Predictive Timing Models

Pierre-Luc Bacon, Borja Balle, Doina Precup
Reasoning and Learning Lab, McGill University

Abstract

We consider the problems of learning and planning in Markov decision processes with temporally extended actions represented in the options framework.

- We propose to use predictions about the duration of extended actions to represent the state.
- We develop a consistent and efficient spectral learning algorithm.

Introduction

Learning good models can be challenging (think of the Atari domain for example). We consider a simpler kind of model: a subjective (agent-oriented) predictive timing model. We define a notion of predictive state over the durations of possible courses of actions. Timing of events is understood to be crucial to animal learning (eg. Machado et al, 2009)

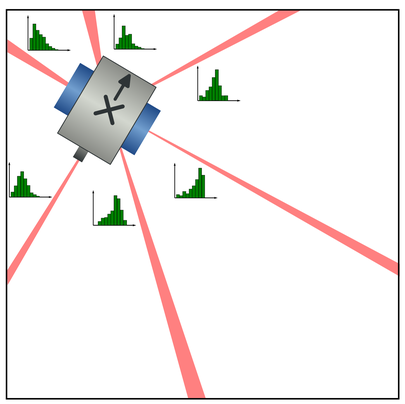

Figure 1: Imagine that a robot has models for options that move radially out from the current position, this would allow localizing with respect to all neighbouring walls.

Other examples:

- Minimum time to goal problems
- Financial agent holding stocks: hold a particular stock while it is above a certain value, and sell under that value.

Options framework

An option is a triple:

$$\langle \mathcal{I} \subseteq \mathcal{S},\ \pi : \mathcal{S} \times \mathcal{A} \to [0,1],\ \beta : \mathcal{S} \to [0,1] \rangle$$

consisting of an **initiation set**, a **policy** π (stochastic or deterministic) and a **termination condition** β.

E.g., robot navigation: if there is no obstacle in front ($\mathcal{I}$), go forward π until you get too close to another object β.

Options Duration Model (ODM)

Instead of predicting a full model at the end of an option (probability distribution over observations), **predict when the option will terminate**.
We have a dynamical system with observations from $\Omega \times \{\sharp, \bot\}$, where:

- $\sharp$ (*sharp*) denotes continuation
- $\bot$ (*bottom*) denotes termination

A trajectory is of the form:

$$(\omega_1, \sharp, \ldots, \omega_1, \sharp, \omega_1, \bot, \omega_2, \sharp, \ldots, \omega_2, \sharp, \omega_2, \bot, \ldots) = (\omega_1, \sharp)^{d_1 - 1}(\omega_1, \bot)(\omega_2, \sharp)^{d_2 - 1}(\omega_2, \bot)\ldots$$

Predictive State Representation

A predictive state representation is a model of a dynamical system where the current state is represented as a set of predictions about the future behavior of the system. A PSR with observations in Σ (finite) is a tuple $\mathcal{A} = \langle \boldsymbol{\alpha}_\lambda, \boldsymbol{\alpha}_\infty, \{\mathbf{A}_\sigma\}_{\sigma \in \Sigma} \rangle$ where $\boldsymbol{\alpha}_\lambda, \boldsymbol{\alpha}_\infty \in \mathbb{R}^n$ are the initial and final weights $\mathbf{A}_\sigma \in \mathbb{R}^{n \times n}$ are the transition weights.

Embedding

Let $\delta(\alpha, \omega)$ be a random variable representing the duration of option ω when started from $s \sim \alpha$. The probability of a sequence of options $\bar{\omega} = \omega_1 \cdots \omega_t$ and their durations $\bar{d} = d_1 \cdots d_t$, $d_i > 0$ is given by:

$$\mathbb{P}[\bar{d} | \alpha, \bar{\omega}] = \boldsymbol{\alpha}^\top \mathbf{A}_{\omega_1,\sharp}^{d_1 - 1} \mathbf{A}_{\omega_1,\bot} \mathbf{A}_{\omega_2,\sharp}^{d_2 - 1} \mathbf{A}_{\omega_2,\bot} \cdots \mathbf{A}_{\omega_t,\sharp}^{d_t - 1} \mathbf{A}_{\omega_t,\bot} \mathbf{1} \ .$$

