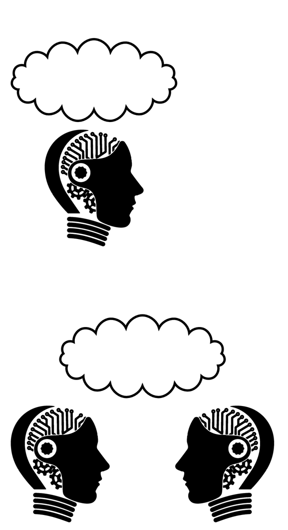
Distributed Gaussian Process Temporal Differences for Actor-critic Learning

John Martin, Zheng Xing, Zhiyuan Yao, Ionut Florescu, Brendan Englot
Stevens Institute of Technology, Hoboken NJ

Summary

We consider Reinforcement Learning with Gaussian Process (GP) temporal differences [2]. Our work studies the extent to which distributed computing can improve the amount of data GP-based value models can handle. By invoking episodic independence, we derive two different distributive models. One model represents the predictive value posterior as a sum of K experts, and the other, as a product. As such, predictions can be distributed to K independent processors. We propose actor-critic methods that exploit these models for efficient policy evaluation and action selection – balancing exploration and exploitation by maximizing the GP-UCB criterion [3]. Our experiments compare the resulting methods to an actor-critic based on the standard GP Temporal Difference value model. We show our methods are able to process more data, and therefore, can solve complex problems which are too data-intensive for the standard model.

Application: Cloud Robotics



Distributed methods can reduce the complexity of robot learning. *Individual robots* can scale their learning effort on-demand by delegating intensive computations to expandable off-board resources. *Collaborative robot groups* stand to benefit from a principled information sharing framework.

References

[1] M. Deisenroth and J. Ng. Distributed gaussian processes. In *ICML 32*, 2015.

[2] Y. Engel, S. Mannor, and R. Meir. Reinforcement learning with gaussian processes. In *ICML 22*, 2005.

[3] N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *ICML 27*, 2010.

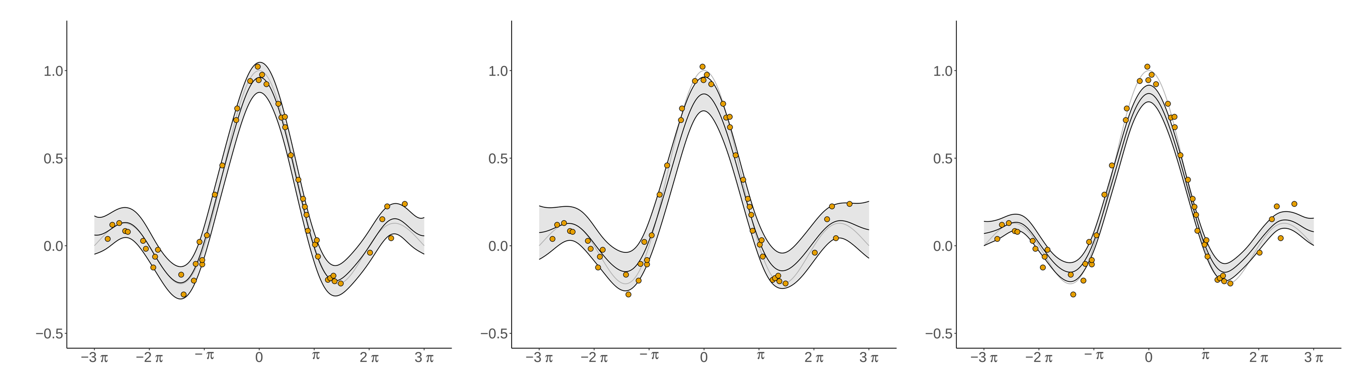
Distributed Actor-critic Procedures

Actor: Our methods consider a distribution over value functions and select actions to maximize the GP Upper Confidence Bound (GP-UCB) [3]

