Learning Multi-Step Predictive State Representations

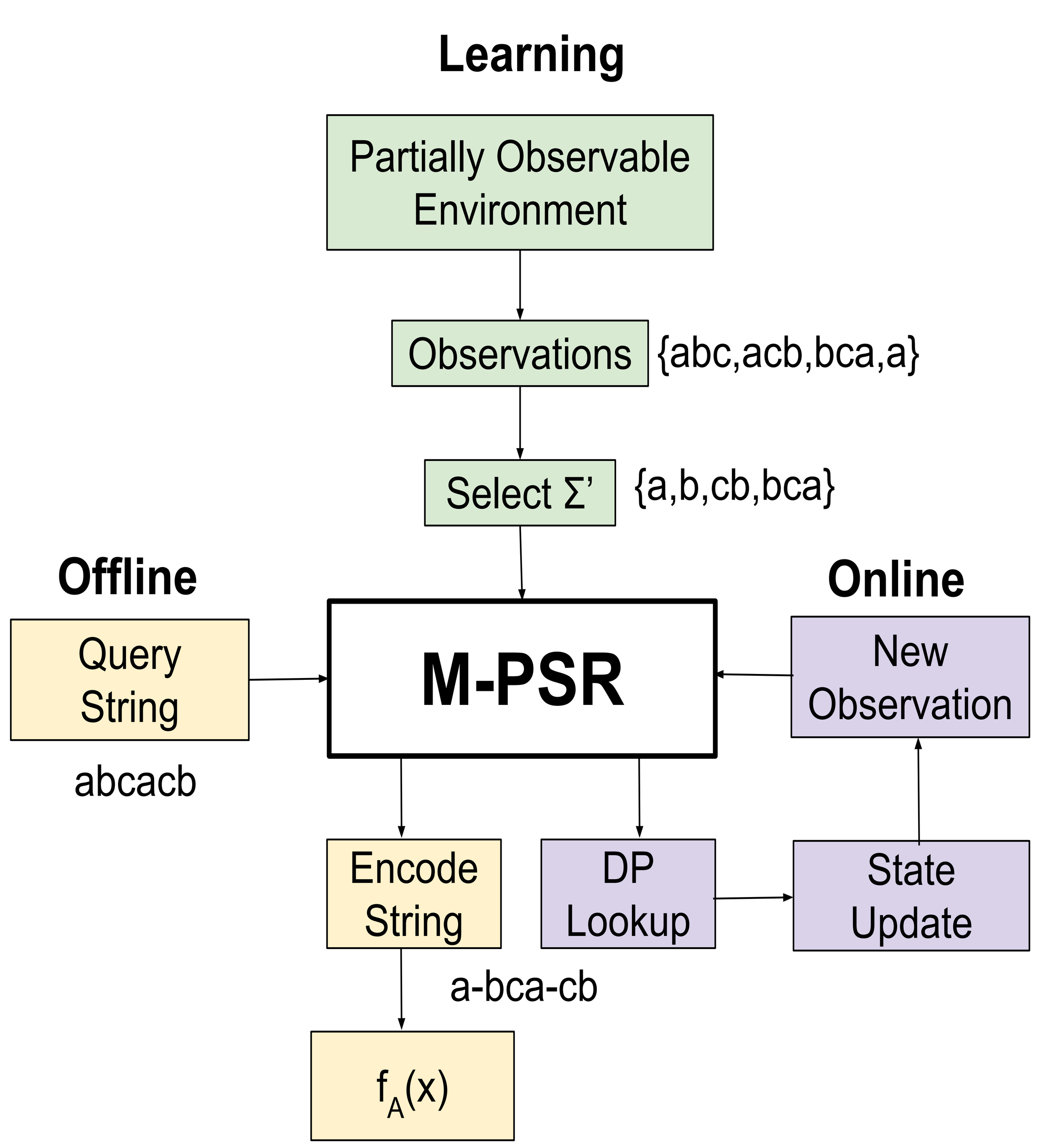
Lucas Langer, Borja Balle, Doina Precup
Reasoning and Learning Lab, McGill University

