Reinforcement Learning

Clayton W. Seitz

November 18, 2024

Markov Decision Processes

- A **decision process** that takes place in discrete time
- The *agent* is embedded in an environment and can make *actions* $A_t \in \mathcal{A}$
- The agent has a state $S_t \in \mathcal{S}$
- The process is called finite if the sets $\mathcal{A}, \mathcal{S}$ are finite
- The agent received a *reward* at each time step R_t

Markov Decision Processes

There is a joint distribution on the reward and next state, given the current state and the taken action

$$p(s', r|s, a) = \mathrm{Pr}\left(S_t = s', R_t = r|S_{t-1} = s, A_t = a\right)$$

Note that this distribution models the reward given state action pairs. It is due to the stochastic nature of the environment

Returns

The cumulative reward received in the future:

$$G_t = R_{t+1} + R_{t+2} + ...$$

It can be discounted to change the timescale of reward focused on

$$G_t = R_{t+1} + \gamma G_{t+2} + \gamma^2 G_{t+3} ... = R_{t+1} + \gamma G_{t+1}$$

for $0 \leq \gamma \leq 1$

The Policy

The *policy* is a prescription for actions to take given the state. Formally it is a conditional distribution on actions given the state $\pi(a|s)$. It is often something to be learned

The *value* of a state given by the policy is the expected return when following that policy from a state s

$$v_\pi(s) = \mathbb{E}_\pi \left[G_t | S_t = s \right]$$

Note we need $p(s', r|s, a)$ to compute this

The Policy

The value function is explicitly computed by the Bellman equation

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) \left[r + \gamma v_\pi(s')\right]$$

It states that the value of the start state must equal the (discounted) value of the expected next state, plus the reward expected along the way.

Policy estimation

The objective is to maximize the total reward, given some initial starting state. We therefore need to find an optimal policy $\pi(a|s)$ to learn how to act under stochastic dynamics of the environment i.e., $p(s', r|s, a)$
$|0.2in$
To actually *evaluate* $v_\pi(s)$ (under a fixed policy), we can use, for example iterative policy evaluation

This can then be used to find better policies

Policy improvement theorem

A policy π is better if it has a higher value (expected return) for all states