$$\alpha(\mathbf{x}) = v(\mathbf{x}) + \kappa\sqrt{p(\mathbf{x})}.$$

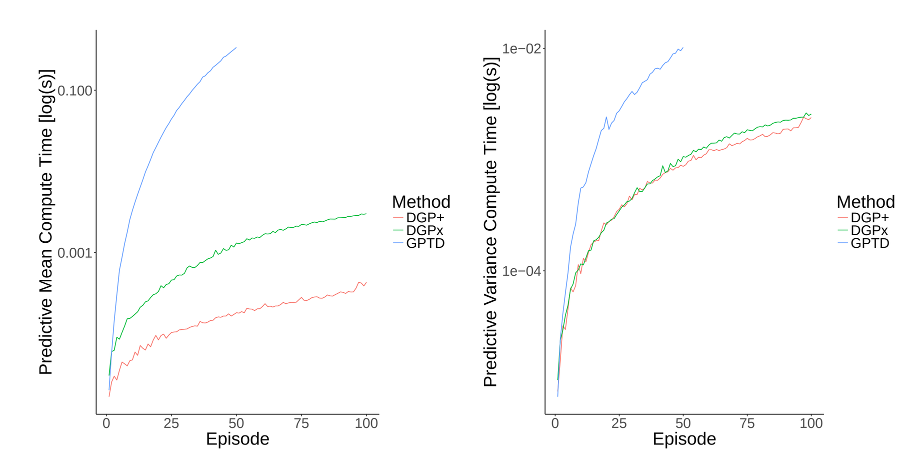
The resulting policy exploits the available data conservatively - acting greedy when uncertainty is low and exploring when it is high.

Critic: The predictive posterior, $p(V|\mathbf{x}_*, \mathcal{D})$, measures longterm utility of a state-action pair, $\mathbf{x}$, along with variance and the associated gradients for action selection. The models we consider allow these predictions to distribute to independent processors.



(a) GP (b) DGP+ (c) DGPx

Complexity: Predictions must invert $\mathbf{K}_{rr} + \mathbf{\Sigma}$. GP-SARSA predictions scale with $\mathcal{O}(N^3)$, where N is the total number of observed transitions. Our distributed methods split this cost among K experts, with n observations each, to achieve $\mathcal{O}(Kn^3)$. Provided $K < N^3/n^3$, we can improve efficiency.



(a) $v(\mathbf{x})$ Time (b) $p(\mathbf{x})$ Time

Background

Reinforcement Learning: Agents select actions according to the discounted return $D(\mathbf{x}) = \sum_{n=0}^{N} \gamma^n R(\mathbf{x}_n)$. Greedy approaches typically strive to maximize its expected value $V(\mathbf{x}_n) = \mathbf{E}[D(\mathbf{x}_n)|R(\mathbf{x}_n), \mathbf{x}_n]$. However, $V(\mathbf{x})$ is inherently latent and must be estimated from sequential observations: states $\mathbf{s}$, actions $\mathbf{a}$, and rewards $R(\mathbf{x})$, where $\mathbf{x} = (\mathbf{s}, \mathbf{a})^\top$.

GP-SARSA: Treat $D(\mathbf{x}) = V(\mathbf{x}) + \xi$ as a random function drawn from a Gaussian Process prior. Apply GP-regression to predict latent values $V(\mathbf{x})$ from observed rewards. The model is $R(\mathbf{x}) = V(\mathbf{x}_n) - \gamma V(\mathbf{x}_{n+1}) + \varepsilon_n$, where $\varepsilon_n = \xi_n - \gamma\xi_{n+1}$, $\xi \sim \mathcal{N}(0, \sigma^2)$. Stack values and rewards into vectors, $\mathbf{r}$, $\mathbf{v}$, assuming $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{vv})$. All variables are jointly Gaussian with $\mathbf{r} = \mathbf{H}\mathbf{v} + \varepsilon$,

$$\begin{pmatrix}\mathbf{v}\\ \mathbf{r}\end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix}\mathbf{0}\\ \mathbf{0}\end{pmatrix}, \begin{pmatrix}\mathbf{K}_{vv} & \mathbf{K}_{vv}\mathbf{H}^\top\\ \mathbf{H}\mathbf{K}_{vv} & \mathbf{H}(\mathbf{K}_{vv} + \sigma^2\mathbf{I})\mathbf{H}^\top\end{pmatrix}\right).$$

The upper diagonal matrix $\mathbf{H}$ encodes correlation with elelemts $1, -\gamma$. Condition predictions of V on $\mathbf{r}$ to obain the posterior moments $\mathcal{N}(V(\mathbf{x})|v(\mathbf{x}), p(\mathbf{x}))$

$$v(\mathbf{x}) = \mathbf{k}_{r*}^\top(\mathbf{K}_{rr} + \mathbf{\Sigma})^{-1}\mathbf{r},$$
$$p(\mathbf{x}) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_{r*}^\top(\mathbf{K}_{rr} + \mathbf{\Sigma})^{-1}\mathbf{k}_{r*}.$$

Mixture of Experts (DGP+SARSA)

$$P(V|\mathbf{x}_*, \mathcal{D}) = \sum_{k=1}^{K} P_k(V|\mathbf{x}_*, \mathcal{D}_k).$$

Product of Experts (DGPxSARSA)

$$P(V|\mathbf{x}_*, \mathcal{D}) = \prod_{k=1}^{K} P_k(V|\mathbf{x}_*, \mathcal{D}_k).$$