Preditive Timing Models

Pierre-Luc Bacon, Borja Balle, Doina Precup

Reasoning and Learning Lab
McGill University

From bad models to good policies (NIPS 2014)

Motivation

- Learning good models can be challenging (think of the Atari domain for example)

Motivation

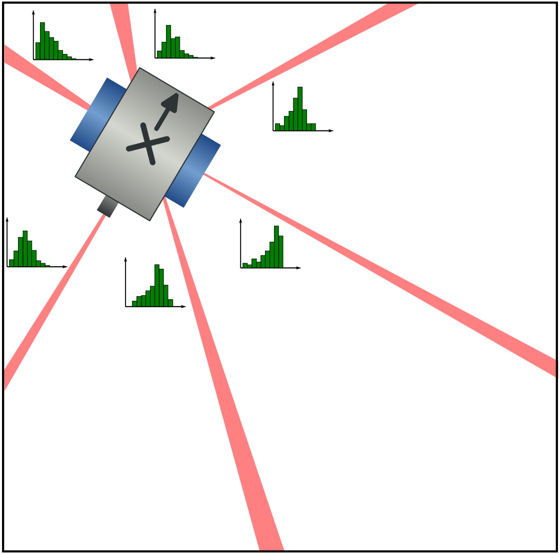
- Learning good models can be challenging (think of the Atari domain for example)
- We consider a simpler kind of model: a **subjective** (agent-oriented) predictive **timing** model.

Motivation

- 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.

Motivation

- 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 models are known to be important in animal learning (eg. Machado et al, 2009)



Hypothetical timing model for a localization task

Today's presentation will mostly be about the learning problem.
Planning results are coming up.

Options framework

An option is a triple:

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

- **initiation set** $\mathcal{I}$
- **policy** π (stochastic or deterministic)
- **termination condition** β

Example

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

Usual option models

1. Expected reward r_ω: for every state, it gives the expected return during ωs execution
2. Transition model p_ω: conditional distribution over next states (reflecting the discount factor γ and the option duration)

Models give predictions about the future, conditioned on the option being executed, i.e. generalized value functions

Options Duration Model (ODM)

Instead of predicting a full model at the end of an option (probability distribution over observations or states), **predict when the option will terminate**, i.e. the expected option duration or the distribution over durations

Model

We have a dynamical system with observations from $\Omega \times \{\sharp, \perp\}$, where:

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

We obtain a coarser representation of the original MDP:

$$
\begin{aligned}
(s_1, \pi_{\omega_1}(s_1)), \dots, (s_{d-1}, \pi_{\omega_1}(s_{d_1-1})), (s_{d1}, \pi_{\omega_2}(s_{d_1})), \dots \rightarrow \\
(\omega_1, \sharp, \dots, \omega_1, \sharp, \omega_1, \perp, \omega_2, \sharp, \dots, \omega_2, \sharp, \omega_2, \perp, \dots) \\
= (\omega_1, \sharp)^{d_1 - 1}(\omega_1, \perp)(\omega_2, \sharp)^{d_2 - 1}(\omega_2, \perp) \dots
\end{aligned}
$$

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 $\mathcal{A} = \langle \boldsymbol{\alpha}_\lambda, \boldsymbol{\alpha}_\infty, \{\mathbf{A}_\sigma\}_{\sigma \in \boldsymbol{\Sigma}} \rangle$ 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

Predicting with PSR

A PSR $\mathcal{A}$ computes a function $f_{\mathcal{A}} : \Sigma^\star \to \mathbb{R}$ that assigns a number to each string $x = x_1 x_2 \cdots x_t \in \Sigma^\star$ as follows:

$$f_{\mathcal{A}}(x) = \boldsymbol{\alpha}_\lambda^\top \mathbf{A}_{x_1} \mathbf{A}_{x_2} \cdots \mathbf{A}_{x_t} \boldsymbol{\alpha}_\infty = \boldsymbol{\alpha}_\lambda^\top \mathbf{A}_x \boldsymbol{\alpha}_\infty \; .$$

The conditional probability of observing a sequence of observations $v \in \Sigma^\star$ after u is:

$$f_{\mathcal{A},u}(v) = \frac{f_{\mathcal{A}}(uv)}{f_{\mathcal{A}}(u)} = \frac{\boldsymbol{\alpha}_\lambda^\top \mathbf{A}_u \mathbf{A}_v \boldsymbol{\alpha}_\infty}{\boldsymbol{\alpha}_\lambda^\top \mathbf{A}_u \boldsymbol{\alpha}_\infty} = \frac{\boldsymbol{\alpha}_u^\top \mathbf{A}_v \boldsymbol{\alpha}_\infty}{\boldsymbol{\alpha}_u^\top \boldsymbol{\alpha}_\infty} \; .$$

The PSR semantics of u is that of a *history*, and v of a *test*.

