

Chapter 5

Monte Carlo Methods

- Monte Carlo method
 - Learning from experience
 - Actual experience
 - Simulated experience
 - Learning from sample episodes
 - Sequences of states, actions, and rewards from experience

5.1 Monte Carlo Prediction

- Monte Carlo(MC) methods
 - Learning the state-value function for a given policy
 - Policy π
 - State-value function v_π
 - Estimating v_π from experience
 - Sampling episodes
 - Estimation

$$\begin{aligned}v_\pi(s) &= \mathbb{E}[R_{t+1} + \gamma R_{t+2} + \cdots | S_t = s] \\ &= \mathbb{E}[G_t | S_t = s]\end{aligned}$$

- Nonstationary estimation
$$v_\pi(S_t) \leftarrow v_\pi(S_t) + \alpha(G_t - v_\pi(S_t))$$

- The first-visit MC method
 - $v_{\pi}(s)$ is estimated as the average of the returns following first visits to s
- The every-visit MC method
 - $v_{\pi}(s)$ is estimated as the average of the returns following all visits to s

First-visit MC prediction, for estimating $V \approx v_\pi$

Input: a policy π to be evaluated

Initialize:

$V(s) \in \mathbb{R}$, arbitrarily, for all $s \in \mathcal{S}$

$Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

Loop forever (for each episode):

Generate an episode following π : $S_0, A_0, R_1, S_1, A_1, R_2, \dots, S_{T-1}, A_{T-1}, R_T$

$G \leftarrow 0$

Loop for each step of episode, $t = T-1, T-2, \dots, 0$:

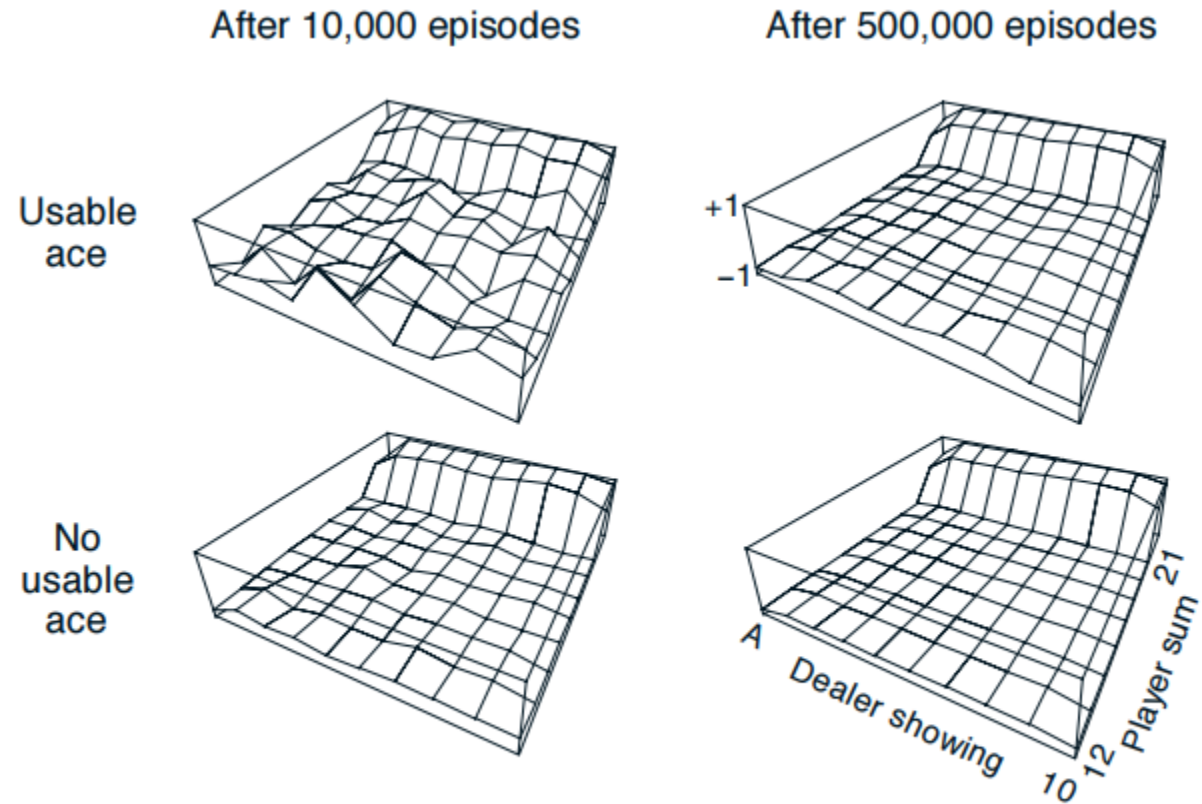
$G \leftarrow \gamma G + R_{t+1}$

Unless S_t appears in S_0, S_1, \dots, S_{t-1} :

