing constant and is given by the sum of the numerator over all values of $s' \in S$. The real value reward for taking action a at a belief state b is computed by:

$$R(b,a) = b^T R^a \quad (2)$$

Predictive state representations, as an alternative to POMDPs, are based on testable experiences. The notion of *test*, used in the definition of PSRs, carries the central idea of relating states of the model to verifiable and observable quantities. A test is an ordered sequence of action-observation pairs $q = a_1 o_1 ... a_k o_k$. The *prediction* for a test q, is the probability of the sequence of observations $o_1, ..., o_k$ being generated, given that the sequence of actions $a_1, ..., a_k$ was taken. If this observation sequence is generated, we say that the test succeeds. The conditional probability of a test q being successful given that the test is performed after history h is: $P(q|h) = \frac{P(hq)}{P(h)}$. A set of tests Q is a PSR of a dynamical system if its prediction, which is called the *prediction vector*, $P(Q|h)$, forms a sufficient statistic for the system after any history h, i.e., if a prediction for any test q at any history h can be computed based on $P(Q|h)$: $P(q|h) = f_q(P(Q|h))$, where $f_q : [0,1]^{|Q|} \rightarrow [0,1]$. The state update operator can be written as: $P(q|hao) = \frac{f_{aoq}(P(Q|h))}{f_{ao}(P(Q|h))}$.

The size of the model, or the number of extension tests, is proportional to the size of the set Q. The number of core tests, $|Q|$, is called the *dimension* of the model. The PSR representation of a dynamical system has at most a number of core test equal to the number of hidden states in the POMDP representation (6). In fact, the PSR model is potentially more compact than the corresponding POMDP. A *linear-PSR* is a PSR in which there exists a projection vector m_q for any test q such that: $P(q|h) = P(Q|h)^T m_q$. A linear PSR model consists of:

- A : finite set of actions;
- O : finite set of observations including rewards;
- Q : finite set of selected tests $\{q_1, q_2, ..., q_k\}$ (core tests);
- m_{ao} : weight vectors for projections of one-step tests, defined for each action $a \in A$ and each observation $o \in O$;
- m_{aoq_i} : weight vectors for projections of one-step extensions of core tests, defined for each action $a \in A$, each observation $o \in Z$ and each core test $q_i \in Q$.

The matrix containing the predictions of all core tests given the underlying states is called the U-matrix ($|S| \times |Q|$). The exact U can be found by searching for the maximum number of linearly independent tests through rank computation given the POMDP model of the environment (6). Predictive states in PSRs are mapped to belief states in POMDPs through the definition of U:

$$P(Q|h) = b^T U \quad (3)$$

Reward in PSRs is considered as part of the observation. In this paper, we assume that exists a finite set of real value rewards, θ. The observable feedback, $o = rz$, contains the discrete reward value $r \in \theta$ and the observation $z \in Z$, as defined in POMDPs. We can associate a scalar reward for each action a at each prediction vector$P(Q|h)$:

$$R(P(Q|h),a) = \sum_r r p(r|P(Q|h),a) = \sum_r \sum_{o\in Z} r P(Q|h)^T m_{ao} \quad (4)$$

To simplify the notations, we let $m_{ar} = r\sum_{o\in Z} m_{ao}$. Therefore, the above equation is:

$$R(P(Q|h),a) = \sum_r P(Q|h)^T m_{ar} \quad (5)$$

Linear PSRs as lossless representations for compressed POMDPs

Linear PSRs are able to find special type of structure called *linear dependency* (1) in dynamical systems and discover the

reduced model for given input POMDPs. This structure is in fact a property of the underlying states of the MDP model. A linearly dependent state in an MDP is defined to be the one whose transitions are expressible by a linear combination of the ones from other states. Linear state dependency is a sufficient condition for the linear PSR to have smaller dimensionality without losing any information. Considering the reward as a part of observation, the compression provided by PSRs will preserve the dynamics together with the values (1).

