5. Monte Carlo Methods

Notes

- Here, we do not assume complete knowledge of the environment . Monte Carlo methods require only *experience* (can be real or simulated). It is unlike DP where we had the transition probabilities of all the states.
- Monte Carlo method are based on averaging sample returns, and in this chapter, we shall only consider episodic tasks. The value estimates and policies are only updated after each episode.
- This is different from the bandit problem from Chapter 2 since this problem includes multiple states. From the perspective of the earlier state, the problem becomes non-stationary as the returns from another state are likely to be different. To handle this non-stationarity, we use the idea of GPI, but we learn, instead of compute, the value functions from the sample returns. However, we can still apply the idea of policy evaluation and policy improvement.
- Monte Carlo Prediction:
  - To estimate $v_\pi(s)$, we need to build up more observed returns, such that the average across all of these returns converge to the expected value.
  - Each occurrence of s in an episode is called a *visit* to s, and there can be multiple visits to s in a single episode.
  - The *first-visit MC method* estimates $v_\pi(s)$ as the average of returns following first visits to s, and the *every-visit MC method* does it with all visits to s. Every visit MC extends more naturally to function approximation and eligibility traces (Chapters 9 and 12).
  - Both methods converge to the expected state value as the number of visits to $s \to \infty$, with the every-visit MC converging quadratically. Easy to see for first-visit MC because the return is an independent, identically distributed estimate of $v_\pi(s)$ with finite variance. The average of the state value will be is unbiased, and standard deviation of its error falls as $1/\sqrt{n}$.
  - In a backup diagram for MC methods, the diagram will only show a single route of the sampled states and actions, unlike a DP backup diagram which shows all possible transitions. This reflects the fundamental difference between the two algorithms.
  - MC methods do not *bootstrap*, which means that the estimates for each states are independent. This also means that the computational requirement of estimating a

single state is independent of the number of states. This makes MC methods attractive when one requires only the estimates of one or a subset of states.

- Monte Carlo Estimation for Action Values:
  - Estimating the action value becomes useful when there is no model of the environment. The MC methods here for estimating $q_*(s, a)$ are essentially the same as for state values, but it looks at state-action pairs instead of only states.
  - With a deterministic policy π, only one action from each state is visited, which means that many other state-action pairs are not visited. This does not allow the MC method to learn. Thus, one can introduce the notion of exploration, where all state-action pairs have a non-zero probability of being selected. However, this cannot be relied upon in general. The most common alternative approach is to use stochastic policies $\pi(a|s)$.
- Monte Carlo Control:
  - With the assumption that episodes are guaranteed exploring starts and there are infinite number of episodes, we can do a Monte Carlo version of the classical policy iteration, where there are alternating steps of policy evaluation and policy improvement. The MC method here will be able to compute each q_{π_k} exactly.
  - Here, policy improvement is done by making the policy greedy (hence deterministic) wrt action value function, where $\pi(s) \doteq \arg\max_a q(s, a)$. This also means that

$$\begin{aligned} q_{\pi_k}(s, \pi_{k+1}(s)) &= q_{\pi_k}(s, \arg\max_a q_{\pi_k}(s, a)) \\ &= \max_a q\pi_k(s, \pi_k(s)) \\ &\geq q_{\pi_k}(s, \pi_k(s)) \end{aligned}$$

    which guarantees convergence to an optimal policy and value function.
  - Tackling the assumption of infinite episode:
    - Can just approximate q_π in each policy evaluation. Just like in the DP methods, we can just take sufficient steps such that the probability of error in the estimates are sufficiently small. However, this requires far too many episodes to be useful in practice, especially for large datasets.
    - For MC policy iteration, it is natural to alternate between evaluation and improvement on an episode-by-episode basis. After every episode, policy evaluation and improvement are performed. This algorithm is known as *Monte Carlo ES (Exploring Starts)*. This algorithm converges to an optimal policy and value function, but this has not been proved. (Page 99)

