Data-Efficient Policy Search using PILCO and Directed-Exploration

Rowan McAllister[1] RTM26@CAM.AC.UK
Mark van der Wilk[1] MV310@CAM.AC.UK
Carl Edward Rasmussen CER54@CAM.AC.UK
Department of Engineering, University of Cambridge, Cambridge, CB2 1PZ

1. Introduction

Reinforcement learning (RL) algorithms solve general sequential decision making problems through learning by trial and error. Many reinforcement learning algorithms are proven to find a good or optimal controller, but may take many interactions with the environment to do so. For real world tasks, this is often impractical, as letting a learner interact with the environment takes time and can be costly.

Here we present an extension to PILCO[4], a model-based reinforcement learning algorithm for continuous state and action spaces, which has already shown unprecedented data-efficiency for a variety of tasks, such as the cart-pole swing-up problem. Interestingly, this performance was achieved without any intentional exploration. We introduce a method for balancing exploration and exploitation based on estimating the *reduction* in the variance of the loss, given data that is likely to be observed and show preliminary results.

2. PILCO

PILCO aims to find a locally optimal controller π for an unknown dynamical system f, with continuous states $\mathbf{x}_t$ and inputs $\mathbf{u}_t$ by minimising a given loss function $\mathcal{L}^\pi(\mathbf{x}_{1:T})$. Dynamics learning occurs after each discrete-time trial of length T (providing a new batch of data) and then PILCO chooses the controller to be used for the next trial. The controller and learning algorithm receive noisy observations of the state $\mathbf{y}$. The dynamical system is modelled using a Gaussian process (GP) prior. We use B to denote the combined belief over the dynamics (i.e. the posterior over f) and the corresponding behaviour of the states.

$$\begin{aligned} \mathbf{x}_{t+1} &= f(\mathbf{x}_t, \mathbf{u}_t) + \boldsymbol{\epsilon}_t \\ f &\sim \mathcal{GP}(0, k((\mathbf{x}, \mathbf{u}), (\mathbf{x}, \mathbf{u})')) \\ \mathbf{u}_t &= \pi(\mathbf{y}) \qquad \mathbf{y}_t = \mathbf{x}_t + \boldsymbol{\nu}_t \\ \boldsymbol{\nu}_t &\sim \mathcal{N}\left(0, \sigma_o^2\right) \qquad \boldsymbol{\epsilon}_t \sim \mathcal{N}\left(0, \sigma_p^2\right) \end{aligned}$$

[1]Equal contributions.

After each trial, the controller was chosen by simply minimising the expected loss for the next trial:

$$\pi^* = \operatorname*{argmin}_\pi \mathbb{E}_B \left[\mathcal{L}^\pi(\mathbf{x}_{1:T})\right] \tag{1}$$

PILCO learns the posterior GP for the dynamics using an approximate inference method that deals with the recursive nature of the latent GP output determining its next input [6]. The expected loss is estimated by approximately simulating state trajectories forward through the GP, approximating each transition using moment matching [4].

3. Data efficient RL

PILCO's current impressive data efficiency comes from its model-based approach. Model-based approaches tend to be more data-efficient than model-free approaches due to their ability to generalise from limited experience by exploiting structure inherent to the problem which is encoded in the model[1; 3].

It is perhaps surprising that PILCO achieves such impressive data efficiency, given that it ignores the second main design choice influencing data-efficiency: the exploration-exploitation trade-off. Any exploration that does happen occurs either due to (a) inherent system stochasticity, (b) a tendency for saturating loss functions to give lower expected losses for very uncertain states[5], and (c) the information from executing a controller informs future estimates of the performance for *all* controllers.

Exploration is a critical component of learning quickly because the controller that is chosen for the next trial determines the additional data that is observed. This new data could have a very large impact on the loss of all future trials if it leads to identifying a good controller. This contribution is currently completely ignored since the next controller is chosen to only minimise the expected loss on the next trial.

For data-efficiency, we use directed-exploration [7], guiding exploration towards policies of *reducible* loss-uncertainty, since this has a chance resulting in an better loss. Choosing a controller that turns out to be bad doesn't

hurt much, since it doesn't have to be chosen again. We choose a UCB-like[2] objective function to minimise the loss, plus an exploration term of the expected decrease in variance of the loss, given new observed data $\mathcal{D}$:

$$\pi^* = \underset{\pi}{\operatorname{argmin}}\, \mathbb{E}_B\left[\mathcal{L}^\pi\right] - \beta\sqrt{\mathbb{V}_B\left[\mathcal{L}^\pi\right] - \mathbb{E}_\mathcal{D}\left[\mathbb{V}_{B|\mathcal{D}}\left[\mathcal{L}^\pi\right]\right]} \tag{2}$$

4. Gaussian process approximation

PILCO relies on analytic GP approximations to ensure smoothness of the optimisation of π, such as the moment matching approach for the expected loss. The variance of the loss is already available as a by-product, so the remaining challenging term, is the calculation of the next trial's expected variance.