Embedding

Let $\delta(s_0, \omega)$ be a random variable representing the duration of option ω when started from s_0

$$\mathbb{P}[\delta(s_0, \omega) = d] = \mathbf{e}_{s_0}^\top \mathbf{A}_{\omega,\sharp}^{d-1} \mathbf{A}_{\omega,\perp} \mathbf{1} \;\; ,$$

$\mathbf{e}_{s_0} \in \mathbb{R}^S$ is an indicator vector with $\mathbf{e}_{s_0}(s) = \mathbb{I}[s = s_0]$

$\mathbf{A}_{\omega,\sharp}(s, s') = \sum_{a \in A} \pi(s, a) P(s, a, s') \underbrace{(1 - \beta(s'))}_{\text{not stopping}}$

$\mathbf{A}_{\omega,\perp}(s, s') = \sum_{a \in A} \pi(s, a) P(s, a, s') \; \underbrace{\beta(s')}_{\text{stopping}} \; ,$

$\mathbf{1} \in \mathbb{R}^S$

Theorem

Let M be an MDP with n states, Ω a set of options, and $\Sigma = \Omega \times \{\sharp, \perp\}$. 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.

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 then given by:

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

Learning

A Hankel matrix a bi-infinite matrix, $\mathbf{H}_f \in \mathbb{R}^{\Sigma^\star \times \Sigma^\star}$ with rows and columns indexed by strings in $\Sigma^\star$, which contains the joint probabilities of prefixes and suffixes.

$$
\begin{array}{c c}
 & \begin{array}{cccc} \epsilon & (\omega_0, \perp) & (\omega_0, \sharp), (\omega_0, \perp) & (\omega_0, \sharp), (\omega_0, \sharp), (\omega_0, \perp), \ldots \end{array} \\
\begin{array}{c} \epsilon \\ (\omega_0, \sharp) \\ (\omega_0, \sharp), (\omega_0, \sharp) \\ (\omega_0, \sharp), (\omega_0, \sharp), (\omega_0, \perp) \\ \vdots \end{array} & \begin{bmatrix} & \vdots & \\ & & \\ \ldots & \mathbb{P}[(\omega_0, \sharp)(\omega_0, \sharp)(\omega_0, \sharp)(\omega_0, \perp)] & \ldots \\ & & \\ & \vdots & \end{bmatrix}
\end{array}
$$

Node: closely related to the so-called *system dynamics matrix*

Data $\longrightarrow$ *Low-rank matrix estimation* $\longrightarrow$ **Hankel matrix** $\longrightarrow$ *Factorization and linear algebra* $\longrightarrow$ **WFA**

Key Idea: The Hankel Trick

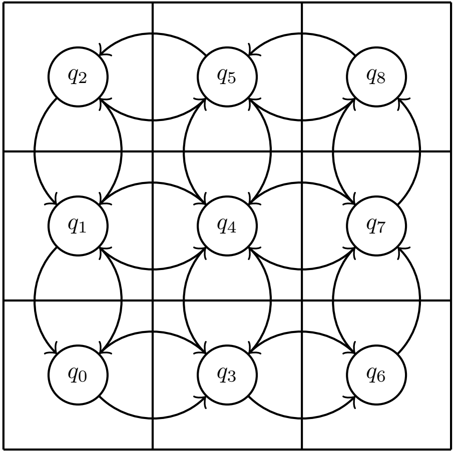
We can recover (up to a change of basis) the underlying PSR through a rank-factorization of the Hankel matrix.

Given the SVD $\mathbf{U}\mathbf{\Lambda}\mathbf{V}^\top$ of $\mathbf{H}$, 3 lines of code suffice:

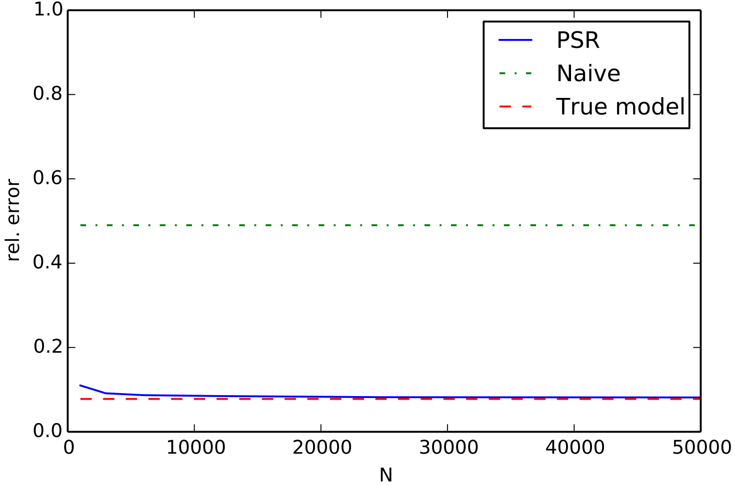
$$\begin{aligned}
\boldsymbol{\alpha}_\lambda^\top &= \mathbf{h}_{\lambda,S}^\top \mathbf{V} \\
\boldsymbol{\alpha}_\infty &= (\mathbf{H}\mathbf{V})^+ \mathbf{h}_{P,\lambda} \\
\mathbf{A}_\sigma &= (\mathbf{H}\mathbf{V})^+ \mathbf{H}_\sigma \mathbf{V}
\end{aligned}$$