While lossless compressions preserve the dynamics of a system in the reduced model, there are other types of compressions which are more relaxed and only preserve properties useful for decision making. The value-directed compression (VDC) algorithm (8) is an example of this type. VDC computes a low dimensional representation of a POMDP directly from the model parameters: R, T, and O by finding a *Krylov* subspace for the reward function under propagating beliefs. The Krylov subspace for a vector and a matrix is the smallest subspace that contains the vector and is closed under multiplication by the matrix. VDC uses a transformation matrix $F = \{R, TR, T^2R, ...\}$ to create a reduced model and preserves the optimal state-action value function.

The value-directed compression method has differences as well as similarities to predictive state representations. Both PSRs and VDC provide linear compression and they are both considered as lossless compression methods. Their approach for recursively growing a set of sufficient variables for prediction is identical. However, they focus on preserving slightly different properties in their compression. In VDC, belief states in the original model might not be correctly recovered but PSRs ensure the accurate prediction of all future observations. Therefore, a next belief state in the original POMDP can be correctly recovered, as described in Equation 3. If reward is considered as part of observation, then PSRs focus on the accurate prediction of observations and rewards. The transformation matrix for VDC, F, can be thought of as a change of basis for the value function, whereas the transformation matrix $F = U^T$ for the PSR model can be viewed as change of basis for the belief space.

Lossy compression with Predictive State Representations

Exact PSRs do not seem to provide more than linear dimensionality reduction. Linear lossless compression is still considered insufficient in practice. This is a motivation to further investigate predictive models that can answer more task-specific questions, but perhaps scale better. Building on the lossy compression version of VDC, we develop an algorithm for learning compact PSRs. Algorithm 1 illustrates this idea. Given a POMDP model and the required dimension for the corresponding PSR approximate model, the algorithm finds the best parameters of an approximate PSR model that minimize the loss in rewards and in observation probabilities. We use the following LPs to find parameters of the approximate PSR:

$$\textit{minimize:} \qquad c_1 \sum_{ao} \varepsilon_{ao} + c_2 \sum_{aoq} \varepsilon_{aoq} \tag{6}$$

subject to:

$$\begin{aligned} &\forall a \in A, \forall o \in Z: \\ &\quad \|T^a O^{ao} e - U m_{ao}\|_\infty \leq \varepsilon_{ao} \\ &\forall a \in A, \forall o \in Z, \forall q \in Q: \\ &\|T^a O^{ao} U(.,q) - U m_{aoq}\|_\infty \leq \varepsilon_{aoq} \end{aligned}$$

$$\textit{minimize:} \qquad c_3 \varepsilon_{ar} \tag{7}$$

subject to:

$$\forall a \in A, \forall r \in \theta \|R^a - U m_{ar}\|_\infty \leq \varepsilon_{ar}$$

where e is the unite vector transpose. Here q is a column of the approximate U, corresponding to an approximate outcome of a core test. But the actual test representation, as an ordered sequence of action-observation pairs, is not important and is not computed here. We alternate between solving the LPs presented in Equations 6 and 7, which adjust the parameters m_{ao}, m_{ar}, and m_{aoq} while keeping the U fixed, and solving these LPs which adjusts U while keeping the parameters m_{ao}, m_{ar}, and m_{aoq} fixed. Convergence of the algorithm is guaranteed just as argued in (8) since the objective function decreases at each iteration. However, the resulting fixed point may not be a global or a local optimum.

Algorithm 1 PSRs Approximate Constructing

Require: POMDP model (S, A, Z, T, O, R); dimension $|Q| \leq |S|$; the number of iterations I.
Initialize U with random values.
$i = 0$.
repeat
1. Solve the LP in 6 for variables m_{ao} and m_{aoq}, and constant U.
2. Solve the LP in 6 for variable U and constants m_{ao}, and m_{aoq}.
3. $i = i + 1$

until $i = I$
$i = 0$.
repeat
1. Solve the LP in 7 for variables m_{ar} and constant U.
2. Solve the LP in 7 for variable U and constants m_{ar}.
3. $i = i + 1$

until $i = I$
RETURN: The parameters of the compressed PSR model.

We need to try several random initializations before settling for a solution. In practice, we may initialize the matrix U with first set of linearly independent vectors found by search as in (6). This algorithm avoids rank computation

which is the main drawback in learning PSRs. The algorithm proposed by McCracken and Bowling (7) also avoids the rank estimation and instead uses a gradient descent approach to estimate predictions of tests in an online fashion.

