Slide 1

Predictive State Representations: An Introduction

Francisco S. Melo
fmelo@isr.ist.utl.pt

Workshop on Reinforcement Learning

Slide 2

Outline of the presentation

- **Background on POMDPs**
- PSRs: What is this all about?
- PSRs vs. other dynamic models
- Discovery, learning and planning in PSRs

Slide 3

Background on POMDPs

To establish the notation, an MDP is a tuple $(\mathcal{X}, \mathcal{A}, \mathsf{T}, r, \gamma)$ where

- $\mathcal{X}$ is the state-space;
- $\mathcal{A}$ is the action-space;
- T represents the transition probability model;
- r represents the reward function;
- γ represents the discount factor.

At all times, the decision-maker has access to the state X_t of the process.

Slide 4

Partially Observable MDPs (POMDPs)

- In a POMDP the state X_t is not accessible;
- The decision-maker receives an observation Z_t that depends on the state X_t and on the previous action A_{t-1};
- The observations Z_t take values in a finite set $\mathcal{Z}$;
- The dependence of Z_t on X_t and A_{t-1} is represented by an *observation model O*:

$$\mathbb{P}\left[Z_t = z \mid X_t = x, A_t = a\right] = \mathsf{O}_a(x, z).$$

The POMDP model is can be represented as a tuple $(\mathcal{X}, \mathcal{A}, \mathcal{Z}, \mathsf{T}, \mathsf{O}, r, \gamma)$.

Slide 5

Optimality in POMDPs

- The decision-maker should still determine a policy π maximizing the expected total discounted reward;
- Since the state X_t is not accessible, the policy can no longer be defined as a mapping $\pi : \mathcal{X} \longrightarrow \mathcal{A}$;
- Instead, *given a model of the POMDP*, the decision-maker can maintain at each time t a *belief* b_t over the state-space:

$$b_t(x) = \mathbb{P}\left[X_t = x \mid H_t\right],$$

  where H_t is the history up to time t;
- This belief works as a (continuous) internal state for the decision-maker.

Slide 6

Optimality in POMDPs (cont.)

- Value-functions can now be defined in terms of beliefs:

$$V(b, \{A_t\}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t \mid b_0 = b\right];$$

- The optimal value function verifies a Bellman-like equation:

$$V^*(b_t) = \max_{a \in \mathcal{A}} \sum_{x \in \mathcal{X}} b_t(x) \left[r(x, a) + \gamma \sum_{y \in \mathcal{X}} \mathsf{T}_a(x, y) \sum_{z \in \mathcal{Z}} \mathsf{O}_a(y, z) V^*(b_{t+1})\right]$$

- The optimal policy is a mapping $\pi^* : \mathbb{B} \longrightarrow \mathcal{A}$.

Slide 7

Algorithms for POMDPs

- POMDPs are PSPACE-hard for finite-horizon[10];
- Exact methods proceed by incrementally "simplifying" the representation of V_t^* [3, 6];
- Function approximation can also be used with RL methods to approximate V^* [7];
- Many other approximated method are available (see [1, 2]).

Slide 8

Outline of the presentation

- Background on POMDPs
- **PSRs: What is this all about?**
- PSRs vs. other dynamic models
- Discovery, learning and planning in PSRs

Slide 9

PSRs: What is this all about?

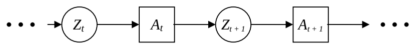
- PSRs were introduced in [8] and further explored in [12, 13];
- Predictive state representations (PSRs) provide a different *dynamical models*;
- Unlike POMDPs, PSRs rely solely on *observable quantities*;
- Building PSR models from observed data should, therefore, be more reliable;
- As we will see, PSRs have larger representational power than other dynamic models.

Slide 10

Futures

We now define some nomenclature:

- A dynamic system generates *sequences* of observations and actions;



- The *future of the system* is any sequence of action-observation pairs from the current time;
- The system can thus be seen as a *probability distribution* over "possible futures";
- A particular finite sequence of action-observation pairs will be referred as a *test*.

Slide 11

Futures (cont.)