Append G to $Returns(S_t)$

$V(S_t) \leftarrow \text{average}(Returns(S_t))$

Example 5.1: Blackjack



Usable ace: ace that can be counted as 11

State: player sum and dealer showing

Action: hit(one more card), stick(stop)

Reward: +1 for winning
-1 for losing
0 for drawing

$\gamma = 1$

Figure 5.1: Approximate state-value functions for the blackjack policy that sticks only on 20 or 21, computed by Monte Carlo policy evaluation. ■

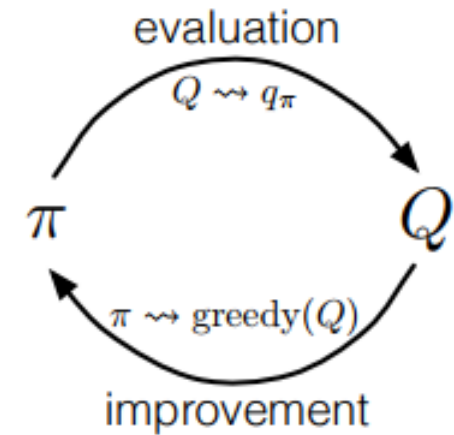
5.2 Monte Carlo Estimation of Action Values

- If a model(of environment) is not available
 - One cannot determine the next state
 - Example) Alpha-go, black jack
 - One may estimate action values rather than state values

- Estimating action values $q_{\pi}(s, a)$
 - All state-action pairs will be visited an infinite number of times in the limit of an infinite number of episodes
- Exploring starts
 - Every state-action pair has a nonzero probability of being selected as the start
 - All state-action pairs will be visited an infinite number of times

5.3 Monte Carlo Control

- Update action values through episodes
- Update policies by action values
$$\pi(s) = \operatorname{argmax}_a q(s, a)$$



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 π : $S_0, A_0, R_1, \dots, S_{T-1}, A_{T-1}, R_T$

$G \leftarrow 0$

Loop for each step of episode, $t = T-1, T-2, \dots, 0$:

$G \leftarrow \gamma G + R_{t+1}$

Unless the pair S_t, A_t appears in $S_0, A_0, S_1, A_1, \dots, S_{t-1}, A_{t-1}$:

Append G to $Returns(S_t, A_t)$

$Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$

$\pi(S_t) \leftarrow \operatorname{argmax}_a Q(S_t, a)$

Example 5.3: Solving Blackjack

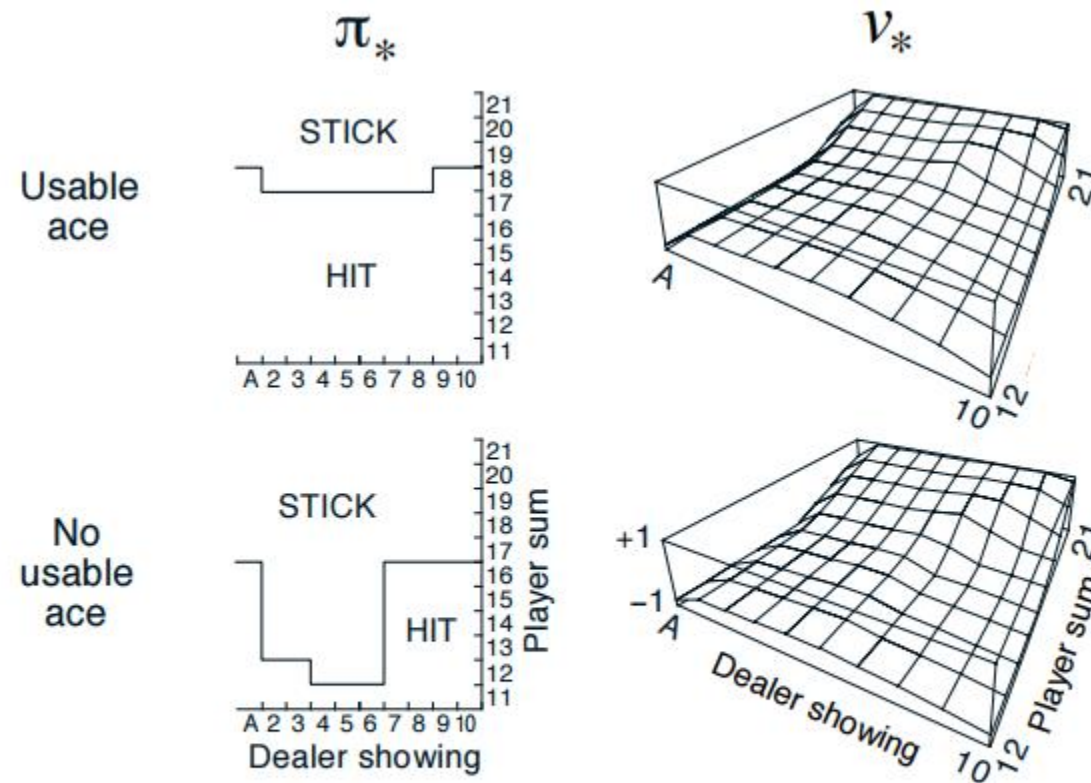
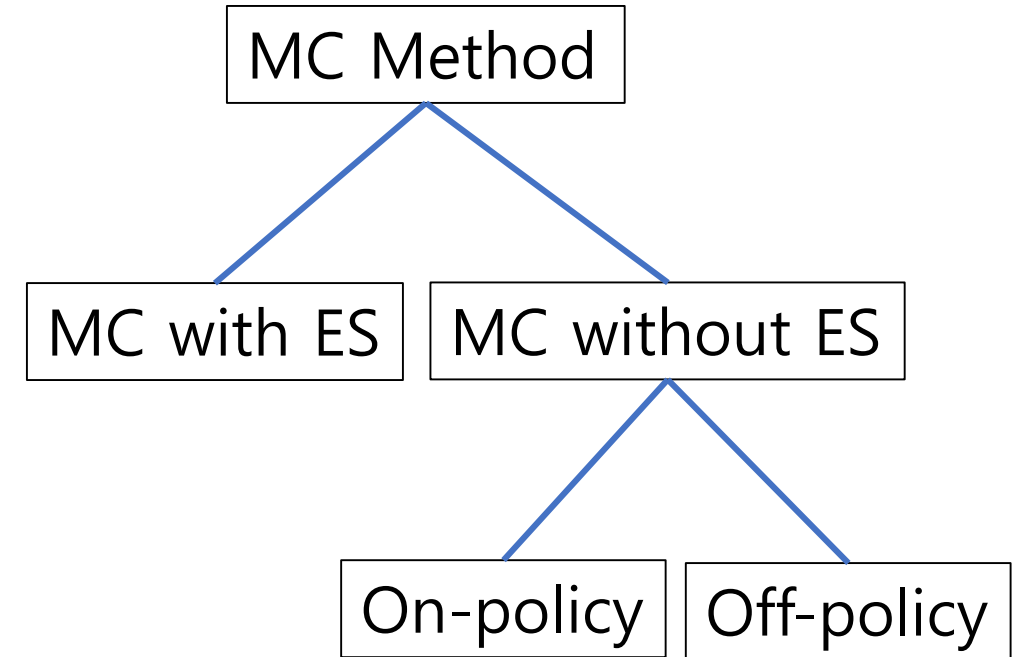


