RL - W5 - Monte Carlo Methods I

5.2 Monte Carlo Estimation of Action Values

$$\begin{aligned}\pi(s) &= \text{argmax}_a q(s,a) && (1)\\ &= \text{argmax}_a \sum_{s',r} p(s',r|s,a)[r+\gamma v(s')] && (2)\end{aligned}$$

With a model of the environment ($p(s',r|s,a)$), state values alone are sufficient to determine a policy; one simply looks ahead one step and chooses whichever action leads to **the best combination of reward and next state**.

For example, in the GridWorld example, given a specific box, $p(s',r|s,a)=1$ for all nearby boxes when actions are chosen correspondingly (e.g. if you want to go to the box on the left and your action is to move to the left, then you will always end up in the box on the left). $p(s',r|s,a)=0$ if otherwise. In this context, the action that maximizes the expression $\sum_{s',r} p(s',r|s,a)[r+\gamma v(s')]$ is the one that assigns probability 1 to the best combination of r and $v(s')$.

Without a model, however, state values alone are not sufficient, because computing the expression $\text{argmax}_a \sum_{s',r} p(s',r|s,a)[r+\gamma v(s')]$ requires knowledge of r's. Since the specific values of r's are part of the environment and we assume no knowledge of the environment, we do not have knowledge of the values of r's . Therefore, instead of just estimating the values of states, we directly estimate the values of state-action pairs and perform the argmax directly according to equation (1).

The only complication is that many state–action pairs may never be visited. If π is a deterministic policy, then in following π one will observe returns only for one of the actions from each state.

This is the general problem of maintaining exploration, there are two methods to help with the problem.

- Exploring starts. By specifying that the episodes start in a state–action pair, and that every pair has a nonzero probability of being selected as the start.
- Consider only policies that are stochastic with a nonzero probability of selecting all actions in each state.

5.3 Monte Carlo Control (with "exploring starts")

In policy iteration one maintains both an approximate policy and evaluation an approximate value function.

For the moment, let us assume that we do indeed observe an infinite number of episodes and that, in addition, the episodes are generated with exploring starts. Under these assumptions, the Monte Carlo methods will compute each q_{π_k} exactly, for arbitrary π_k.

For any action-value function q, the corresponding greedy policy is the one that, for each $s \in S$, deterministically chooses an action with maximal action-value:

$$\pi(s) \doteq \text{argmax}_a q(s,a) \qquad (3)$$

Policy improvement then can be done by constructing each π_{k+1} as the greedy policy with respect to q_{π_k}.