Analysis of the approximation error

Value function error Let $\varepsilon^*_{ar} = \max_{a\in A} \varepsilon_{ar}$. We define error in the value function for a given belief point $b \in B$ in horizon i, $\varepsilon_{v_i}(b)$, to be the difference between value of b according to the optimal POMDP value function of horizon i, $V^*_i(b)$, and value of the corresponding state in approximate PSR value function of horizon i, $\hat{V}^*_i(\hat{U}^T b)$. Also, let $\varepsilon^*_{v_i}$ be the worst such error over the entire belief space.

$$\varepsilon^*_{v_i} = max_{b\in B}\varepsilon_{v_i}(b) = max_{b\in B}|b^T(\alpha^*_i - \hat{U}\beta^*_i)| \qquad (8)$$

We need to show this error is bounded. We denote value function of the POMDP model by a set of α vectors, value function of the approximate PSR by a set of β vectors, action correspond to the best α vector at b by a^*, and action correspond to the best β vector at $\hat{U}^T b$ by a^*_β.

$$\begin{aligned}
\varepsilon^*_{v_i} &= |b^T(\alpha^*_i - \hat{U}\beta^*_i)| \\
&= |b^T R^{a^*} - b^T\hat{U}\hat{m_{a^*_\beta}}r + \gamma\sum_o P(o|a^*,b)b^T_{a^*o}\alpha^*_{i-1} \\
&- P(o|a^*_\beta,\hat{U}^T b)\hat{U}\beta^*_{i-1}| \\
&\le |b^T R^{a^*} - b^T\hat{U}\hat{m_{a^*}}r + \gamma\sum_o P(o|a^*,b)b^T_{a^*o}(\alpha^*_{i-1} - \hat{U}\beta^*_{i-1})| \\
&= |b^T R^{a^*} - b^T\hat{U}\hat{m_{a^*}}r + \gamma\sum_{o\in O} P(o|a^*,b)\varepsilon_{v_{i-1}}| \\
&\le |\varepsilon^*_{ar} + \gamma\varepsilon_{v_{i-1}}\sum_{o\in O} P(o|a^*,b)| = \varepsilon^*_{ar} + \gamma\varepsilon_{v_{i-1}}
\end{aligned}$$

It is anticipated that the error for v_0 is bounded:

$$\begin{aligned}
\varepsilon_{v_0} &= |b^T(\alpha^*_0 - \hat{U}\beta^*_0)| \\
&\le \|b\|_1\,\|\alpha^*_0 - \hat{U}\beta^*_0\|_\infty \text{ (Holder inequality)} \\
&= \|\alpha^*_0 - \hat{U}\beta^*_0\|_\infty \text{ (b is a probability vector)} \\
&\le \max_{a\in A}\|R^a - \hat{U}\hat{m_{ar}}\|_\infty \\
&\le \max_{a\in A}\varepsilon_{ar} = \varepsilon^*_{ar}
\end{aligned}$$

Therefore:

$$\varepsilon^*_{v_i} \le \varepsilon^*_{ar}\frac{1-\gamma^n}{1-\gamma} \qquad (9)$$

Belief estimation error We show a bound on the maximum divergence between a belief state b^t and belief state $\tilde{b}^t$ inferred from the approximate PSRs: $\varepsilon^t_b = \|\tilde{b}^t - b^t U\|_\infty$. Let b^0 be the initial belief state, the corresponding approximate belief by the reduced PSR be: $\tilde{b}^0 = b^0\tilde{U}$, maximum error on predicting an observation o be: $\varepsilon^t_{ao} = |\tilde{b}^t m_{ao} - b^t T^a O^{ao}|$, and maximum error on predicting an observation o and a core test q be: $\varepsilon^t_{aoq} = |\tilde{b}^t m_{aoq} - b^t T^a O^{ao} U(.,q)|$.

