Bayesian Actor Critic: A Bayesian Model for Value Function Approximation and Policy Learning

Mohammad Ghavamzadeh MOHAMMAD.GHAVAMZADEH@INRIA.FR
INRIA Lille - Nord Europe, Team SequeL, France

Yaakov Engel YAKIENGEL@GMAIL.COM

1. Introduction

Actor-critic (AC) methods were among the earliest to be investigated in reinforcement learning (RL). AC algorithms are based on the simultaneous online estimation of the parameters of two structures, called the *actor* and the *critic*. The actor corresponds to a conventional action-selection policy, mapping states to actions in a probabilistic manner. The critic corresponds to a conventional value function, mapping states to expected cumulative future reward. Thus, the critic addresses a problem of *prediction*, whereas the actor is concerned with *control*. These problems are separable, but are solved simultaneously to find an optimal policy, as in policy iteration.

In this work,[1] we present a Bayesian take on the AC architecture. The proposed Bayesian actor-critic (BAC) model uses a Bayesian class of non-parametric critics based on the Gaussian process temporal-difference (GPTD) learning (Engel et al., 2005). Such critics model the action-value function as a GP, allowing Bayes' rule to be used in computing a posterior distribution over action-value functions, conditioned on the observed data. The Bayesian actor in BAC is based on the Bayesian policy gradient (BPG) approach proposed in Ghavamzadeh and Engel (2007b). The actor uses the posterior distribution over action-value functions computed by the critic, and derives a posterior distribution for the gradient of the average discounted return with respect to the policy parameters. Appropriate choices of prior covariance (kernel) between state-action values that make action-value function compatible with the parametric family of policies, allow us to obtain closed-form expressions for the posterior distribution of the policy gradient. The posterior mean serves as our estimate of the gradient and is used to update the policy, while the posterior covariance allows us to gauge the reliability of the update.

2. Bayesian Actor-Critic

In AC methods, one defines a class of smoothly parameterized stochastic policies $\{\mu(\cdot|\boldsymbol{x};\boldsymbol{\theta}), \boldsymbol{x}\in\mathcal{X}, \boldsymbol{\theta}\in\Theta\}$. Algorithms typically estimate the gradient of the expected return, $\eta(\boldsymbol{\theta}) = \mathbf{E}\big[\sum_{t=0}^{T}\gamma^t R(\boldsymbol{x}_t,\boldsymbol{a}_t)\big]$, w.r.t. the policy parameters $\boldsymbol{\theta}$ from observed system trajectories, and then improve the policy by adjusting its parameters in the direction of the gradient. The policy gradient theorem (Marbach, 1998, Konda & Tsitsiklis, 2000, Sutton et al., 2000) states that the gradient of the expected return is given by

$$\nabla_{\boldsymbol{\theta}}\eta(\boldsymbol{\theta}) = \int_{\mathcal{Z}} d\boldsymbol{z}\ \pi(\boldsymbol{z};\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}}\log\big(\mu(\boldsymbol{a}|\boldsymbol{x};\boldsymbol{\theta})\big)\ Q(\boldsymbol{z}), \qquad (1)$$

where $\boldsymbol{z}=(\boldsymbol{x},\boldsymbol{a})$, and $\pi(\boldsymbol{z};\boldsymbol{\theta})$ is a discounted weighting of state-action pairs under policy $\mu(\boldsymbol{\theta})$. Moreover, by the *compatibility* assumption (Konda & Tsitsiklis, 2000, Sutton et al., 2000) we may replace the exact (but unknown) action-value function $Q(\boldsymbol{z})$ in (1) by an approximate action-value function $\hat{Q}(\boldsymbol{z})$.

Assumption 1 (Compatibility)**.** *Suppose that $\hat{Q}(\boldsymbol{z})$ is parametrized by a vector $\boldsymbol{w}$ of n parameters, $\hat{Q}(\boldsymbol{z}) = \hat{Q}(\boldsymbol{z};\boldsymbol{w})$, then $\nabla_{\boldsymbol{w}}\hat{Q}(\boldsymbol{z};\boldsymbol{w}) = \nabla_{\boldsymbol{\theta}}\log(\mu(\boldsymbol{a}|\boldsymbol{x};\boldsymbol{\theta}))$.*

The BAC model consists of a non-parametric Bayesian critic and a parametric Bayesian actor. First the critic computes a posterior distribution over action-value functions using the data generated by the current policy. Then, the actor uses the posterior moments calculated by the critic and the observed data, and computes a posterior distribution over policy gradients. The posterior mean serves as the estimate of the policy gradient and is used to update the policy, while the posterior covariance serves as a measure for the reliability of this update. In the following, we first describe how each component of BAC is formulated, and then finish the paper by a sketch of the BAC algorithm.

