A Planning Algorithm for Predictive State Representation

Masoumeh T. Izadi

Doina Precup

School of Computer Science, McGill University

www.cs.mcgill.ca/~mtabae

www.cs.mcgill.ca/~dpreciup
Motivation

Prediction and control is partially observable environments is usually approached using two kinds of methods:

- Partially Observable Markov Decision Processes (POMDPs)
- History-based representations

POMDPs are expressive but the cost of finding an optimal way of behaving is prohibitive.
Predictive State Representation (PSR) is an alternative to POMDP introduced very recently by Littman et al. (2002). They are:

- general and expressive, like POMDPs
- potentially more compact than POMDPs
- grounded in the experience of the agent, like history-based methods

No algorithm for optimal control using PSRs have been proposed yet!

We present a simple policy iteration algorithm using PSRs.
Predictive State Representations (PSRs)

- PSRs are based on the notion of tests.
- A test is an ordered sequence of action-observation pairs
  \[ q = a_1 o_1 \ldots a_k o_k \]
- The prediction for test \( q \) given history \( h \), \( p(q|h) \), is the probability of seeing sequence \( o_1 \ldots o_k \) after seeing history \( h \) and taking actions \( a_1 \ldots a_k \)
- A set of tests \( Q \) is a PSR if its prediction vectors, \( [p(q_1|h), \ldots p(q_{|Q|}|h)] \), form a sufficient statistic for the dynamical system.
Linear PSRs

- For any test $q$, there exists a projection vector $m_q$ s.t. $P(q|h) = p(Q|h)^T m_q$.

- An outcome vector $u_q$ indicates the probability of the test $q$ when applied from each underlying state $s_i$.

- The outcome vectors of the core tests in $Q$ are concatenated to form the $(n \times |Q|)$ state-test prediction matrix $U$. 
Policy Evaluation

- Assume a policy $\pi : H \rightarrow A$ mapping histories to actions and an initial state $I$.
- The value of $\pi$ is given by:

$$V(\pi) = \sum_{q \in \Psi_h} P(q|I, \pi) V(q|I, \pi)$$

where $V(q|I, \pi)$ is the total expected return for test $q$ and $\Psi_h$ is the set of all tests of length $t$ under policy $\pi$.

- For each action-observation pair, $ao$, we can define a projection matrix: $M^{ao} = (U^+ T^a O^{ao} U)^T$ and a projection vector $m^{ao} = (U^+ T^a O^{ao} e_n^T)^T$. 
Considering the probability distribution over tests that $\pi$ generates and the expected return of a test given $I, \pi$, we have:

$$P(q|I, \pi)V(q|I, \pi) = \sum_{i=1}^{h} \frac{IU \prod_{j=1}^{i} M_{a_j o_j}^{T} U^{+}}{IU m_{a_1 o_1 ... a_i o_i}^{T}} R^{a_i} P(a_i | \pi, a_1 o_1 ... a_{i-1} o_{i-1})$$

This algorithm requires a large amount of pre-computation.

It can work well and good results can be achieved with a small horizon.
Policy Improvement

- Consider a policy tree, alternating actions and observations;
- A deterministic policy defines a path through the tree;
- Start with a path, compute values of alternatives that are one-step different;
- Greedily move to a better policy, if one is found in this way.
A Small Example

- 4 deterministic actions: N, S, E, W;
- Goal state g generates a distinct observation and reward +1;
- All other states generate the same observation, n, and no reward;
Results

![Graph showing policy value over horizon]

- PSR
- Witness
Conclusion and Future Work

- Replacing belief states by prediction vectors opens up new possibilities for planning algorithms.
- Our exact algorithm is still exponential in the horizon length.
- Approximate planning algorithms using PSR modeling.
- Learning PSRs from data (see also Singh et al., ICML03).