Dynammic Programming & Monte Carlo Methods

DP: mathematical optimization method & computer programming method that refers to simplifying a problem by breaking it down to simpler sub-problems in a recursive manner

Policy Evaluation & Policy Iteration

$V_\pi(s)$ - Expected reward starting at s & following π
- Quantifies how good it is tbe in s using π

$Q_\pi(s,a)$ - Expected reward starting at s & following policy π, taking action a

π^* - optimal policy with highest long term reward

Policy Iteration
- have initial policy π_0 & value fn V_{π_0}
- Evaluate π_i & find V_{π_i}
- Improve V_{π_0} using greedy strategy
- Converge to π^*

$\cdot \pi_0 \xrightarrow{E} V_{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} V_{\pi_1} \cdots$

MC: a broad class of computational algorithms that rely on repeated sampling to obtain numerical results

Monte Carlo Action Value Estimation

To estimate $V_\pi(s)$ for a given π

Input: π to be evaluated
Initialize:
$V(s) \in \mathbb{R}$ arbitrarily

Loop
Generate an episode following π:
$S_0, A_0, R_1, S_1, \ldots, R_T$
$G \leftarrow 0$
Loop
$G \leftarrow \gamma G + R_{t+1}$
$V(S_t) \leftarrow$ average (G)

Only State

Action Values

- we must estimate value of each action

Goal: Estimate q_*

- we visit (s, a) pairs probabilistically
- we want to make sure every (s, a) is visited

Monte Carlo Control

How can we use MC to approximate π_*

$$\pi_0 \xrightarrow{E} q_{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} q_{\pi_1} \xrightarrow{I} \pi_2 \cdots$$

MC with Exploring Starts to estimate $\pi \simeq \pi_*$

Initialize $\pi(s) \in A(s)$
$Q(s,a) \in \mathbb{R}$ $s \in S$, $a \in A(s)$

Loop
Choose $S_0 \in S$, $A_0 \in A(S_0)$ randomly
Generate Episode from $S_0 A_0 \ldots R_T$
Get G
$Q(S_t, A_t) \leftarrow$ Average $(S_t A_t)$
$\pi(S_t) \leftarrow \max Q(S, a)$