Bayesian Actor is responsible for computing a posterior distribution for the policy gradient given the sequence of individual observed transitions $\mathcal{D}_t$. We start with the expression for the policy gradient given in (1). We place a GP prior over action-value functions using a prior covariance kernel defined on state-action pairs $k(\boldsymbol{z},\boldsymbol{z}') = \mathbf{Cov}[Q(\boldsymbol{z}),Q(\boldsymbol{z}')]$. Making use of the linearity of (1) in Q, and denoting $\boldsymbol{g}(\boldsymbol{z};\boldsymbol{\theta}) = \pi(\boldsymbol{z};\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}}\log\big(\mu(\boldsymbol{a}|\boldsymbol{x};\boldsymbol{\theta})\big)$, we obtain the posterior mo-

[1] This extended abstract is a summary of the work in Ghavamzadeh and Engel (2007a).

ments of the policy gradient given the observed data

$$\mathbf{E}[\nabla_{\boldsymbol{\theta}}\eta|\mathcal{D}_t] = \int_{\mathcal{Z}} d\boldsymbol{z}\boldsymbol{g}(\boldsymbol{z};\boldsymbol{\theta})\mathbf{E}\big[Q(\boldsymbol{z})|\mathcal{D}_t\big], \tag{2}$$

$$\mathbf{Cov}[\nabla_{\boldsymbol{\theta}}\eta|\mathcal{D}_t] = \int_{\mathcal{Z}^2} d\boldsymbol{z}d\boldsymbol{z}'\boldsymbol{g}(\boldsymbol{z};\boldsymbol{\theta})\mathbf{Cov}\big[Q(\boldsymbol{z}),Q(\boldsymbol{z}')|\mathcal{D}_t\big]\boldsymbol{g}(\boldsymbol{z}';\boldsymbol{\theta})^\top.$$

Bayesian Critic is responsible for providing the actor with the posterior moments of $Q(\boldsymbol{z})$. Fortunately, GPTD (Engel et al., 2005) provides a well-developed machinery for this procedure. The GPTD model is a direct application of GP regression in TD learning. It is based on a statistical generative model relating the observed reward signal to the unobserved action-value function. Under certain assumptions on the distribution of the discounted return random process (Engel et al., 2005), we can obtain the posterior moments of Q as

$$\begin{aligned}\mathbf{E}\big[Q(\boldsymbol{z})|\mathcal{D}_t\big] &= \boldsymbol{k}_t(\boldsymbol{z})^\top\boldsymbol{\alpha}_t, \\ \mathbf{Cov}\big[Q(\boldsymbol{z}),Q(\boldsymbol{z}')|\mathcal{D}_t\big] &= k(\boldsymbol{z},\boldsymbol{z}') - \boldsymbol{k}_t(\boldsymbol{z})^\top\boldsymbol{C}_t\boldsymbol{k}_t(\boldsymbol{z}').\end{aligned} \tag{3}$$

Now we return to the actor and substitute the expressions for the posterior moments of the action-value function in (2) with the critic's results of (3), we get

$$\begin{aligned}\mathbf{E}[\nabla_{\boldsymbol{\theta}}\eta|\mathcal{D}_t] &= \int_{\mathcal{Z}} d\boldsymbol{z}\boldsymbol{g}(\boldsymbol{z};\boldsymbol{\theta})\boldsymbol{k}_t(\boldsymbol{z})^\top\boldsymbol{\alpha}_t, \\ \mathbf{Cov}[\nabla_{\boldsymbol{\theta}}\eta|\mathcal{D}_t] &= \\ \int_{\mathcal{Z}^2} d\boldsymbol{z}d\boldsymbol{z}'\boldsymbol{g}(\boldsymbol{z};\boldsymbol{\theta})&\big(k(\boldsymbol{z},\boldsymbol{z}') - \boldsymbol{k}_t(\boldsymbol{z})^\top\boldsymbol{C}_t\boldsymbol{k}_t(\boldsymbol{z}')\big)\boldsymbol{g}(\boldsymbol{z}';\boldsymbol{\theta})^\top.\end{aligned}$$

The equations above provide us with the general form of the posterior policy gradient moments. We are now left with a computational issue; namely, how to compute the integrals in these expressions? It can be shown (Ghavamzadeh & Engel, 2007a) that by choosing the prior covariance to be the sum of an arbitrary state-kernel and the Fisher kernel over state-action pairs, the integrals can be computed analytically.

