Contents 0

Chapter 1

Reinforcement learning is about how an agent can learn to interact with its environment. Reinforcement learning uses the formal framework of Markov decision processes to define the interaction between a learning agent and its environment in terms of states, actions, and rewards.

1.3 Elements of Reinforcement Learning

Policy defines the way that an agent acts, it is a mapping from perceived states of the world to actions. It may be stochastic.

Reward defines the goal of the problem. A number given to the agent as a (possibly stochastic) function of the state of the environment and the action taken.

Value function specifies what is good in the long run, essentially to maximise the expected reward. The central role of value estimation is arguably the most important thing that has been learned about reinforcement learning over the last six decades.

Model mimics the environment to facilitate planning. Not all reinforcement learning algorithms have a model (if they don't then they can't plan, i.e. must use trial and error, and are called model free).

Chapter 2

Reinforcement learning involves evaluative feedback rather than instructive feedback. We get told whether our actions are good ones or not, rather than what the single best action to take is. This is a key distinction between reinforcement learning and supervised learning.

2.1 A k-armed Bandit Problem

In the k-armed bandit problem there are k possible actions, each of which yields a numerical reward drawn from a stationary probability distribution for that action. We want to maximise the expected total reward, taking an action at each *time step*. Some notation:

- Index timesteps by t
- Action A_t
- Corresponding reward R_t
- *Value* of action a is $q_*(a) = \mathbb{E}[R_t|A_t = a]$
- Estimate of value of action a at t is denoted $Q_t(a)$

We therefore want to choose $\{a_1, \ldots, a_T\}$ to maximise $\sum_{t=1}^{T} q_*(a_t)$.

At each timestep, the actions with the highest estimated reward are called the *greedy* actions. If we take this action, we say that we are *exploiting* our understanding of the values of actions. The other actions are known as *non-greedy* actions, sometimes we might want to take one of these to improve our estimate of their value. This is called *exploration*. The balance between exploration and exploitation is a key concept in reinforcement learning.

2.2 Action-value Methods

We may like to form estimates of the values of possible actions and then choose actions according to these estimates. Methods such as this are known as *action-value methods*. There are, of course, many ways of generating the estimates $Q_t(a)$.

An ε-greedy method is one in which with probability ε we take a random draw from all of the actions (choosing each action with equal probability), providing some exploration.

2.6 Tracking a Non-stationary Problem

If we decide to implement the sample average method, then at each iteration that we choose the given action we update our estimate by

$$Q_{n+1} = Q_n + \frac{1}{n}[R_n - Q_n] \tag{1}$$

Note that this has the (soon to be familiar) form

$$\text{NewEstimate} \leftarrow \text{OldEstimate} + \text{StepSize} \times [\text{Target} - \text{OldEstimate}]. \tag{2}$$

If the problem was non-stationary, we might like to use a time weighted exponential average for our estimates (*exponential recency-weighted average*). This corresponds to a constant step-size $\alpha \in (0, 1]$ (you can check).

$$Q_{n+1} = Q_n + \alpha[R_n - Q_n]. \tag{3}$$

We might like to vary the step-size parameter. Write $\alpha_n(a)$ for the step-size after the n^{th} reward from action a. Of course, not all choices of $\alpha_n(a)$ will give convergent estimates of the values of a. To converge with probability 1 we must have

$$\sum_n \alpha_n(a) = \infty \qquad \text{and} \qquad \sum_n \alpha_n(a)^2 < \infty. \tag{4}$$

Meaning that the coefficients must be large enough to recover from initial fluctuations, but not so large that they don't converge in the long run. Although these conditions are used in theoretical work, they are seldom used in empirical work or applications. (Most reinforcement learning problems have non-stationary rewards, in which case convergence is undesirable.)