where:
$\mathbf{e}_{s_0} \in \mathbb{R}^S, \mathbf{e}_{s_0}(s) = \mathbb{I}[s = s_0]$
$\mathbf{A}_{\omega,\sharp}(s, s') = \Sigma_{a \in A} \pi(s, a) P(s, a, s')(1 - \beta(s'))$
$\mathbf{A}_{\omega,\bot}(s, s') = \Sigma_{a \in A} \pi(s, a) P(s, a, s') \beta(s')$,
$\mathbf{1} \in \mathbb{R}^S$

Theorem

Let M be an MDP with n states, Ω a set of options, and $\Sigma = \Omega \times \{\sharp, \bot\}$. For every distribution α over the states of M, there exists a PSR $\mathcal{A} = \langle \boldsymbol{\alpha}, \mathbf{1}, \{\mathbf{A}_\sigma\} \rangle$ with at most n states that computes the distributions over durations of options executed from a state sampled according to α.

Learning

A Hankel matrix a bi-infinite matrix, $\mathbf{H} \in \mathbb{R}^{\Sigma^\star \times \Sigma^\star}$ with rows and columns indexed by strings in $\Sigma^\star$, which contains the joint probabilities of prefixes and suffixes. We can recover (up to a change of basis) the underlying PSR through the SVD decomposition $\mathbf{U}\mathbf{\Lambda}\mathbf{V}^\top$ of $\mathbf{H}$:

$$\boldsymbol{\alpha}_\lambda^\top = \mathbf{h}_{\lambda,S}^\top \mathbf{V}$$
$$\boldsymbol{\alpha}_\infty = (\mathbf{H}\mathbf{V})^+ \mathbf{h}_{P,\lambda}$$
$$\mathbf{A}_\sigma = (\mathbf{H}\mathbf{V})^+ \mathbf{H}_\sigma \mathbf{V}$$

Experiments

We first illustrate our approach on an empty square grids of different sizes.

- Option terminates after running into a wall
- Primitive actions keep the agent in the same spot w.p. 0.1

We also tested the approach in a simulated robot with continuous state and nonlinear dynamics using the Box2D physics engine.

Results

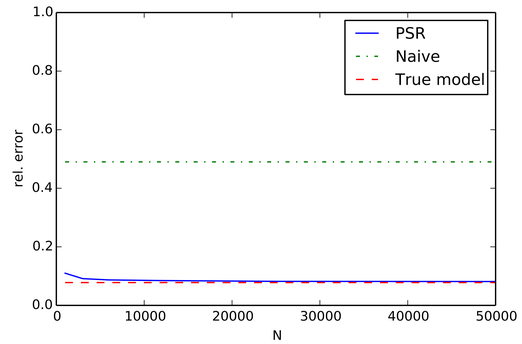

Figure 2: The "naive" method consists in predicting the empirical mean durations, regardless of history. The PSR state updates clearly help.

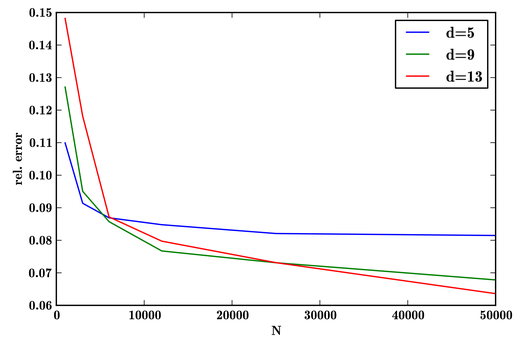

Figure 3: Relative error as a function of the number of samples for different grid sizes.

Conclusion

We presented an approach to learn a predictive model for option durations. Timing models get around the problems of:

- large action spaces (by using a finite set of options)
- large observation spaces (by focusing only on continuation and termination).

We can also show that the value function can be expressed as linear function of the PSR state of an ODM.