McGill

Abstract

With spectral learning algorithms, we consider the problem of making predictions in partially observable environments. Spectral algorithms learn a linear representation of a system's internal states and can be used to make predictions of future observations conditioned on the past (Boots et al., 2011). Such predictions have been applied to robotics, where expectations of the future guide planning decisions (Bacon et al., 2015). Spectral algorithms have the advantage that they are statistically consistent with well known error bounds (Citation). In this work, we develop a novel extension to a previous spectral based representation. We augment Predictive State Representations (PSRs) with multi-step transition operators to get Multi-Step Predictive State Representations (M-PSRs). In experiments, the novel representation far outperforms its predecessor when using regularized models.

Overview



Multi-Step Predictive State Representations (M-PSRs)

As with traditional PSRs, M-PSRs compute a probability function on observation strings $f_A : \Sigma^* \mapsto [0,1]$. A traditional PSR is defined by the tuple: $\langle \Sigma, \boldsymbol{\alpha}_\lambda, \boldsymbol{\alpha}_\infty, \{\mathbf{A}_\sigma\}_{\sigma\in\Sigma}\rangle$, where Σ is the set of observations, α_λ is an initial weighting on states, α_∞ is a normalizing vector, and $\{\mathbf{A}_\sigma\}_{\sigma\in\Sigma}$ is a set of transition operators. An M-PSR expresses transitions more compactly by including operators which represent sequences of observations: $\Sigma' \subset \Sigma^*$ and a way of partitioning strings which we call an encoding function κ.

Probability Queries

With traditional PSRs, transition operators are applied for each observation σ_i in $x = \sigma_1 \cdots \sigma_n$.

$$f_{\mathcal{A}}(x) \;=\; f_{\mathcal{A}}(\sigma_1 \cdots \sigma_n) \;=\; \boldsymbol{\alpha}_\lambda^\top \mathbf{A}_{\sigma_1} \cdots \mathbf{A}_{\sigma_n} \boldsymbol{\alpha}_\infty$$

For M-PSRs, we apply multi-step transition operators for each observation sequence s_i in $\kappa(x) = s_1 \cdots s_m$

$$f_{\mathcal{A}}(x) \;=\; f_{\mathcal{A}}(s_1 \cdots s_m) \;=\; \boldsymbol{\alpha}_\lambda^\top \mathbf{A}_{s_1} \cdots \mathbf{A}_{s_m} \boldsymbol{\alpha}_\infty$$

Base M-PSRs

As an example, a natural approach to build an M-PSR for timing $\Sigma = \{\sigma\}$, is to use powers of a fixed base b:

$$\Sigma' \;=\; \{\sigma,\; \sigma^b,\; \sigma^{b^2}, \ldots, \sigma^{b^K}\}$$

For given observation string: σ^t, we compute $t = t_0 b^0 + t_1 b^1 + t_2 b^2 + \cdots + t_K b^K$ and then the encoding follows:

$$\kappa(\sigma^t) \;=\; (\sigma^{b^K})^{t_K}\, (\sigma^{b^{K-1}})^{t_{K-1}} \cdots (\sigma^b)^{t_1}\, (\sigma)^{t_0}$$

Data-Driven M-PSRs

For a general M-PSR we propose that $\kappa(x)$ be the encoding of minimal length. $\kappa(x)$ is computed by dynamic programming, giving the additional benefit that one can perform optimal state updates for online applications.