Figure 5.2: The optimal policy and state-value function for blackjack, found by Monte Carlo ES. The state-value function shown was computed from the action-value function found by Monte Carlo ES. ■

5.4 Monte Carlo Control without Exploring Starts

- On-policy methods
 - To improve a policy, action values are evaluated using the policy
- Off-policy methods
 - To improve a policy, action values are evaluated using the other policy



- On-policy method
 - The policy is soft, in general
 - Meaning that $\pi(a|s) > 0$ for all $s \in \mathcal{S}$ and all $a \in \mathcal{A}(s)$
- Example) ϵ -greedy method
 - Most of the time we choose an action that has maximal estimated action value as

$$\Pr(a) = \begin{cases} \frac{\epsilon}{|\mathcal{A}(s)|}, & \text{if } a \text{ is not maximal} \\ 1 - \epsilon + \frac{\epsilon}{|\mathcal{A}(s)|}, & \text{otherwise} \end{cases}$$

for some $\epsilon > 0$

On-policy first-visit MC control (for ε -soft policies), estimates $\pi \approx \pi_*$

Algorithm parameter: small $\varepsilon > 0$

Initialize:

$\pi \leftarrow$ an arbitrary ε -soft policy

$Q(s, a) \in \mathbb{R}$ (arbitrarily), for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

$Returns(s, a) \leftarrow$ empty list, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

Repeat forever (for each episode):

Generate an episode following π : $S_0, A_0, R_1, \dots, S_{T-1}, A_{T-1}, R_T$

$G \leftarrow 0$

Loop for each step of episode, $t = T-1, T-2, \dots, 0$:

$G \leftarrow \gamma G + R_{t+1}$

Unless the pair S_t, A_t appears in $S_0, A_0, S_1, A_1, \dots, S_{t-1}, A_{t-1}$:

Append G to $Returns(S_t, A_t)$

$Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$

$A^* \leftarrow \operatorname{argmax}_a Q(S_t, a)$ (with ties broken arbitrarily)

For all $a \in \mathcal{A}(S_t)$:

$$\pi(a|S_t) \leftarrow \begin{cases} 1 - \varepsilon + \varepsilon/|\mathcal{A}(S_t)| & \text{if } a = A^* \\ \varepsilon/|\mathcal{A}(S_t)| & \text{if } a \neq A^* \end{cases}$$

5.5 Off-policy Prediction via Importance Sampling

- Off-policy learning
 - Example
 - To use two policies, the **target policy** and the **behavior policy**
 - The target policy is learned about and becomes the optimal policy
 - The behavior policy is more exploratory and is used to generate behavior