- Given a k-length test $T = (a_1, z_1, a_2, z_2, \ldots, a_k, z_k)$, a *prediction* for T is the probability of observing $z_1, \ldots, z_k$ given that the first k actions are $a_1, \ldots, a_k$, *i.e.*,

$$\mathsf{P}(T) = \mathbb{P}\left[Z_1 = z_1, Z_2 = z_2, \ldots, Z_k = z_k \mid A_1 = a_1, A_2 = a_2, \ldots, A_k = a_k\right];$$

- The set of all possible tests for a system is countable;
- It is possible to order the possible tests $T_1, T_2, \ldots$ by increasing order of length;
- We define the *system dynamics vector*, d, as an infinite line vector with ith component $d_i = \mathsf{P}(T_i)$.

Slide 12

Histories

- A *history* of the system is a sequence of occurred action-observation pairs;
- We can define *history-conditional* predictions as

$$\mathsf{P}(T \mid H) = \mathbb{P}\left[Z_1 = z_1, \ldots, Z_k = z_k \mid H, A_1 = a_1, \ldots, A_k = a_k\right];$$

- We now define the *system dynamics matrix*, $\mathcal{D}$, as an infinite matrix with ijth component $\mathcal{D}_{ij} = \mathsf{P}(T_j \mid H_i)$:

$$\mathcal{D}_{ij} = \mathsf{P}(T_j \mid H_i) = \frac{\mathsf{P}(H_i, T_j)}{\mathsf{P}(H_i)}.$$

Slide 13

The system dynamics matrix $\mathcal{D}$

The system dynamics matrix can be visualized as

	T_1	$\dots$	T_n	$\dots$
$H_1 = \emptyset$	$\mathsf{P}(T_1)$	$\dots$	$\mathsf{P}(T_n)$	$\dots$
H_2	$\mathsf{P}(T_1 \mid H_2)$	$\dots$	$\mathsf{P}(T_n \mid H_2)$	$\dots$
$\vdots$	$\vdots$		$\vdots$	
H_m	$\mathsf{P}(T_1 \mid H_m)$	$\dots$	$\mathsf{P}(T_n \mid H_m)$	$\dots$
$\vdots$	$\vdots$		$\vdots$	

Slide 14

Linear dimension of a system

- The *linear dimension* of a system is the rank of its dynamics matrix;
- In a system with linear dimension k, $\mathcal{D}$ has certainly k linearly independent columns;
- Let $Q = \{q_1, q_2, \dots, q_k\}$ be any such k columns;
- The tests $q_1, \dots, q_k$ are known as *core tests*;
- The submatrix obtained from $\mathcal{D}$ by considering only the core tests is denoted $\mathcal{D}(Q)$;

> *The state of a linear PSR given a history H is given by the (line) vector*
>
> $$\mathsf{P}(Q \mid H) = \big[\mathsf{P}(q_1 \mid H), \dots, \mathsf{P}(q_k \mid H)\big].$$

Slide 15

State updates in linear PSRs

- For any test T, there is a parameter vector m_T such that
$$\mathsf{P}(T \mid H) = \mathsf{P}(Q \mid H) m_T;$$
- Given a new action-observation pair, the state of the PSR can be updated componentwise as
$$\mathsf{P}(q_i \mid H, a, z) = \frac{\mathsf{P}(a, z, q_i \mid H)}{\mathsf{P}(a, z \mid H)} = \frac{\mathsf{P}(Q \mid H) m_{(a,z,q_i)}}{\mathsf{P}(Q \mid H) m_{(a,z)}}$$

> *The parameters of the linear PSR are the vectors $m_{(a,z,q_i)}$ and $m_{(a,z)}$, with $a \in \mathcal{A}$, $z \in \mathcal{Z}$ and $i = 1, \dots, k$, in a total of $(k+1)\,|\mathcal{A}|\,|\mathcal{Z}|$ k-vectors.*

Slide 16

Nonlinear PSRs

- The state of the linear PSR allows to determine the prediction for any test T: it is a *sufficient statistic* for the history;
- It may happen that a set of tests $C = \{c_1, \dots, c_m\}$, with $m < k$ such that the corresponding predictions are *nonlinear* sufficient statistics for the history, *i.e.*,
$$\mathsf{P}(T \mid H) = f_T(\mathsf{P}(C \mid H))$$
for some nonlinear function f_T independent of H;
- In this case, the state of the PSR is the vector $\mathsf{P}(C \mid H)$ and can be updated componentwise as
$$\mathsf{P}(c_i \mid H, a, z) = \frac{\mathsf{P}(a, z, c_i \mid H)}{\mathsf{P}(a, z \mid H)} = \frac{f_{(a,z,c_i)}(\mathsf{P}(C \mid H))}{f_{(a,z)}(\mathsf{P}(C \mid H))}.$$

Slide 17

Outline of the presentation

- Background on POMDPs
- PSRs: What is this all about?
- **PSRs vs. other dynamic models**
- Discovery, learning and planning in PSRs

Slide 18

PSRs vs. other dynamic models

The results presented in this section can all be found in [13].

