Notes for chapter 1

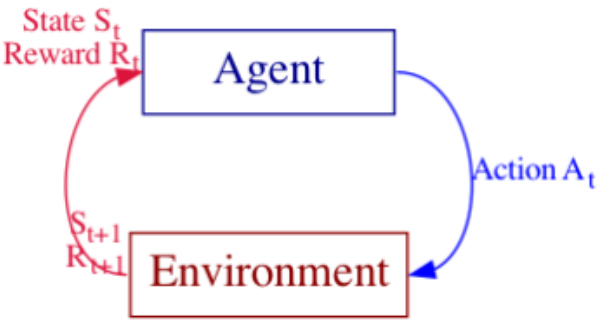
Reinforcement Learning jargon

Reinforcement Learning solves problems that involve making *Sequential Optimal Decisions under Uncertainty*. Let's break down this sentence:

- *Uncertainty* means that the problem involves random variables that evolve over time. I.e. a *stochastic process*.
- *Optimal Decisions*, refers to the technical term *Optimization*. This means that there is a well-quantity to be maximized.
- *Sequential*, refers to the fact that as we forward in time, the random variables and the maximization goal evolve and need to be adjusted.

Putting together the three notions, these problems that are solved by RL have the common feature of *overpowering the uncertainty by persisent steering towards the goal*. We refer to the class of problems that are solved by RL with the technical term *Stochastic Control*.

Introduction to the Markov Decision Process Framework



The above image shows the general MDP framework. The *Agent* is an AI algorithm and the *Environment* is an abstract entity that serves up uncertain outcomes to the Agent. At each time step t, the Agent observes an abstract piece of information (the State) and a numerical (real number) quantity, the Reward. Upon observing a State and Reward at time step t, the Agent responds by taking some kind of Action. This action represents some activity performed by the AI algorithm. Upon recieving the Action from the Agent the Environment responds by serving up the next time step's random State and random Reward. The State is assumed to have the Markov Property, which means

- The next State/Reward depends only on the current State (for a given Action).
- The current State encapsulates all relevant information from the history of the interaction between the Agent and the Environment.

- The current State is a sufficient statistic of the future (for a given Action).

The goal of the Agent at each t is to maximize the Expected Sum of all future Rewards by controlling the Action as a function of the observed state. The function that determines the Action based on the State is the *Optimal Policy* function.

Denote the time steps as $t = 1, 2, 3, ...4$. The Markov State at t is denoted as $S_t \in \mathcal{S}$, where $\mathcal{S}$ is the State Space. Action at t is denoted as $A_t \in \mathcal{A}$, where $\mathcal{A}$ is the Action Space. Reward at t is denoted as $R_t \in \mathcal{D}$, where $\mathcal{D} \subset \mathbb{R}$ is countable and represents the feedback served by the Environment, along with the State, at time step t.

We represent the transition probabilities from one time step to the next with the following notation

$$p(r, s'|s, a) = \mathbb{P}[(R_{t+1} = r, S_{t+1} = s')|S_t = s, A_t = a]$$

$\gamma \in [0, 1]$ is known as the discount factor used to discount Rewards when accumulating Rewards, as follows:

$$\text{Return } G_t = R_{t+1} + \gamma \cdot R_{t+2} + \gamma^2 \cdot R_{t+3} + ... + \gamma^{k-1} \cdot R_{t+k}$$

γ allows us to model situations where a future reward is less desirable than a current ward of the same quantity. The goal is to find a Policy $\pi : \mathcal{S} \to \mathcal{A}$ that maximizes $\mathbb{E}[G_t|S_t = s]$, $\forall s \in \mathcal{S}$.

Value Function, Bellman Equations, Dynamic Programming and RL

The Value Function $V^\pi : \mathcal{S} \to \mathbb{R}$ for a given Policy π is defined as:

$$V^\pi(s) = \mathbb{E}_{\pi,p}[G_t|S_t = s], \forall s \in \mathcal{S}$$

The Value Function tells us how much "accumulated future reward" (i.e. Return) we expect to obtain from a given state. The randomness under the expectation comes from the uncertain future states and rewards the Agent is going to see (based on p). The Value Function for a given policy can be expressed recursively:

$$V^\pi(s) = \sum_{r,s'} p(r, s'|s, \pi(s)) \cdot (r + \gamma \cdot V^\pi(s')), \forall s \in \mathcal{S}$$

When the agent follows a deterministic policy π, in a given state s, it takes an action $a = \pi(s)$, then sees a random next state s' and a random reward r. So $V^\pi(s)$ can be broken into the expectation of r and the remainder of the future expected accumulated rewards.

The Optimal Value Function $V^* : \mathcal{S} \to \mathbb{R}$ is defined as:

$$V^*(s) = \max_\pi V^\pi(s) = \max_\pi p(r, s'|s, \pi(s)) \cdot (r + \gamma \cdot V^\pi(s')), \forall s \in \mathcal{S}$$

Furthermore we can prove there exists an Optimal Policy π^* achieving $V^*(s)$, $\forall s \in \mathcal{S}$. That is,

$$V^{\pi^*}(s) = V^*(s), \forall s \in \mathcal{S}$$

The problem of calculating $V^\pi(s)$ (Value Function for a given poliy) is known as the *Prediction* problem, since this amounts to statistical estimation of the expected returns from any given state, under some Policy π.

The problem of calculating the Optimal Value Funtion $V^*(s)$, is known as the *Control* problem (since this requires steering of the policy s.t. we obtain the maximum expected return from any given state).