Note: The use of SVD makes the algorithm robust to noisy estimation of $\mathbf{H}$.

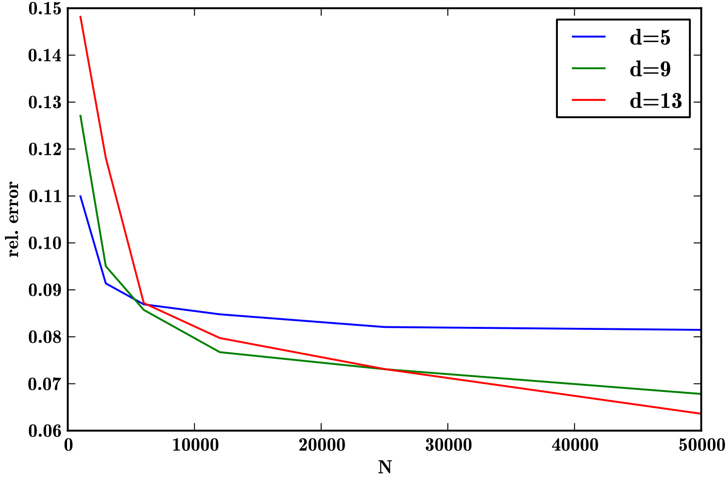
Synthetic experiment



Four options: go N, E, W, or S until the agent hits a wall. A primitive action succeeds with probability 0.9. We report the relative errors: $\frac{|\mu_{\mathbf{A}} - d_\omega|}{\max\{\mu_{\mathbf{A}}, d_\omega\}}$



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



Relative error as a function of the number of samples for different grid sizes

Continuous domain

$\lvert\Omega\rvert$	(K_r, K_s)	$h=1$	$h=2$	$h=3$	$h=4$	$h=5$	$h=6$	$h=7$	$h=8$
4	(2 , 1)	0.19 (199)	0.25 (199)	0.26 (196)	0.30 (198)	0.31 (172)	0.33 (163)	0.31 (173)	0.30 (172)
	(1 , 1)	0.15 (133)	0.28 (126)	0.31 (134)	0.35 (131)	0.36 (131)	0.36 (131)	0.36 (132)	0.36 (133)
8	(2 , 1)	0.40 (176)	0.47 (163)	0.49 (163)	0.51 (176)	0.52 (162)	0.51 (164)	0.50 (163)	0.52 (167)
	(1 , 1)	0.38 (166)	0.48 (162)	0.46 (195)	0.51 (164)	0.52 (162)	0.51 (162)	0.51 (165)	0.54 (169)

Simulated robot with continuous state and nonlinear dynamics.
We use the Box2D physics engine to simulate a circular differential wheeled robot (Roomba-like)

Future work

Planning: We have been able to show that given a policy over options: and some ODM state then the value function is a linear function the PSR state.

This suggests that the ODM state might be sufficient for planning

Also on the agenda:

- Try to gain a better theoretical understanding of the environment vs PSR-rank relationship.
- Conduct planning experiments on the learnt models.

Thank you

The off-policy case

The exploration policy will be reflected in the empirical Hankel matrix. We can compensate by forming an auxiliary PSR. For a uniform policy, we would have:

$$\begin{aligned}
\boldsymbol{\alpha}_{\lambda}^{\pi} &= \mathbf{e}_0 \\
\boldsymbol{\alpha}_{\infty}^{\pi} &= \mathbf{1} \\
\mathbf{A}_{\omega_i,\sharp}^{\pi}(0, \omega_i) &= |\Omega| \\
\mathbf{A}_{\omega_i,\sharp}^{\pi}(\omega_i, \omega_i) &= 1 \\
\mathbf{A}_{\omega_i,\sharp}^{\pi}(0, 0) &= |\Omega| \\
\mathbf{A}_{\omega_i,\sharp}^{\pi}(\omega_i, 0) &= 1
\end{aligned}$$

and take compute the corrected Hankel by taking the Hadamard product:

$$\mathbf{H} = \hat{\mathbf{H}} \odot \mathbf{H}_{\pi}$$