EL2805 Reinforcement Learning

Exercise Session 5

December 1, 2019

Department of Automatic Control
School of Electrical Engineering
KTH Royal Institute of Technology

5 Exercises

Some of these exercises have been inspired by, or taken from, [1]. If you want to solve more exercises, see any of those books.

5.1

How would you explain the parameter-update equation in REINFORCE;

$$\theta^{(k+1)} = \theta^{(k)} + \alpha_k \left(\sum_{t=1}^{T} \nabla \log \pi_\theta(s_{t,k}, a_{t,k}) \right) \left(\sum_{t=1}^{T} r_{t,k} \right),$$

to a bachelor student? What is the intuition/interpretation of all the terms?

5.2

Consider a policy parametrization using soft-max in action preferences:

$$\pi_\theta(s,a) = \frac{e^{h(s,a,\theta)}}{\sum_b e^{h(s,b,\theta)}},$$

with linear action preferences

$$h(s,a,\theta) = \theta^T x(s,a),$$

where $x(s,a)$ are feature vectors. Find the eligibility vector $\nabla \log \pi_\theta(s,a)$.

5.3

Consider an application where our action set is continuous $\mathcal{A} = \mathbb{R}$. We do a policy parametrization using Gaussian functions:

$$\pi_\theta(s,a) = \frac{1}{\sigma(s,\theta)\sqrt{2\pi}} \exp\left\{-\frac{(a-\mu(s,\theta))^2}{2\sigma(s,\theta)^2}\right\}.$$

a) Give three examples of RL tasks where the action sets are continuous.

b) Let $\theta = [\theta_\mu, \theta_\sigma]^T$ and take

$$\mu(s,\theta) = \theta_\mu^T x_\mu(s) \text{ and } \sigma(s,\theta) = \exp\left\{\theta_\sigma^T x_\sigma(s)\right\},$$

where $x(s)$ are feature vectors. Why is the variance parametrized differently?

c) Compute the eligibility vector $\nabla \log \pi_\theta(s,a)$.
Hint: Compute $\nabla_{\theta_\mu} \{\log \pi_\theta(s,a)\}$ and $\nabla_{\theta_\sigma} \{\log \pi_\theta(s,a)\}$ separately.

5.4

Consider again the rock-paper-scissors game. We play against an opponent who selects an action in an i.i.d. manner according to a known distribution $\mu = (\mu_R, \mu_S, \mu_P)$.

a) Model it as an MDP with the following state-space $\mathcal{S} = \{\text{Initial, Win, Lose, Terminal}\}$, deterministic rewards and time-horizon $T = 3$. *Note:* Let the Lose-state represent either a loss or a draw.

Assume now that the opponent's distribution μ is **unknown** to us. We will now need to learn how to play the game (optimally). In order to do so, we will use policy gradient methods.

b) Propose a policy parametrization.

c) Explain how we can use REINFORCE to learn an optimal policy.

d) **(Computer exercise)** Implement the MDP in *a)* (so as to be able to generate trajectories), and your proposed RL algorithm. What policy do you learn? Try different μ:s.

5.5

Assume $u < t$. Prove the *causality principle*:

$$\mathbb{E}_{\pi_\theta}[\nabla \log \pi_\theta(s_t, a_t) r_u] = 0.$$

Note: From this, you can deduce that:

$$\nabla J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=1}^{T} \nabla \log \pi_\theta(s_t,a_t) \underbrace{\sum_{u=t}^{T} r(s_u,a_u)}_{\text{reward to go}}\right].$$

6 Solutions

Note: log denotes the natural logarithm (with base e).

Answer to Problem 5.1

First note that we can expand the update as:

$$\begin{aligned}\theta^{(k+1)} &= \theta^{(k)} + \alpha_k \left(\sum_{t=1}^{T} \nabla \log \pi_\theta(s_{t,k}, a_{t,k}) \right) \left(\sum_{t=1}^{T} r_{t,k} \right) \\ &= \theta^{(k)} + \alpha_k \left(\sum_{t=1}^{T} \frac{\nabla \pi_\theta(s_{t,k}, a_{t,k})}{\pi_\theta(s_{t,k}, a_{t,k})} \right) \left(\sum_{t=1}^{T} r_{t,k} \right),\end{aligned}$$

using a simple rule of calculus.

Each update to the parameters of our policy is proportional to the product of a total return ($\sum r_{t,k}$), and a vector: the gradient of the probability of taking the action actually taken divided by the probability of taking that action. This vector points in the direction in parameter space that most increases the probability of repeating the action we took ($a_{t,k}$) on future visits to the same state ($s_{t,k}$).

The update increases the parameter vector in this direction proportional to the return, and inversely proportional to the action probability. The former makes sense because it causes the parameter to move most in the directions that favor actions that yield the highest return. The latter makes sense because otherwise actions that are selected frequently are at an advantage (the updates will be more often in their direction) and might win out even if they do not yield the highest return.

Solution to Problem 5.2

$$\nabla \log \pi_\theta(s, a) = x(s, a) - \sum_b \pi_\theta(s, b) x(s, b).$$

Solution to Problem 5.3

Part b)
The variance has to be non-negative. This is enforced by the exponential function.
Part c)
We have that

$$\nabla \log \pi_\theta(s, a) = \begin{bmatrix} \nabla_{\theta_\mu} \\ \nabla_{\theta_\sigma} \end{bmatrix} \log \pi_\theta(s, a) = \begin{bmatrix} \nabla_{\theta_\mu} \{\log \pi_\theta(s, a)\} \\ \nabla_{\theta_\sigma} \{\log \pi_\theta(s, a)\} \end{bmatrix}$$

with

$$\nabla_{\theta_\mu} \{\log \pi_\theta(s, a)\} = \frac{1}{\sigma(s, \theta)^2} \left(a - \mu(s, \theta)\right) x_\mu(s)$$

and

$$\nabla_{\theta_\sigma} \{\log \pi_\theta(s, a)\} = \left(\frac{(a - \mu(s, \theta))^2}{\sigma(s, \theta)^2} - 1 \right) x_\sigma(s).$$

Solution to Problem 5.4

Part a)

- **State space:** $\mathcal{S} = \{\text{Initial, Win, Lose, Terminal}\}$.
- **Actions:** $\mathcal{A}(\text{Initial}) = \{R, P, S\}$, $\mathcal{A}(\text{Win, Lose, Terminal}) = \{\text{Continue}\} = \{C\}$.
- **Rewards:**

  - $r(s = \text{Win}, a = C) = 1$,
  - all other zero.

