Draft: Online PGPE

Pierpaolo Necchi

May 21, 2016

In PGPE we consider a deterministic controller $F : \mathbb{S} \times \Theta \to \mathbb{A}$ that, given a set of parameters $\theta \in \Theta \subseteq \mathbb{R}^{D_\theta}$, maps a state $s \in \mathbb{S}$ to an action $a = F(s;\theta) = F_\theta(s) \in \mathbb{A}$. The policy parameters are drawn from a probability distribution p_ξ, with hyperparameters $\xi \in \Xi \subseteq \mathbb{R}^{D_\xi}$. Combining these two hypotheses, the agent follows a stochastic policy π_ξ given by

$$\pi_\xi(s,a) = \pi(s,a;\xi) = \int_\Theta p_\xi(\theta) \mathbb{1}_{F_\theta(s)=a} d\theta \tag{1}$$

If we assume that the policy π_ξ is differentiable with respect to ξ, the policy gradient theorem states that the gradient of the objective function is given by

$$\begin{aligned} \nabla_\xi J(\xi) &= \mathbb{E}\left[\nabla_\xi \log \pi_\xi(S,A) Q_{\pi_\xi}(S,A)\right] \\ &= \int_{\mathbb{S}} \mu_\xi(s) \int_{\mathbb{A}} \pi_\xi(s,a) \nabla_\xi \log \pi_\xi(s,a) Q_{\pi_\xi}(s,a) da ds \end{aligned} \tag{2}$$

This expression can be rewritten in a different form that will allow us to formulate various online versions of the PGPE algorithm, similar to the standard policy gradient algorithms. First, by using the likelihood trick we have

$$\begin{aligned} \pi_\xi(s,a) \nabla_\xi \log \pi_\xi(s,a) &= \nabla_\xi \pi_\xi(s,a) \\ &= \int_\Theta \nabla_\xi p_\xi(\theta) \mathbb{1}_{F_\theta(s)=a} d\theta \\ &= \int_\Theta p_\xi(\theta) \nabla_\xi \log p_\xi(\theta) \mathbb{1}_{F_\theta(s)=a} d\theta \\ &= \mathbb{E}_{\theta \sim p_\xi}\left[\nabla_\xi \log p_\xi(\theta) \mathbb{1}_{F_\theta(s)=a}\right] \end{aligned}$$

Then, by exchanging the integral over the action space with the expectation

$$\begin{aligned} \int_{\mathbb{A}} \pi_\xi(s,a) \nabla_\xi \log \pi_\xi(s,a) Q_{\pi_\xi}(s,a) da &= \mathbb{E}_{\theta \sim p_\xi}\left[\nabla_\xi \log p_\xi(\theta) \int_{\mathbb{A}} \mathbb{1}_{F_\theta(s)=a} Q_{\pi_\xi}(s,a) da\right] \\ &= \mathbb{E}_{\theta \sim p_\xi}\left[\nabla_\xi \log p_\xi(\theta) Q_{\pi_\xi}(s, F_\theta(s))\right] \\ &= \mathbb{E}_{\theta \sim p_\xi}\left[\nabla_\xi \log p_\xi(\theta) Q_{\pi_\xi}(s,\theta))\right] \end{aligned}$$

Finally, pluggin this equality in the policy gradient theorem, we obtain

$$\begin{aligned}\nabla_\xi J(\xi) &= \mathbb{E}\left[\nabla_\xi \log p_\xi(\theta) Q_{\pi_\xi}(S,\theta)\right] \\ &= \int_{\mathbb{S}} \mu_\xi(s) \int_\Theta p_\xi(\theta) \nabla_\xi \log p_\xi(\theta) Q_{\pi_\xi}(S,\theta) d\theta ds\end{aligned} \tag{3}$$

This expression is very similar to the original policy gradient theorem, but the expectation is taken over the controller parameters and not on the actions. Thus, we might interpret this result as if the agent chose the parameters θ, which then lead to an action through the deterministic mapping F_θ. The agent policy is in parameters space and not in the control space. As in the standard policy gradient methods, we can add a state-dependent baseline $B(S)$ to the gradient without increasing the bias

$$\nabla_\xi J(\xi) = \mathbb{E}\left[\nabla_\xi \log p_\xi(\theta) \left(Q_{\pi_\xi}(S,\theta) - B(S)\right)\right] \tag{4}$$

Indeed,

$$\begin{aligned}\mathbb{E}\left[\nabla_\xi \log p_\xi(\theta) B(S)\right] &= \int_{\mathbb{S}} \mu_\xi(s) \int_\Theta p_\xi(\theta) \nabla_\xi \log p_\xi(\theta) B(s) d\theta ds \\ &= \int_{\mathbb{S}} \mu_\xi(s) B(s) ds \underbrace{\int_\Theta \nabla_\xi p_\xi(\theta) d\theta}_{\nabla_\xi 1 = 0} = 0\end{aligned}$$

This result can be used to design several actor-only or actor-critic online algorithms that are the parameter-based equivalents of the traditional control-based policy-gradient algorithms, such as REINFORCE, GPOMDP or QAC.