At time $t=0$, $\varepsilon^0_b = \|\tilde{b}^0 - b^0 U\|_\infty = 0$ since the initial beliefs are identical. Also, $\varepsilon^0_{ao} = |\tilde{b}^0 m_{ao} - b^0 T^a O^{ao} e| = |b^0 U m_{ao} - b^0 T^a O^{ao} e| \le \varepsilon_{ao}$ and $\varepsilon^0_{aoq} = |\tilde{b}^0 m_{aoq} - b^0 T^a O^{ao} U(.,q)| = |b^0 U m_{aoq} - b^0 T^a O^{ao} U(.,q)| \le \varepsilon_{aoq}$.

Let x be the probability to observe o followed by the test q after executing the action a, $x = b^t T^a O^{ao} U(.,q)$, and y be the probability to observe o after executing the action a, $y = b^t T^a O^{ao} e$. Therefore $x \le y$. At time $t+1$ we have:

$$\begin{aligned}
\varepsilon^{t+1}_b &= \|\tilde{b}^{t+1} - b^{t+1}U\|_\infty \\
&= \max_{q\in Q}|\tilde{b}^{t+1}(q) - b^{t+1}U(.,q)| \\
&= \max_{q\in Q}|\frac{\tilde{b}^t m_{aoq}}{\tilde{b}^t m_{ao}} - \frac{b^t T^a O^{ao} U(.,q)}{b^t T^a O^{ao} e}| \\
&\le \max_{q\in Q}\{\frac{b^t T^a O^{ao} U(.,q) + \varepsilon^t_{aoq}}{b^t T^a O^{ao} e - \varepsilon^t_{ao}} - \frac{b^t T^a O^{ao} U(.,q)}{b^t T^a O^{ao} e} \\
&\quad , \frac{b^t T^a O^{ao} U(.,q)}{b^t T^a O^{ao} e} - \frac{b^t T^a O^{ao} U(.,q) - \varepsilon^t_{aoq}}{b^t T^a O^{ao} e + \varepsilon^t_{ao}}\} \\
&\le \max_{\substack{x,y\in[0,1] \\ x\le y, x\ge\varepsilon^t_{aoq}, y>\varepsilon^t_{ao}}} \max\{\frac{x+\varepsilon^t_{aoq}}{y-\varepsilon^t_{ao}} - \frac{x}{y}, \frac{x}{y} - \frac{x-\varepsilon^t_{aoq}}{y+\varepsilon^t_{ao}}\}
\end{aligned}$$

The first inequality is because: $\tilde{b}^t m_{aoq} \in [b^t T^a O^{ao} U(.,q) - \varepsilon^t_{aoq}, b^t T^a O^{ao} U(.,q) + \varepsilon^t_{aoq}]$ and $\tilde{b}^t m_{ao} \in [b^t T^a O^{ao} e - \varepsilon^t_{ao}, b^t T^a O^{ao} e + \varepsilon^t_{ao}]$.

We can use partial derivation to prove that, under the constraints $x, y \in [0,1], x \le y, x \ge \varepsilon^t_{aoq}$ and $y > \varepsilon^t_{ao}$, we have:

$\frac{x+\varepsilon^t_{aoq}}{y-\varepsilon^t_{ao}} - \frac{x}{y} \le \frac{\varepsilon^t_{ao}+\varepsilon^t_{aoq}}{1-\varepsilon^t_{ao}}$ and $\frac{x}{y} - \frac{x-\varepsilon^t_{aoq}}{y+\varepsilon^t_{ao}} \le \frac{\varepsilon^t_{ao}+\varepsilon^t_{aoq}}{1-\varepsilon^t_{ao}}$

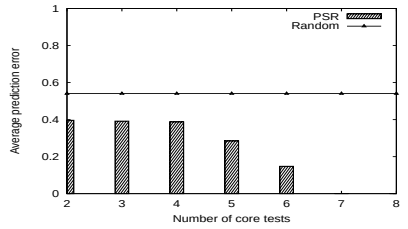
Therefore:

$$\varepsilon^{t+1}_b \le \frac{\varepsilon^t_{ao} + \varepsilon^t_{aoq}}{1-\varepsilon^t_{ao}}$$