- Monte Carlos Control without Exploring Starts:
  - There are two approaches to this - *on-policy* and *off-policy* methods. On-policy methods attempt to evaluate or improve a policy that is used to make decisions, whereas off-policy methods evaluate or improve a policy different from the one used to generate data.
  - On-policy control methods have *soft* policies, where $\pi(a|s) > 0,\ \forall s \in \mathcal{S},\ \forall a \in \mathcal{A}(s)$, but also shift towards a deterministic optimal policy. Method discussed here uses ϵ-greedy policies, where all non-greedy actions are given a $\epsilon/|\mathcal{A}(s)|$ probability of selection, and the greedy action has probability of $1 - \epsilon + \epsilon/|\mathcal{A}(s)|$. ϵ-greedy policies are an example of ϵ-soft policies.
  - The idea of on-policy MC control is still that of GPI. This means that for any ϵ-soft policy π, any ϵ-greedy policy with respect to q_π is guaranteed to be better than or equal to π. (page 101)
  - In a new environment where the ϵ-soft mechanism is "moved inside" the environment, the best policy and value functions is the same as in the original environment.
  - Proof that GPI works for ϵ-soft policies can be found on page 102. With ϵ-soft policies, the assumption of exploring starts can be eliminated. Note that the policy here is only near-optimal as it has an exploratory component to it.
- Off-policy prediction via Importance Sampling:
  - Off-policy learning involves two policies - 1) the *target policy* π which is learned and becomes the optimal policy, and 2) the *behaviour policy* b which is more exploratory and is used to generate behaviour.
  - Off-policy methods often have greater variance and are slower to converge due to the data being generated by a different policy. However, they are more powerful and general (on-policy methods are a subset of off-policy methods, where the target and behaviour policies are the same).
  - Let us consider the case where the target and behaviour policy are fixed and given, and we want to estimate v_π or q_π from the episodes following the behaviour policy $b,\ b \neq \pi$.
    - We hold the assumption of coverage, which means that we require that $\pi(a|s) > 0$ implies that $b(a|s) > 0$ (every action taken under π is also taken by b). In general, the target policies are usually deterministic, whilst the behaviour policies are stochastic.
    - Almost all off-policy methods uses *importance sampling*, a technique which estimates expected values under one distribution given samples from another.

Given a starting state S_t, the probability of subsequent state-action trajectory under a policy π is

$$\begin{aligned}&\Pr\{A_t, S_{t+1}, A_{t+1}, \dots, S_T | S_t, A_{t:T-1} \sim \pi\} \\ &\quad = \pi(A_t|S_t)p(S_{t+1}|S_t, A_t)\pi(A_{t+1}|S_{t+1}) \dots p(S_T|S_{T-1}, A_{T-1}) \\ &\quad = \prod_{k=t}^{T-1} \pi(A_k|S_k)p(S_{k+1}|S_k, A_k).\end{aligned}$$

The importance-sampling ratio is

$$\rho_{t:T-1} \doteq \frac{\prod_{k=t}^{T-1} \pi(A_k|S_k)p(S_{k+1}|S_k, A_k)}{\prod_{k=t}^{T-1} b(A_k|S_k)p(S_{k+1}|S_k, A_k)} = \frac{\prod_{k=t}^{T-1} \pi(A_k|S_k)}{\prod_{k=t}^{T-1} b(A_k|S_k)},$$

which does not depend on the MDP's trajectory probabilities, and only depends on the policies and the subsequent state-action trajectory.

