Introduction to Reinforcement Learning

Edward Young - ey245@cam.ac.uk

States, Actions, and Rewards

- RL involves an interaction between an **agent** and an **environment**.
- The agent chooses an **action** to take based upon the current **state** of the environment.
- As a result of that action, the environment returns a **reward**, and transitions to a new state.
- The collection of states for the environment is the **state space**, which we denote by $\mathcal{S}$
- The collection of actions for the environment is the **action space**, which we denote by $\mathcal{A}$
- The environment is characterised by a **dynamics** function and a **reward** function.
  1. The dynamics function gives the probability of transitioning into state S', given that you take action A in state S. This is denoted $p(S'|S, A)$.
  2. The reward function indicates the reward that the agent receives for being in state S and taking action A. This is denoted $R(S, A)$.
- The interaction between the agent and the environment continues until we reach a **terminal state**.

Policies

An agent's **policy** dictates how the agent behaves in response to being in a particular state.

1. A **deterministic policy** takes in a state, and returns the action that the agent takes in that state. We will denote deterministic policies by $\mu(S) \in \mathcal{A}$
2. A **stochastic policy** takes in a state, and returns a probability distribution over actions that the agent could take. We will denote stochastic policies by $\pi(A|S)$

Reward, Return, and Value

The **return** at time t is the sum of rewards obtained after time t until the time T at which the terminal state is reached, discounted according to how far in the future they are:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots + \gamma^{T-t-1} R_T = \sum_{k=0}^{T-t-1} \gamma^k R_{t+k+1}$$

We call γ the **discount rate**. The discount rate quantifies how much we weight we place on nearby rewards vs. distant rewards.

1. The **state value function** is the expected return, conditional on being in a particular state:

$$V_\mu(S) = \mathbb{E}_\mu[G_t|S_t = S]$$

2. The **action value function** is the expected return, conditional on being in a particular state S and taking a particular action A:

$$Q_\mu(S, A) = \mathbb{E}_\mu[G_t|S_t = S, A_t = A]$$

Note that value functions *depend on the policy*: the value of being in a particular state (or being in a state and taking an action) depends on how we behave afterwards.

Optimal Policies and Values

We define the **optimal action value function** to be the greatest possible value, over all possible policies:

$$Q^*(S, A) = \max_{\mu} Q_\mu(S, A)$$

The **optimal policy** is the policy which, for each state, selects the action which maximises the optimal action value function:

$$\mu^*(S) = \arg\max_A Q^*(S, A)$$

The optimal action value function is the action value function for the optimal policy, *i.e.*

$$Q^*(S, A) = Q_{\mu^*}(S, A)$$

The optimal value function satisfies the **Bellman optimality equation**

$$Q^*(S, A) = R(S, A) + \gamma \sum_{S' \in \mathcal{S}} p(S'|S, A) \max_{A'} Q^*(S', A')$$

Types of RL algorithms

- **Model-based vs. Model-free**. A model-based method is a method that requires us to either use the dynamics function or an estimate of it. A model-free method does not have such requirements. Almost all algorithms we will be covering are model-free.
- **On-policy vs. Off-policy**. An on-policy algorithm attempts to learn the value function of the policy being used by the agent. An off-policy algorithm attempts to learn a different value function, typically the optimal value function. We will start with off-policy algorithms and then move onto on-policy algorithms.
- **Action-value vs. policy-gradient**. An action-value method attempts to learn only a value function. A policy-gradient method additionally attempts to learn a policy as well. We will begin with an action-value method and then move onto policy-gradient methods.

Appendix: Mathematical notation

Here we list some mathematical notation and symbols for those who haven't encountered them before:

- $\in$ should be read as *in*
- $|$ should be read as *given that*
- $\mathbb{E}$ should be read as *the expected value of*
- $\sum$ should be read as *the sum of*
- max is the largest value of something
- $\arg\max$ is the value which maximises something, *i.e.*the value at which the maximum is attained