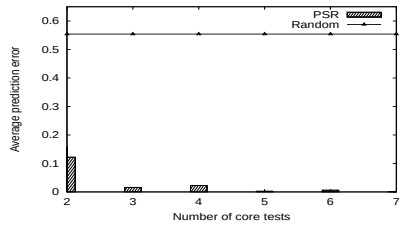
Let us define $sgn(m_{ao})$ as the vector indicating the sign of every entry of m_{ao} (i.e. $sgn(m_{ao})(q) = 1$ if $m_{ao}(q) \ge 0$ and $sgn(m_{ao})(q) = -1$ if $m_{ao}(q) < 0$).

$$\begin{aligned}
\varepsilon^{t+1}_{ao} &= |\tilde{b}^{t+1} m_{ao} - b^{t+1} T^a O^{ao} e| \\
&\le \max\{|(b^{t+1}U + sgn(m_{ao})\varepsilon^{t+1}_b)m_{ao} - b^{t+1}T^a O^{ao} e| \\
&\quad , |(b^{t+1}U - sgn(m_{ao})\varepsilon^{t+1}_b)m_{ao} - b^{t+1}T^a O^{ao} e|\} \\
&\le \max\{|(b^{t+1}U m_{ao} - b^{t+1}T^a O^{ao} e) + sgn(m_{ao})\varepsilon^{t+1}_b m_{ao}| \\
&\quad , |(b^{t+1}U m_{ao} - b^{t+1}T^a O^{ao} e) - sgn(m_{ao})\varepsilon^{t+1}_b m_{ao}|\} \\
&\le \max\{\varepsilon_{ao} + |sgn(m_{ao})\varepsilon^{t+1}_b m_{ao}| \\
&\quad , \varepsilon_{ao} + |-sgn(m_{ao})\varepsilon^{t+1}_b m_{ao}|\} \\
&\le \varepsilon_{ao} + |sgn(m_{ao})\varepsilon^{t+1}_b m_{ao}| \\
&\le \varepsilon_{ao} + m^*_{ao}\varepsilon^{t+1}_b \text{ where } m^*_{ao} = \max_{a\in A, o\in O} sgn(m_{ao})m_{ao}
\end{aligned}$$

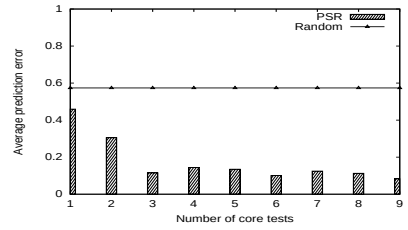
We can also follow the same steps to prove that: $\varepsilon^{t+1}_{aoq} \le \varepsilon_{aoq} + m^*_{aoq}\varepsilon^{t+1}_b$ where $m^*_{aoq} = \max_{a\in A, o\in O, q\in Q} sgn(m_{aoq})m_{aoq}$, which means: $\varepsilon^{t+1}_{ao} \le \varepsilon_{ao} + m^*_{ao}\frac{\varepsilon^t_{ao}+\varepsilon^t_{aoq}}{1-\varepsilon^t_{ao}}$, $\varepsilon^{t+1}_{aoq} \le \varepsilon_{aoq} + m^*_{aoq}\frac{\varepsilon^t_{ao}+\varepsilon^t_{aoq}}{1-\varepsilon^t_{ao}}$, $\varepsilon^{t+1}_b \le \frac{\varepsilon^t_{ao}+\varepsilon^t_{aoq}}{1-\varepsilon^t_{ao}}$ $\square$.



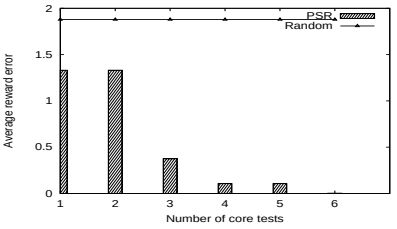
(a)



(b)



(c)



(d)

Figure 1: The average error on predicting observations as a function of the number of core tests in: (a) shuttle, (b) network, (c) 4x4 grid, (d) robot coffee problems.