- We want to estimate the expected returns under the target policy π, but only have the expected returns G_t from the behaviour policy b, which only gives $v_b(s) = \mathbb{E}[G_t|S_t = s]$, not $v_\pi(s)$. However, using $\rho_{t:T-1}$, we can transform the returns to get

$$v_\pi(s) = \mathbb{E}[\rho_{t:T-1}G_t|S_t = s]$$

- Here, it is convenient to number time steps in a way that increases across episode boundaries, such that time-step numbers can be used to refer to a particular step in a particular episode. We can use $\mathcal{J}(s)$ to denote the timesteps in which state s is visited for an every-visit method, and the timesteps for the first visits to s within an episode.
- $\{G_t\}_{t\in\mathcal{J}(s)}$ are the returns from the behaviour policy that pertain to state s and $\{\rho_{t:T(t)-1}\}_{t\in\mathcal{J}(s)}$ are the corresponding importance-sampling ratios. The estimate of $v_\pi(s)$ can be given by

$$V(s) \doteq \frac{\sum_{t\in\mathcal{J}(s)} \rho_{t:T(t)-1}G_t}{|\mathcal{J}(s)|} \quad \text{OR} \quad V(s) \doteq \frac{\sum_{t\in\mathcal{J}(s)} \rho_{t:T(t)-1}G_t}{\sum_{t\in\mathcal{J}(s)} \rho_{t:T(t)-1}}$$

which are the *ordinary* and the *weighted* importance sampling.

- For single-visit methods, the ordinary importance sampling is unbiased but has unbounded variance, while the weighted importance sampling is biased but has bounded variance (the variance converges to zero even if the variance of the

ratios is infinite). In practice, weighted importance sampling has dramatically lower variance and is strongly preferred.

    - For every-visit methods, both methods are biased, but it falls asymptotically to zero as the number of samples increases. In practice, every-visit methods are preferred because they are easier to implement and easier to extend to approximations.

- Incremental Implementation:
    - For on-policy methods, we can essentially use the same update methods used in Chapter 2 (bandit problem), but here we average the returns.
    - For off-policy methods, we need to separately consider the ordinary and weighted importance sampling methods. In ordinary importance sampling, we can just use the same incremental update method used in Chapter 2, but using a scaled return. In the weighted importance sampling method, we can update V_n:

$$V_n \doteq \frac{\sum_{k=1}^{n-1} W_k G_k}{\sum_{k=1}^{n-1} W_k}, \quad n \geq 2$$
$$V_{n+1} = V_n + \frac{W_n}{C_n}[G_n - V_n] \text{ where } V_1 \text{ is arbitrary},$$
$$C_{n+1} \doteq C_n + W_{n+1} \text{ where } C_0 = 0$$

    - The off-policy MC prediction implementation can be found in page 110.
- Off-policy Monte Carlo Control:
    - This method follows the behaviour policy in the environment, whilst learning about and improving the target policy. This requires the behaviour policy to have a non-zero probability of selecting all actions that might be selected by the target policy. For a behaviour policy to explore all possibilities, the policy has to be *soft*.
    - With an ϵ-soft behaviour policy, which can be change between or even within episodes, the policy π converges to optimal at all states as the number of returns obtained for each state-action pair increases.
    - The implementation algorithm can be found in page 111. Note that the method learns only from the tail of episodes. If non-greedy actions are common, then the learning rate will be slow.
- Variance in Importance Sampling:
    - Taking into account the internal structures of the returns, eg. in terms of discounted rewards. can help reduce the variance of the off-policy estimators.
    - DIscounting-aware Importance Sampling:

- Let us take an example of discounting where $\gamma = 0$ in a 100-step episode. This would mean that the return from time 0 will be $G_0 = R_1$, but its importance sampling will still be a product of 100 factors: $\frac{\pi(A_0|S_0)}{b(A_0|S_0)} \frac{\pi(A_1|S_1)}{b(A_1|S_1)} \cdots \frac{\pi(A_{99}|S_{99})}{b(A_{99}|S_{99})}$. However, it is only necessary to scale it by the first factor, and the other 99 are irrelevant. The other factors are independent of the return and have expected value of 1. They do not change the update, but they can add enormously to the variance.
- We can think of discounting as the probability of termination, or also known as the degree of partial termination. For example, we can think of G_0 partially terminating in one step at R_1 with a degree $1 - \gamma$, or partially terminating in two steps at $R_1 + R_2$ with a degree of $(1 - \gamma)\gamma$, and so on. The partial returns are called *flat partial returns*, where there is not discounting and it does not extend all the way to termination:

  $$\bar{G}_{t:h} \doteq R_{t+1} + R_{t+2} + \cdots + R_h, \quad 0 \leq t < h \leq T,$$