$$\pi_{k+1}(s) = \text{argmax}_a q_{\pi_k}(s,a) \qquad (4)$$

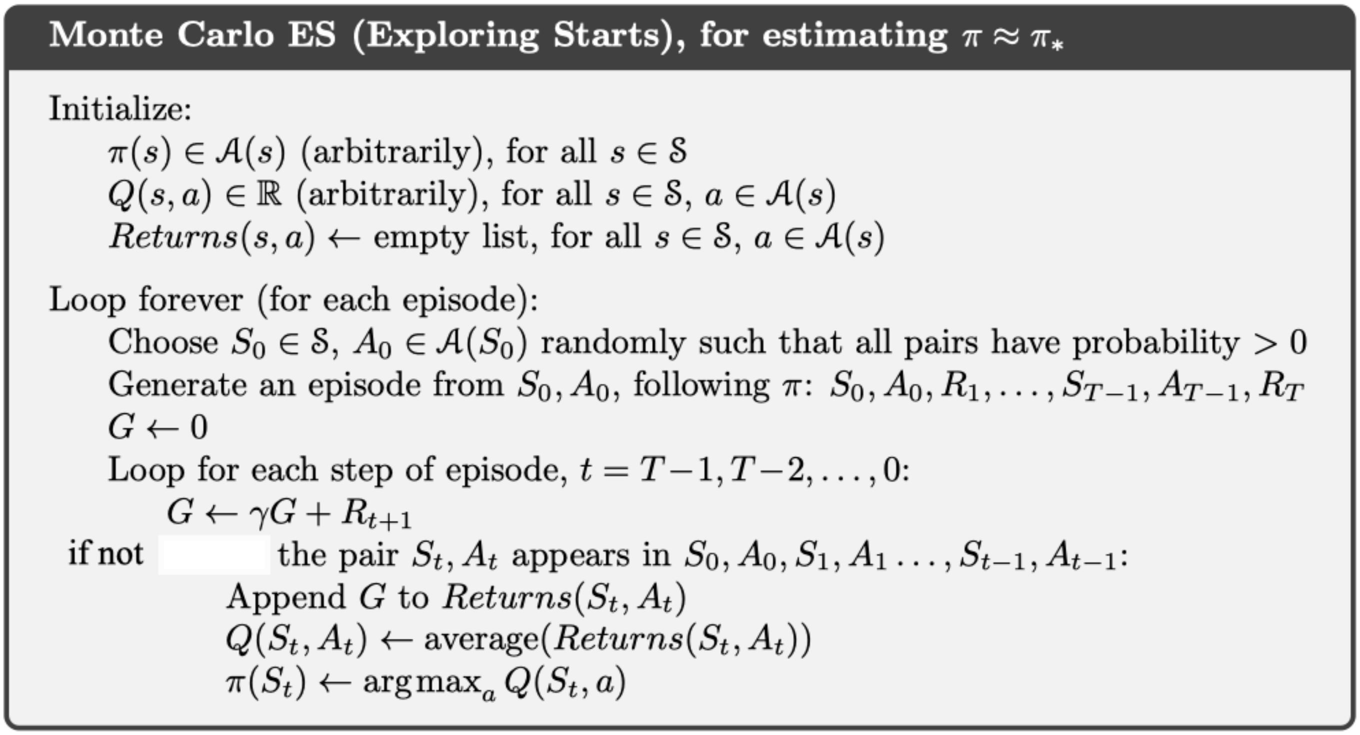
In the previous dynamic programming chapter we have demonstrated that, by using this strategy of acting greedily w.r.t to the current value function, the new value function is strictly greater than the current value function when the current value function is not optimal and the new value function is equal to the current function only when the current value function is optimal.

Two unlikely assumptions

- Episodes have exploring starts
- Policy evaluation could be done with an infinite number of episodes (**the book tackles the second one first)**
    - first approach: hold firm to the idea of approximating q_{π_k} in each policy evaluation
        - allows a boundary of errors
    - second approach: give up trying to complete policy evaluation before returning to policy improvement.
        - don't evaluate the values of all state-action pair / estimate them for a fixed number of times, regardless of whether convergence occurs

This algorithm is still using first-visit of state-action pairs

Note: a G value is accumulated in each Returns(s, a) per outer loop



Monte Carlo ES (Exploring Starts), for estimating $\pi \approx \pi_*$

```
Initialize:
    $\pi(s) \in \mathcal{A}(s)$ (arbitrarily), for all $s \in \mathcal{S}$
    $Q(s,a) \in \mathbb{R}$ (arbitrarily), for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$
    $Returns(s,a) \leftarrow$ empty list, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

Loop forever (for each episode):
    Choose $S_0 \in \mathcal{S}$, $A_0 \in \mathcal{A}(S_0)$ randomly such that all pairs have probability $> 0$
    Generate an episode from $S_0, A_0$, following $\pi$: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
    $G \leftarrow 0$
    Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
        $G \leftarrow \gamma G + R_{t+1}$
  if not          the pair $S_t, A_t$ appears in $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:
            Append $G$ to $Returns(S_t, A_t)$
            $Q(S_t, A_t) \leftarrow$ average$(Returns(S_t, A_t))$
            $\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$
```