Empirical evaluation

A complete model of a system should be able to predict the probability of an arbitrary observation o, after a history h. The POMDP model can make such predictions using its internal state update function, given by Equation 1, and the observation prediction: $P(o|a,b) = \sum_{s\in S} b(s) \sum_{s'\in S} T(s,a,s')O(a,s',o)$. The approximate PSR calculates the probability of the future observation o by: $\hat{P}(o|a,h) = \hat{P}(Q|h)^T \hat{m}_{ao}$. We use these equations as the baseline to evaluate the accuracy of predictions using the approximate PSR model and compare it to the predictions of the POMDP. The initial belief state b_0 of the POMDP is generated randomly, and the initial core test probabilities $\hat{P}(Q)$ of the approximate PSR are given by $\hat{P}(Q) = b_0^T U$. We choose a random action a, and calculate the predictions $P(o|a,h)$ and $\hat{P}(o|a,h)$, $\forall o \in O$. We sample also the next

Domain	Coffee	Hall	Hall2	SD
dim(time)	1(0.36)	5(20.59)	5(20.46)	2(350.93)
	2(0.56)	10(35.42)	10(41.92)	3(557.09)
	3(0.68)	15(60.87)	15(69.46)	5(759.2)
	4(0.98)	20(74.47)	20(107.28)	8(1166.04)
	5(1.15)	25(94.93)	25(157.73)	10(1459.56)
	6(1.18)	30(114.9)	30(184.23)	

Table 1: The runtime of algorithm 2 in seconds for the coffee domain (rewards and observations), the hallway and the hallways2 problems (observations only). For the coffee domain, 2 is the maximum number of core tests that we can find.

underlaying state and generate an observation o according to T and O matrixes. The observation o and the action a are used by both the POMDP and the approximate PSR to update their internal states. The average prediction error computed by: $E^2 = \frac{1}{t\times|Z|}\sum_{i=1}^{t}\sum_{o=1}^{|Z|}(\hat{P}_i(o) - P_i(o))^2$.

In our experiments, we have considered $t = 100$. The error on the predictions become more important as the horizon grows, this is due to the fact that the update of state for the PSR uses approximate values, therefore, after long train the cumulated error makes that the PSR state far from the POMDP belief state. The parameters c_1 and c_2 are set to 1, as we found there is no significant difference in the average error when we consider different values of c_1 and c_2. The maximum number of iterations of Algorithm 1 is set to 20. In practice, the parameters m_{ao}, m_{aoq} and U converge often to a fixed point before 10 iterations. We repeated the algorithm 5 times for every problem.

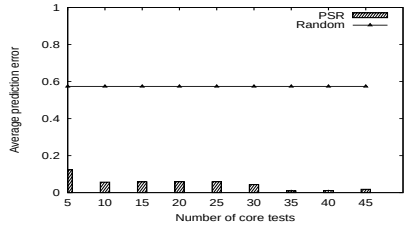
We tested our discovery and learning algorithm on several standard POMDP problems used in (8; 3): the shuttle(8 states, 3 actions, 5 observations), network(7 states, 4 actions, 2 observations), 4x4 grid(16 states, 4 actions, 2 observations), robot coffee domain(32 states, 2 actions, 3 observations), the spoken dialogue domain (SD)(433 states, 37 actions, 16 observations), and hallways(with 60 and 90 states respectively). Figures 1 and 2 present the average error E as function of the number of core tests used for every problem. The error E is contained in the interval $[0,1]$ because it measures a difference between two probabilities.

In all the experiments, the compression of the POMDP was successful, as we needed only a few core tests to construct an approximate PSR model with low prediction error. For the SD problem for example, we can use only 8 core tests instead of 433 to make predictions on future observations with an average error of 0.08 over the 100 time steps. This is very promising. Of course this domain contains a lot of structure (see (8)).

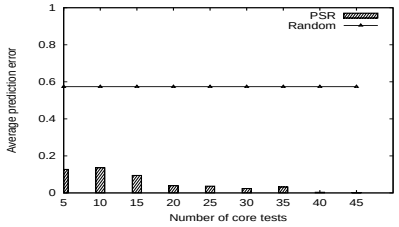
The random prediction makes an average error of 0.55 in most of the other problems. In the shuttle problem, the average error become null when the number of core tests used is 7, which is a gain itself, even not considerable. The results for the network are very promising, we can predict the future observations of the system by using only 3 tests, while the prediction error is nearly null (0.0159). For the coffee domain, we have included the rewards in the set of observations, because this problem contains lot of terminal and equivalent states, and can be represented with only 2

core tests, if rewards are not taken in consideration. So, we tested the performance of the compressed model on the prediction of observations and immediate reward at the same time. With only 7 core tests, the PSR model is able to make predictions with an average error of 0.0333.