Theorem 1. *A dynamical system described by a POMDP with k states has linear dimension no greater than k.*

Intuition:

- The beliefs work as the POMDP states (sufficient statistics for the history);
- The belief for a k-state POMDP is a k-vector;
- The system dynamics matrix $\mathcal{D}$ can be determined by noticing that, for a test $T = (a_1, z_1, \ldots, a_k, z_k)$,

$$\mathsf{P}(T \mid H) = b(H) \underbrace{\mathsf{T}_{a_1} \mathsf{diag}\left(O_{a_1}(\cdot, z_1)\right) \cdots \mathsf{T}_{a_k} \mathsf{diag}\left(O_{a_k}(\cdot, z_k)\right) \mathbf{1}}_{m_T}.$$

Slide 19

PSRs vs. other dynamic models (cont.)

Corollary 2. *A dynamical system described by a HMM with k states has linear dimension no greater than k.*

Theorem 3. *A dynamical system described by a n-order Markov model has linear dimension $k \leq (|\mathcal{A}|\,|Z|)^n$.*

Slide 20

PSRs vs. other dynamic models (cont.)

- In a POMDP, all parameter vectors m_T are *componentwise positive*;
- In a general system, this need not happen;

Therefore,

Theorem 4. *There are dynamical systems with finite linear dimension that cannot be modelled by any finite POMDP.*

Corollary 5. *There are dynamical systems with finite linear dimension that cannot be modelled by any HMM.*

Corollary 6. *There are dynamical systems with finite linear dimension that cannot be modelled by any n-order Markov model.*

Slide 21

Outline of the presentation

- Background on POMDPs
- PSRs: What is this all about?
- PSRs vs. other dynamic models
- **Discovery, learning and planning in PSRs**

Slide 22

Discovery, learning and planning in PSRs

Given a dynamic system with system dynamics matrix $\mathcal{D}$, 3 problems immediately arise when dealing with PSRs:

1. How to choose the core tests q_i (discovery)?
2. How to suitably determine the parameters $m_{(a,z)}$ and $m_{(a,z,q_i)}$ (learning)?
3. How to plan using the PSR model (planning)?

I now provide some references on the three above problems and sketch the main ideas.

Slide 23

Discovery in PSRs

- *Discovery* deals with the problem of building *good* core tests;
- The predictions of these tests will constitute the *state* of the PSR;
- Few algorithms address the problem of discovery;
- In [4], the Analytic Discovery and Learning (ADL) algorithm is proposed to determine the core tests if $\mathsf{P}(T \mid H)$ can be determined for all T and H;
- This algorithm iteratively builds submatrices of $\mathcal{D}$ until two such consecutive submatrices yield the same rank.

Slide 24

Discovery in PSRs (cont.)

- To alleviate the assumption on the computability of $\mathsf{P}(T \mid H)$, the authors consider dynamical systems *with reset*;
- This reset is used to generate *iid* samples of $\mathsf{P}(T \mid H$ and estimate this value.
- A related approach is followed in [9].

Slide 25

Learning in PSRs

- *Learning* deals with the problem of estimating the PSR parameters;
- The paper in [4] also uses the data sampled from the system to estimate the PSR parameters, by inverting the state-update equation;
- In [14], the need for resets is alleviated by considering *suffixes* in the observed histories;
- The papers [9, 13] use approximated gradient ascent techniques to estimate the PSR parameters;
- In [11], a modified PSR model is considered and SVD analysis is used to learn the modified model parameters.