  The full return can then be written as: (page 113)

  $$G_t \doteq (1 - \gamma) \sum_{h=t+1}^{T-1} \gamma^{h-t-1} \bar{G}_{t:h} + \gamma^{T-t-1} \bar{G}_{t:T}.$$

  The ordinary importance-sampling estimator can the be written as a *discounting-aware* importance sampling estimator:

  $$V(s) \doteq \frac{\sum_{t \in \mathcal{J}(s)} \left((1-\gamma) \sum_{h=t+1}^{T(t)-1} \gamma^{h-t-1} \rho_{t:h-1} \bar{G}_{t:h} + \gamma^{T(t)-t-1} \rho_{t:T(t)-1} \bar{G}_{t:T(t)} \right)}{|\mathcal{J}(s)|}$$

- Per-decision Importance Sampling
  - In a scenario where there is not discounting, we can use an estimator called *per-decision importance sampling*

    $$\tilde{G}_t = \rho_{t:t} R_{t+1} + \gamma \rho_{t:t+1} R_{t+2} + \cdots + \gamma^{T-t-1} \rho_{t:T-1} R_T$$

    which gives an alternate importance-sampling estimator as well.
- Advantages of MC methods:
  - Can be used to learn optimal behaviour directly from interaction with the environment, with no model of the dynamics of the environment.
  - Can be used with simulations or sample models.

- It is easy and efficient to focus on a small subset of states
- It is less harmed by violations of the Markov property because it does not bootstrap (estimate based off estimates of successor states) (discussed later)

Exercises

5.1 1) There is jump in the last two rows in the rear because a hand of either 20 and 21 has a much higher chance of winning than a hand of lesser value. 2) The row drops of on the left because having an ace means that there is a non-zero probability that the dealer has a natural. 3) With the usable ace case, the ace can be used as an 11 as well, which can increase the chances of the player getting a high value hand. With the no usable ace case, the ace only has a value of 1, which means that the player is required to hit, thus increasing the chance of the player going bust.

5.2 There will be no difference in first-visit and every-visit MC methods because a state can only occur once in a single game of blackjack. Attempting to use an every-visit MC method would just give identical results to the first-visit MC method.

5.3 The backup diagram for MC estimation of q_π is: [state-action pair value] →reward→ [state value] →reward→ [state-action pair value] →reward→ ... [end]

5.4 The pseudo-code can be changed and written as:

Initialise:

...

$Returns(s,a) = 0,\ n(s,a) = 0 \text{ for all } s \in \mathcal{S},\ a \in \mathcal{A}(s)$

Loop Forever (for each episode):

...

Loop ...

Unless ...

$n(s,a) \leftarrow n(s,a) + 1$

$Returns(s,a) \leftarrow Returns(s,a) + \frac{1}{n(s,a)}(G - Returns(s,a))$

...

5.5 Let $\rho_{t:T(t)-1} = 1\ \forall t$. For the first-visit method, $|\mathcal{J}(s)| = 1$, which means $V(s) = 10/1 = 10$. For the every-visit method, $|\mathcal{J}(s)| = 10$, which means that $V(s) = (10 + 9 + 8 + \cdots + 1)/10 = 5.5$.

5.6 Estimate of action values $Q(s,a)$ using the weighted importance-sampling method:

$$Q(s,a) = \frac{\sum_{t \in \mathcal{J}(s,a)} \rho_{t+1:T(t)-1} G_t}{\sum_{t \in \mathcal{J}(s,a)} \rho_{t+1:T(t)-1}}$$

5.7 We know that the initial estimates of the value function is biased towards the behaviour policy, thus in the first few episodes, the estimate is a reasonable estimate. However, after a few more episodes, the estimator will try to correct itself and decrease its bias, which can result in an increase in error. However, we know that the bias of this method converges asymptotically to zero as the number of episodes increase, thus resulting in a lower error.