- **Transitions:**
  - $p(s' = \text{Win}|s = \text{Initial}, a = R) = \mu_S$ (We play Rock and opponent plays Scissors),
  - $p(s' = \text{Lose}|s = \text{Initial}, a = R) = 1 - \mu_S$,
  - $p(s' = \text{Win}|s = \text{Initial}, a = S) = \mu_P$,
  - $p(s' = \text{Lose}|s = \text{Initial}, a = S) = 1 - \mu_P$,
  - $p(s' = \text{Win}|s = \text{Initial}, a = P) = \mu_R$,
  - $p(s' = \text{Lose}|s = \text{Initial}, a = P) = 1 - \mu_R$,
  - $p(s' = \text{Terminal}|s = \text{Win}, a = C) = 1$,
  - $p(s' = \text{Terminal}|s = \text{Lose}, a = C) = 1$,
  - $p(s' = \text{Terminal}|s = \text{Terminal}, a = C) = 1$.
- **Time-horizon and objective:** Finite-horizon $T = 3$, $\mathbb{E}\{\sum_{t=1}^{T} r(s_t, a_t)\big|s_1 = \text{Initial}\}$.

Part b)

We can, for example, use a softmax policy with linear action preferences

$$\pi_\theta(s, a) = \begin{cases} \frac{\exp\{\theta^T x(s,a)\}}{\sum_b \exp\{\theta^T x(s,b)\}} & \text{if } s = \text{Initial}, a \in \mathcal{A}(\text{Initial}), \\ 1 & \text{otherwise.} \end{cases}$$

Since we only have one action to chose from in {Win, Lose, Terminal}, we do not need to parametrize the policy there. The simplest feature vector is a one-hot encoding:

$$x(s = \text{Initial}, a) = \begin{bmatrix} \mathbb{I}\{a = R\} \\ \mathbb{I}\{a = S\} \\ \mathbb{I}\{a = P\} \end{bmatrix},$$

where $\mathbb{I}$ is the indicator function. Note that, for this choice, $\theta \in \mathbb{R}^3$.

Part c)

In REINFORCE, we first select the initial parameters $\theta^{(0)}$ arbitrarily.

We then repeatedly play the game (i.e., generate a trajectory/episode) under the policy induced by $\theta^{(k)}$. After finishing the episode, we update our parameters using

$$\theta^{(k+1)} = \theta^{(k)} + \alpha_k \left(\sum_{t=1}^{T} \nabla \log \pi_\theta(s_{t,k}, a_{t,k}) \right) \left(\sum_{t=1}^{T} r(s_{t,k}, a_{t,k}) \right),$$

where $s_{t,k}$ is the state at time t in the current episode k. The expression for $\nabla \log \pi_\theta(s_{t,k}, a_{t,k})$ was computed in Problem 5.2:

$$\nabla \log \pi_\theta(s_{t,k}, a_{t,k}) = x(s_{t,k}, a_{t,k}) - \sum_b \pi_\theta(s_{t,k}, b) x(s_{t,k}, b).$$

(However, note that $\nabla \log \pi_\theta(s_{t,k} \in \{\text{Win, Lose, Terminal}\}, a_{t,k}) = 0$.)

Solution to Problem 5.5

For $u < t$:

$$\begin{aligned}
\mathbb{E}\left\{\nabla \log \pi_\theta(s_t, a_t) r(s_u, a_u)\right\} &= \sum_{s,a} \sum_{s',a'} \Pr\left\{s_t = s', a_t = a', s_u = s, a_u = a\right\} r(s,a) \nabla \log \pi_\theta(s', a') \\
&= \sum_{s,a} \sum_{s',a'} \Pr\left\{a_t = a' | s_t = s'\right\} \Pr\left\{s_t = s', s_u = s, a_u = a\right\} r(s,a) \nabla \log \pi_\theta(s', a') \\
&= \sum_{s,a} \sum_{s',a'} \pi_\theta(s', a') \Pr\left\{s_t = s', s_u = s, a_u = a\right\} r(s,a) \nabla \log \pi_\theta(s', a') \\
&= \sum_{s,a} \sum_{s'} \Pr\left\{s_t = s', s_u = s, a_u = a\right\} r(s,a) \underbrace{\sum_{a'} \pi_\theta(s', a') \nabla \log \pi_\theta(s', a')}_{=0} \\
&= 0,
\end{aligned}$$

where the second equality follows from the Markovian assumptions on the system and the policy. The last factor is zero because:

$$\begin{aligned}
\sum_{a'} \pi_\theta(s', a') \nabla \log \pi_\theta(s', a') &= \sum_{a'} \pi_\theta(s', a') \frac{\nabla \pi_\theta(s', a')}{\pi_\theta(s', a')} \\
&= \sum_{a'} \nabla \pi_\theta(s', a') \\
&= \nabla \sum_{a'} \pi_\theta(s', a') \\
&= \nabla 1 \\
&= 0.
\end{aligned}$$

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction.* MIT press, 1998.