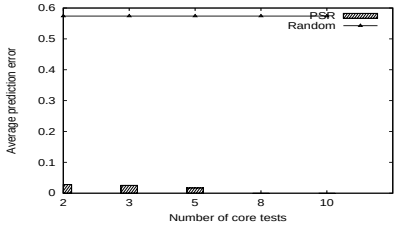
We remark also that in general, the average error decreases when the number of core tests grows. This is natural, because with more variables the LP has higher freedom degree, and can adjust the values in order to minimize the objective function better. The error become null when $|Q| = |S|$. In some special cases, the average error E increases slightly when we increase the number of core tests, this is due to the fact E depends on the random initialization of U, and E correspond to a local minima in these particular cases.



(a)



(b)



(c)

Figure 2: The average error on predicting observations as a function of the number of core tests in: (a) hallway, (b) hallway2, (c) spoken dialogue problems.

Conclusions and future work

The existing works on exact learning methods for PSRs demonstrate that the intractability of finding an exact model for POMDPs remains the same for the case of PSRs. In this paper, we investigated the possibility of finding an approximate PSR model. We formulated this problem in linear programming frameworks. Using approximate model learning seems to have a definite advantage in PSRs as confirmed by our experimental results. Our results illustrates that the reduced model can predict the observations and rewards probabilities with high precision compared to the exact model. The impact of this method is more pronounced for problems with special structure. The approximate PSR representation provides similar compression as lossy value-directed compression for POMDPs. However, there are more potential advantageous in applying PSRs instead of POMDPs for planning and plan generalization. The immediate next step to this research is to use the approximate models resulted from the algorithm in planning to target the curse of dimensionality in decision making under uncertainty. In this paper, we illustrated the theoretical bound for value function error using the approximate PSR model. However, we were not able to measure this error empirically, as PSR planning is still an open problem and there exist only attempts to extend POMDP solution methods to PSRs (2).

References

1. M. T. Izadi and D. Precup Model Minimization by Linear PSR. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2005.

2. M. T. Izadi On knowledge representation and decision making under uncertainty. *Ph.D. Thesis*, McGill University, 2007.

3. Anthony R. Cassandra Exact and Approximate Algorithms for Partially Observable Markov Decision Processes. *Ph.D. Thesis*, Brown University, 1998.

4. M. R. James and S. Singh. Learning and discovery of predictive state representations in dynamical systems with reset. In *Proceedings of the twenty-first international conference on Machine learning (ICML '04)*, pages 53–59, 2004.

5. M. R. James and S. Singh. Predictive State Representations: A New Theory for Modeling Dynamical Systems. In *Uncertainty in Artificial Intelligence: Proceedings of the Twentieth Conference (UAI '04)*, page 512-519, 2004.

6. M. Littman, R. Sutton, and S. Singh. Predictive representations of state. In *Advances in Neural Information Processing Systems 14 (NIPS '01)*, pages 1555-1561, 2002.

7. P. McCracken and M. Bowling. Online discovery and learning of predictive state representations. In *Advances in Neural Information Processing Systems (NIPS '06)*, pages 875–882. MIT Press, Cambridge, MA, 2006.

8. P. Poupart and C. Boutilier. Value-directed compression for pomdps. In *Advances in Neural Information Processing Systems (NIPS '03)*, pages 1547–1554, 2003.

9. E. Wiewiora. Learning predictive representations from a history. In *Proceedings of the 22nd international conference on Machine learning (ICML '05)*, pages 964–971, 2005.

10. B. Wolfe, M. R. James, and S. Singh. Learning predictive state representations in dynamical systems without reset. In *Proceedings of the 22nd international conference on Machine learning (ICML '05)*, pages 980–987, 2005.