CS5180 Reinforcement Learning

Exercise 5: Off-Policy Monte-Carlo & Temporal-Difference Learning

Anway Shirgaonkar

Q1 Off-policy methods

(a) Derivation of the weighted-average update rule.

From equation 5.7, we have:

$$V_n \doteq \frac{\sum_{k=1}^{n-1} W_k G_k}{\sum_{k=1}^{n-1} W_k}$$

Now,

$$V_{n+1} \doteq \frac{\sum_{k=1}^{n} W_k G_k}{\sum_{k=1}^{n} W_k}$$

We have, $C_n = \sum_{k=1}^{n} W_k$

$$V_{n+1} \doteq \frac{\sum_{k=1}^{n-1} W_k G_k}{C_n} + \frac{W_n G_n}{C_n}$$

Multiplying and dividing by C_{n-1} for one of the terms,

$$V_{n+1} \doteq \frac{C_{n-1}}{C_{n-1}} * \frac{\sum_{k=1}^{n-1} W_k G_k}{C_n} + \frac{W_n G_n}{C_n}$$

$$V_{n+1} \doteq \frac{C_{n-1} V_n}{C_n} + \frac{W_n G_n}{C_n}$$

$$V_{n+1} \doteq \frac{V_n (C_n - W_n)}{C_n} + \frac{W_n G_n}{C_n}$$

$$V_{n+1} \doteq \frac{V_n C_n - V_n W_n}{C_n} + \frac{W_n G_n}{C_n}$$

$$V_{n+1} \doteq V_n - \frac{V_n W_n}{C_n} + \frac{W_n G_n}{C_n}$$

$$\boxed{V_{n+1} \doteq V_n + \frac{W_n}{C_n}(G_n - V_n)}$$

(b) If the target policy is a greedy policy, then $\pi(A_t|S_t) = 1$. It will be zero for the non-greedy actions in a state, which will drive the importance sampling ratio to 0 as well, in which case evaluating action values further makes no sense. Hence, if we always evaluate the action values when $\pi(A_t|S_t) = 1$.

Therefore, $\frac{\pi(A_t|S_t)}{b(A_t|S_t)} = \frac{1}{b(A_t|S_t)}$

Q2 Temporal difference vs. Monte-Carlo

(a) I believe that TD method would be better since the update rule depends on the value of the current and the immediate next states. If we see the Monte-Carlo method, we'll have to wait for an episode to end to update the values.

So basically, consider the case when we start driving home from work and estimate the time to reach as 50 minutes. Now, if we get stuck in a traffic (new state) we would immediately like to update the value of our state, i.e. predict that the time to reach will increase. This is only possible when using TD learning. When using MC method, we'll have wait until the end of episode to update our value estimates.

(b) Yes, I think Monte Carlo would be better off in case of episodic tasks. Especially in a case where the episodes are especially short, in case of a blackjack game, using MC is preferable since we get the reward (either win or lose) only at the end of the episode. It will not make sense to use TD here, since we don't get rewards at every step.