An intuition of this term can be gained by considering the Monte Carlo estimation. First a new set of observations $\mathcal{D}$ is sampled given the current belief. This is the forward simulation needed for calculating the expected loss. For that sample $\mathcal{D}$, new trajectories are sampled from the updated belief $B|\mathcal{D}$. The variance is then taken over trajectories, while the expectation is taken over hypothetical data.

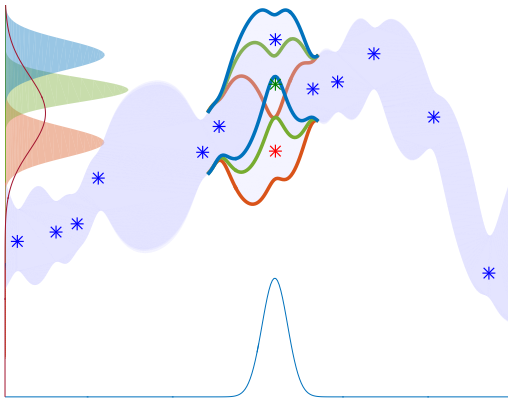

Figure 1. Schematic of expected variance. Input distribution below. Coloured (RGB) are GP posteriors conditioned on fantasy data and corresponding output distributions. Marginal output distribution in a line.

In order to obtain a tractable analytic approximation, we simplify[2] by assuming that (a) observation of $\mathbf{y}_t$ in $\mathcal{D}$ only reduces the variance for $\mathbf{x}_{t+1}$ in the next simulation, (b) costs for consecutive times are independent, and (c) the locations of the observations are at the mean of $\mathbf{x}_t$. This reduces the problem to finding the expected variance for a single output given a random input, with the expectation over a fantasy observation. Figure 1 shows this schematically. Three hypothetical observations are shown, with their corresponding output distributions. Considering a hypothetical observation tells us that the variance will decrease, but since the observation is uncertain, we take the expectation over all hypothetical observations[3].

5. Initial experiments

We compare three different methods each with an adapted exploration term:

- PILCO: No exploration term
- Total variance: $\beta\sqrt{\mathbb{V}_B\left[\mathcal{L}^\pi\right]}$
- Reduced variance: $\beta\sqrt{\mathbb{V}_B\left[\mathcal{L}^\pi\right] - \mathbb{E}_\mathcal{D}\left[\mathbb{V}_{B|\mathcal{D}}\left[\mathcal{L}^\pi\right]\right]}$

β was hand-tuned to 0.1. Preliminary results are shown in figure 2. Figure 2 shows that exploration of some form has

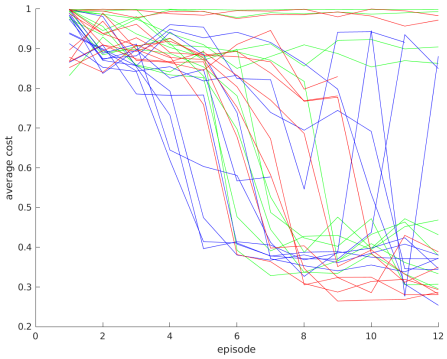

Figure 2. Cart-pole swing-up performance per training episode for PILCO (green), "total variance" (blue), "reduced variance" (red). Different learning trials are shown with different lines.

a positive impact on the data efficiency of PILCO. Interestingly, it seems that the total variance method manages to explore faster earlier, while the reduction in variance method achieves a better loss slightly later.

6. Conclusion

Preliminary results indicate that exploration is an important addition to PILCO when improving on its data efficiency. There are still many open questions left for future work. Firstly, there are many possible improvements for the estimation of the expected variance which are still analytically tractable. Secondly, a more principled method of choosing β is needed. Finally, the algorithms are encouraged to explore, but are evaluated on-policy. A more comprehensive empirical evaluation can show the behaviour more clearly.

[2]Some of the assumptions presented here are not strictly necessary, but are presented for brevity.

[3]Note that the total certainty about any prediction can not change by considering future observations from the current belief. The expected variance plus the variance of the expectation equals the variance without considering any fantasy data.

References

[1] Atkeson, C. G. and Santamaria, J. C. (1997). A comparison of direct and model-based reinforcement learning. In *In International Conference on Robotics and Automation*. Citeseer.

[2] Auer, P. (2003). Using confidence bounds for exploitation-exploration trade-offs. *The Journal of Machine Learning Research*, 3:397–422.

[3] Boone, G. (1997). Efficient reinforcement learning: Model-based acrobot control. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 1, pages 229–234. IEEE.

[4] Deisenroth, M. and Rasmussen, C. (2011). PILCO: A model-based and data-efficient approach to policy search. In *International Conference on Machine Learning (ICML 2011)*, pages 465–472, New York, NY, USA.

[5] Deisenroth, M. P., Fox, D., and Rasmussen, C. E. (2015). Gaussian processes for data-efficient learning in robotics and control. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(2):408–423.

[6] McHutchon, A. (2014). *Nonlinear modelling and control using Gaussian processes*. PhD thesis, University of Cambridge.

[7] Thrun, S. B. (1992). Efficient exploration in reinforcement learning.