CS 6101 Week #4 Notes: Actor-Critic Introduction, Value Functions and Q-Learning

Note taking: Alexandre Gravier, Joel Lee
LaTeX transcription: Alexandre Gravier <al.gravier@gmail.com>

Contents

1 Recap about policy gradients

We define $J(\theta) \doteq E_{\tau \sim p_\theta(\tau)} \left[\sum_t r\left(\mathbf{s}_t, \mathbf{a}_t\right)\right]$ so that the objective of RL can be defined as an optimization exercise consting in finding an assignment of policy parameters $\theta^\star = \arg\max_\theta E_{\tau \sim p_\theta(\tau)} J(\theta)$.

$J(\theta)$ is not usually optimizable as such due to f.e. dimensionality issues, so we use a sample-based unbiased estimate: $J(\theta) \approx \frac{1}{N} \sum_i \sum_t r\left(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}\right)$. Taking the gradient of $J(\theta)$ along θ allows maximizing the expected reward as per the policy.

1.1 Policy differentiation with a "convenient identity"

Let $r(\tau) \doteq \sum_{t=1}^{T} r\left(\mathbf{s}_t, \mathbf{a}_t\right)$ the total reward of a trajectory τ.

$$
\begin{aligned}
\nabla_\theta J(\theta) &= \nabla_\theta E_{\tau \sim p_\theta(\tau)}[r(\tau)] && && \text{(1a)} \\
&= \nabla_\theta \int \pi_\theta(\tau) r(\tau) d\tau && \text{by definition of expectation} && \text{(1b)} \\
&= \int \nabla_\theta \pi_\theta(\tau) r(\tau) d\tau && \text{by linearity} && \text{(1c)}
\end{aligned}
$$

At this point, the expression $\int \nabla_\theta \pi_\theta(\tau) r(\tau) d\tau$ seems rather intractable. This is where the following "convenient identity" can be used to derive a tractable expression of $\nabla_\theta J(\theta)$.

A convenient identiy

$$\pi_\theta(\tau) \nabla_\theta \log \pi_\theta(\tau) = \pi_\theta(\tau) \frac{\nabla_\theta \pi_\theta(\tau)}{\pi_\theta(\tau)} = \nabla_\theta \pi_\theta(\tau) \tag{2}$$

Exploration in Computer Science Research: Deep Reinforcement Learning (CS 6101, 2019), National University of Singapore.

Furthermore, we can expand the definition of $\pi_\theta(\tau)$ and take its logarithm:

$$\pi_\theta(\tau) = p(\mathbf{s}_1) \prod_{t=1}^{T} \pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t) p(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t) \tag{3a}$$

$$\Leftrightarrow \log \pi_\theta(\tau) = \log p(\mathbf{s}_1) + \sum_{t=1}^{T} \log \pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t) + \log p(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t) \tag{3b}$$

Using (2) and (3b) in (1c), the gradient of the objective becomes:

$$\begin{aligned}
\nabla_\theta J(\theta) &= \int \pi_\theta(\tau) \nabla_\theta \log \pi_\theta(\tau) r(\tau) d\tau && \text{using (2)} && (4a)\\
&= E_{\tau \sim p_\theta(\tau)} \left[\nabla_\theta \log \pi_\theta(\tau) r(\tau)\right] && \text{by definition of expectation} && (4b)\\
&= E_{\tau \sim p_\theta(\tau)} \left[\nabla_\theta \left[\log p(\mathbf{s}_1) + \sum_{t=1}^{T} \log \pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t) + \log p(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)\right] r(\tau)\right] && && (4c)
\end{aligned}$$

We note that in the expression $\nabla_\theta \left[\log p(\mathbf{s}_1) + \sum_{t=1}^{T} \log \pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t) + \log p(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)\right]$ of the gradient w.r.t. θ in (4c), the terms $\log p(\mathbf{s}_1)$ and $\log p(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)$ are independent of θ, so we are left with:

$$\begin{aligned}
\nabla_\theta J(\theta) &= E_{\tau \sim p_\theta(\tau)} \left[\nabla_\theta \left[\sum_{t=1}^{T} \log \pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t)\right] r(\tau)\right] && && (5a)\\
&= E_{\tau \sim p_\theta(\tau)} \left[\left(\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t)\right) r(\tau)\right] && \text{by linearity} && (5b)\\
&= E_{\tau \sim p_\theta(\tau)} \left[\left(\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t)\right) \sum_{t=1}^{T} r(\mathbf{s}_t, \mathbf{a}_t)\right] && \text{by definition of } r(\tau) && (5c)
\end{aligned}$$

In Equation (5c), the gradient of J is now a computable function of π_θ only.

We earlier mentioned the sample estimate of $J(\theta) \approx \frac{1}{N} \sum_i \sum_t r(\mathbf{s}_{i,t}, \mathbf{a}_{i,t})$; similarly $\nabla_\theta J(\theta)$ is approximated with samples, leading us to the algorithm:

$$\textbf{REINFORCE algorithm:} \begin{cases} \text{sample } \left\{\tau^i\right\} \text{ from } \pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t) \\ \nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \left(\left(\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t)\right) \sum_{t=1}^{T} r(\mathbf{s}_t, \mathbf{a}_t)\right) \\ \theta \leftarrow \theta + \alpha \nabla_\theta J(\theta) \end{cases} \tag{6}$$

In (6), there is no use of the Markov property, so the algorithm can be used as such on POMDPs.

1.2 The bad news

1.3 Variance reduction with "rewards to go"

The "**rewards to go**" trick computes the expected future rewards based on from the current time t to reduce variance in J. It comes from the observation that at time t, all past rewards cannot be affected by policy decisions.

Rewards to go -

References