5.8 In an episode with n timesteps, all of the actions, except for the last, in these timesteps take the *left* action. Thus, the expected square of the importance-sampling-scaled return would involve a sum over all of the n timesteps as well. The expected return would look like:

$$\begin{aligned}
&\mathbb{E}_b\left[\left(\frac{1}{T-1}\sum_{k=0}^{T-1}\prod_{t=0}^{k}\frac{\pi(A_t|S_t)}{b(A_t|S_t)}G_0\right)^2\right] \\
&= \frac{1}{2}\cdot 0.1\left(\frac{1}{0.5}\right)^2 \\
&\quad + \frac{1}{2}\left[\frac{1}{2}\cdot 0.9\cdot\frac{1}{2}\cdot 0.1\left(\frac{1}{0.5}\frac{1}{0.5}\right)^2 + \frac{1}{2}\cdot 0.1\left(\frac{1}{0.5}\right)^2\right] \\
&\quad + \dots \\
&= 0.1\sum_{k=1}^{\infty}\sum_{m=0}^{k-1}\frac{1}{k}\cdot 0.9^m\cdot 2^m\cdot 2 = \infty
\end{aligned}$$

5.9 The first-visit MC prediction algorithm can be amended to give:

Input: ...

Initialize:

...

$Returns(s) \leftarrow 0, \text{ for all } s \in \mathcal{S}$

$n(s) \leftarrow 0, \text{ for all } s \in \mathcal{S}$

Loop forever (for each episode):

...

Loop ... :

...

Unless ... :

$Returns(S_t) \leftarrow Returns(S_t) + \frac{1}{n(S_t)}[Returns(S_t) - G]$

$v(S_t) \leftarrow Returns(S_t)$

5.10

$$
\begin{aligned}
V_{n+1} &\doteq \frac{\sum_{k=1}^{n} W_k G_k}{\sum_{k=1}^{n} W_k} \\
&= \frac{W_n G_n + \sum_{k=1}^{n-1} W_k G_k}{\sum_{k=1}^{n-1} W_k} \frac{\sum_{k=1}^{n-1} W_k}{\sum_{k=1}^{n} W_k} \\
&= \left(\frac{W_n G_n}{C_{n-1}} + V_n \right) \frac{C_{n-1}}{C_n} \\
&= \frac{W_n G_n}{C_n} + V_n \frac{C_n - W_n}{C_n} \\
&= V_n + \frac{W_n}{C_n}[G_n - V_n]
\end{aligned}
$$

5.11 Since the target policy $\pi(s)$ is deterministic, $\pi(A_t|S_t) = 1$ for all action-state pairs.

5.12 With the noise turned on, the off-policy MC method took a larger number of episodes to form a 'smoother' optimal policy. Without noise, 1000 episodes was sufficient to reach a good optimal policy, whilst with noise, 10000 was required to reach a good optimal policy.

5.13

$$\mathbb{E}[\rho_{t:T-1}R_{t+1}] = \mathbb{E}\left[\frac{\pi(A_t|S_t)}{b(A_t|S_t)}\frac{\pi(A_{t+1}|S_{t+1})}{b(A_{t+1}|S_{t+1})}]\cdots\frac{\pi(A_{T-1}|S_{T-1})}{b(A_{T-1}|S_{T-1})}R_{t+1}\right]$$

$$\text{know that } \mathbb{E}\left[\frac{\pi(A_k|S_k)}{b(A_k|S_k)}\right] = \sum_a \pi(A|S_k) = 1$$

$$\mathbb{E}[\rho_{t:T-1}R_{t+1}] = \mathbb{E}\left[\frac{\pi(A_t|S_t)}{b(A_t|S_t)}R_{t+1}\right] = \mathbb{E}[\rho_{t:t}R_{t+1}]$$