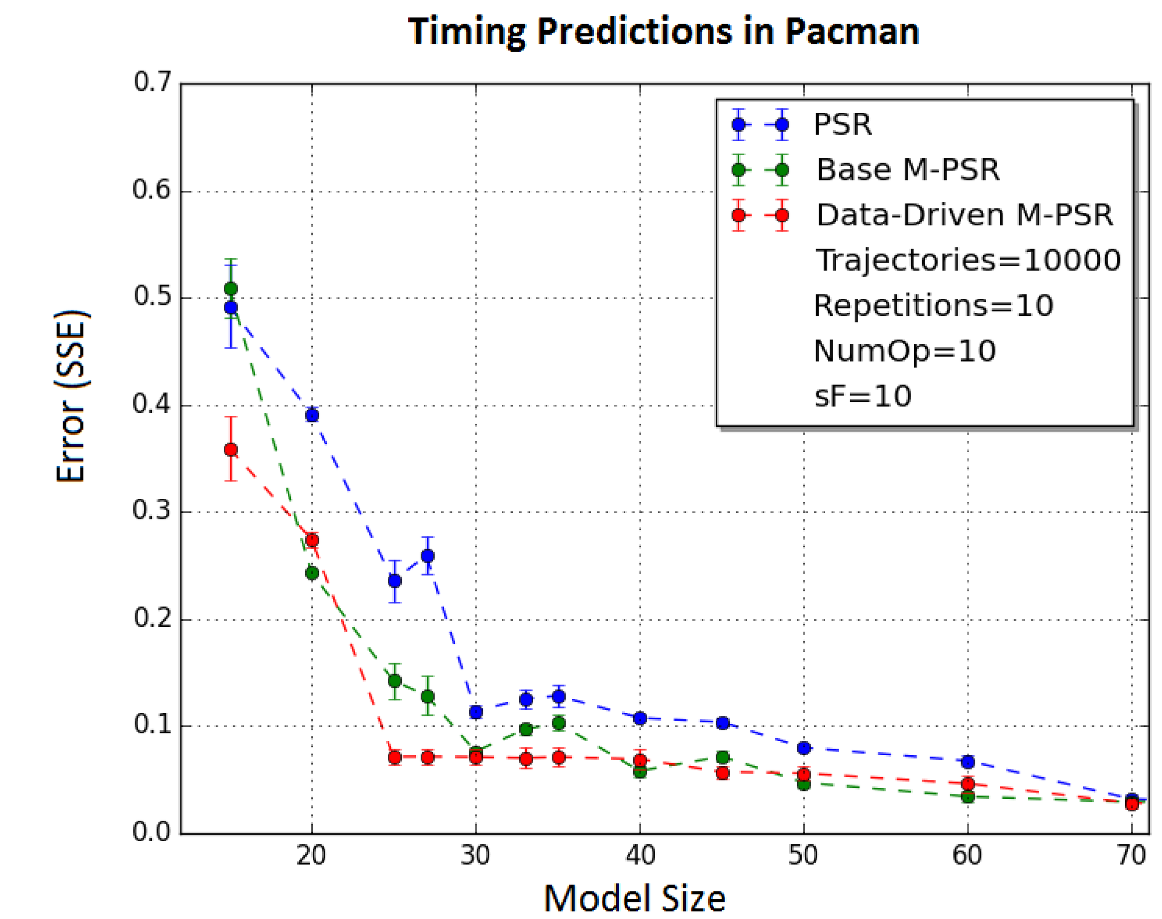
$$\kappa(x) \;=\; \operatorname*{argmin}_{z\in\Sigma',\; x=yz,\; |y|<|x|} |\kappa(y)(z)|$$

With this choice of κ, we build Σ' by iteratively selecting sub-strings from the training set:

$$\Sigma'_{i+1} \;=\; \operatorname*{argmin}_{u\in \mathrm{sub}_M} \sum_{x\in\mathrm{train}} |\kappa_{\Sigma'_i \cup \{u\}}(x)|$$

Experiments

We simulated robot trajectories in labyrinth environments. Observations are time and wall colors.



figs/DLMO/MO_1k.png

Conclusion

M-PSRs offer large improvements over traditional PSRs when using smaller model sizes, as is often desired in practice. These improvements offer various prospects for future work:

- Applications: planning, physiological signal recordings, and financial time-series
- Further optimization of the encoding function κ and selection of transition sequences Σ'
- Theoretical analysis of M-PSRs