Slide 26

Planning in PSRs

- *Planning* deals with the problem of optimal policy determination in PSRs;
- To address planning in PSRs, a reward structure must be appended to the PSR model;
- This is done by considering a reward to be issued *together with each observation*;
- Tests are now sequences $T = (a_1, (z_1, r_1), a_2, (z_2, r_2), \dots, a_k, (z_k, r_k))$, with r_i taking values in some finite set $\mathcal{R} \subset \mathbb{R}$;
- Similarly, histories are now sequences $H = (a_1, (z_1, r_1), a_2, (z_2, r_2), \dots, a_k, (z_k, r_k))$;
- All other concepts carry on without modification.

Slide 27

Planning in PSRs (cont.)

- With this formulation, it is possible to define an expected immediate reward $R(H, a)$ as

$$R(H,a) = \sum_{r \in \mathcal{R}} r \mathbb{P}\left[r \mid H, a\right] =$$

$$= \mathsf{P}(Q \mid H) \sum_{r \in \mathcal{R}} r \sum_{z \in \mathcal{Z}} m_{(a,(r,z))} = \quad = \mathsf{P}(Q \mid H) \hat{m}_a;$$

- The immediate expected reward is a linear function of the state vector $\mathsf{P}(Q \mid H)$;
- With this formalism, a value function can be defined *with similar properties to the optimal value function in POMDPs*;
- POMDP solution methods (exact and approximate) can then be applied straightforwardly to PSRs [5].

Slide 28

*

References

[1] D. A. Aberdeen. A (revised) survey of approximate methods for solving partially observable Markov decision processes. Technical report, National ICT Australia, Canberra, Australia, 2003.

[2] A. R. Cassandra. *Exact and approximate algorithms for partially observable Markov decision processes.* PhD thesis, Brown University, May 1998.

[3] A. R. Cassandra, M. L. Littman, and N. L. Zhang. Incremental pruning: A simple, fast, exact method for partially observable Markov decision processes. In *Proceedings of the 13th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 54–61, Providence, Rhode Island, 1997. Morgan Kaufmann Publishers.

[4] M. R. James and S. P. Singh. Learning and discovery of predictive state representations in dynamical systems with reset. In *Proceedings of the 21st International Conference on Machine Learning (ICML'04)*, pages 53–60, New York, NY, 2004. ACM Press.

Slide 29

[5] M. R. James, S. P. Singh, and M. L. Littman. Planning with predictive state representations. In *Proceedings of the 2004 International Conference on Machine Learning and Applications*, pages 304–311, 2004.

[6] M. L. Littman. The Witness algorithm: Solving partially observable Markov decision processes. Technical Report CS-94-40, Department of Computer Sciences, Brown University, December 1994.

[7] M. L. Littman, A. R. Cassandra, and L. P. Kaelbling. Learning policies for partially observable environments: Scaling up. In A. Prieditis and S. Russell, editors, *Proceedings of the 12th International Conference on Machine Learning (ICML'95)*, pages 362–370, San Francisco, CA, 1995. Morgan Kaufmann Publishers.

[8] M. L. Littman, R. S. Sutton, and S. Singh. Predictive representations of state. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14, pages 1555–1561, Cambridge, MA, USA, 2002. MIT Press.

[9] P. McCracken and M. Bowling. Online discovery and learning of predictive state representations. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in

Slide 30

Neural Information Processing Systems, volume 18, pages 875–882, Cambridge, MA, 2006. The MIT Press.

[10] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of Markov chain decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

[11] M. Rosencrantz, G. Gordon, and S. Thrun. Learning low dimensional predictive representations. In *Proceedings of the 21st International Conference on Machine Learning (ICML'04)*, pages 88–95, New York, NY, 2004. ACM Press.

[12] M. R. Rudary and S. P. Singh. A nonlinear predictive state representation. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 16. The MIT Press, 2003.

[13] S. P. Singh, M. R. James, and M. R. Rudary. Predictive state representations: A new theory for modeling dynamical systems. In *Proceedings of the 20th Annual Conference on Uncertainty in Artificial Intelligence (UAI'04)*, pages 512–519, Arlington, Virginia, 2004. AUAI Press.

[14] B. Wolfe, M. R. James, and S. Singh. Learning predictive state representations in dynamical systems without reset. In *Proceedings of the 22nd International

Slide 31

Conference on Machine Learning (ICML'05), volume 119 of *ACM International Conference Proceeding Series*, pages 985–992. ACM Press, 2005.