Proposition 1. *Let $k(\boldsymbol{z},\boldsymbol{z}') = k_x(\boldsymbol{x},\boldsymbol{x}') + k_F(\boldsymbol{z},\boldsymbol{z}')$ for all $(\boldsymbol{z},\boldsymbol{z}') \in \mathcal{Z}^2$, where $k_x : \mathcal{X}^2 \to \mathbb{R}$ is an arbitrary positive definite kernel function, and $k_F(\boldsymbol{z},\boldsymbol{z}') = \boldsymbol{u}(\boldsymbol{z})^\top\boldsymbol{G}^{-1}\boldsymbol{u}(\boldsymbol{z}')$ is the Fisher kernel in which $\boldsymbol{u}(\boldsymbol{z}) = \nabla_{\boldsymbol{\theta}}\log\big(\mu(\boldsymbol{a}|\boldsymbol{x};\boldsymbol{\theta})\big)$ and $\boldsymbol{G} = \mathbf{E}\big[\boldsymbol{u}(\boldsymbol{z})\boldsymbol{u}(\boldsymbol{z})^\top\big]$ are the score vector and the Fisher information matrix corresponding to policy $\mu(\boldsymbol{\theta})$, respectively. Then*

$$\begin{aligned}&\mathbf{E}[\nabla_{\boldsymbol{\theta}}\eta|\mathcal{D}_t] = \boldsymbol{U}_t\boldsymbol{\alpha}_t\ , \quad \boldsymbol{U}_t = \big[\boldsymbol{u}(\boldsymbol{z}_0),\boldsymbol{u}(\boldsymbol{z}_1),\ldots,\boldsymbol{u}(\boldsymbol{z}_t)\big], \\ &\mathbf{Cov}\left[\nabla_{\boldsymbol{\theta}}\eta|\mathcal{D}_t\right] = \boldsymbol{G} - \boldsymbol{U}_t\boldsymbol{C}_t\boldsymbol{U}_t^\top.\end{aligned}$$

An immediate consequence of Proposition 1 is that, in order to compute the posterior moments of the policy gradient, we only need to be able to evaluate (or estimate) the score vectors $\boldsymbol{u}(\boldsymbol{z}_i)$ and the Fisher information matrix $\boldsymbol{G}$ of the policy. Algorithm 1 is a pseudocode sketch of the BAC algorithm, using either the *regular* or the *natural* gradient in the policy update, and with $\boldsymbol{G}$ estimated using $\hat{\boldsymbol{G}}_t$.

Algorithm 1 Bayesian Actor-Critic

BAC$(\boldsymbol{\theta}, M, \epsilon)$
- $\boldsymbol{\theta}$ initial policy parameters
- $M > 0$ trajectories for gradient evaluation
- $\epsilon > 0$ termination threshold

done = false
while not done **do**
  Run **GPTD** for M episodes. **GPTD** returns $\boldsymbol{\alpha}_t, \boldsymbol{C}_t, \boldsymbol{U}_t, \hat{\boldsymbol{G}}_t$
  $\Delta\boldsymbol{\theta} = \boldsymbol{U}_t\boldsymbol{\alpha}_t$ (regular gradient) or $\Delta\boldsymbol{\theta} = \hat{\boldsymbol{G}}_t^{-1}\boldsymbol{U}_t\boldsymbol{\alpha}_t$ (natural gradient)
  $\boldsymbol{\theta} := \boldsymbol{\theta} + \beta\Delta\boldsymbol{\theta}$
  if $|\Delta\boldsymbol{\theta}| < \epsilon$ **then** done = true
end while
return $\boldsymbol{\theta}$

3. Discussion

The BAC algorithms discussed in this paper present a Bayesian framework for reasoning about value function and policy in RL. The strengths of this approach lie primarily in its ability to provide 1) a non-parametric representation for value functions, and 2) confidence measure on value function predictions and policy gradient estimates. These properties are quite unique and have many potential uses in large MDPs such as dealing with high-dimensional state and action spaces, balancing exploration and exploitation, and determining the size and direction of the policy update. The GPTD approach has been successfully applied to high-dimensional control problems (Engel et al., 2006). We believe its combination with policy learning provides a more powerful tool to tackle high-dimensional problems, specifically those with large and continuous action spaces.

References

Engel, Y., Mannor, S., & Meir, R. (2005). RL with Gaussian processes. *ICML22* (pp. 201–208).

Engel, Y., Szabo, P., & Volkinshtein, D. (2006). Learning to control an octopus arm with Gaussian process temporal difference learning. *NIPS18* (pp. 347–354).

Ghavamzadeh, M., & Engel, Y. (2007a). Bayesian Actor-Critic algorithms. *ICML24* (pp. 297–304).

Ghavamzadeh, M., & Engel, Y. (2007b). Bayesian policy gradient algorithms. *NIPS19* (pp. 457–464).

Konda, V., & Tsitsiklis, J. (2000). Actor-Critic algorithms. *NIPS12* (pp. 1008–1014).

Marbach, P. (1998). *Simulated-based methods for Markov decision processes.* Doctoral dissertation, MIT.

Sutton, R., McAllester, D., Singh, S., & Mansour, Y. (2000). Policy gradient methods for RL with function approximation. *NIPS12* (pp. 1057–1063).