Monte Carlo Methods

Monte Carlo Prediction:-

The Monte Carlo prediction method is used to approximate the action-value function. To do so, the agent follows a policy (random policy) & generate a set of episodes with state-action-reward sequences.

Using these episodes, we learn a table known a Q-table that is of the size $|S| \times |A|$ where each cell corresponds to a (state, action) pair.

Eg. episode 1:- $S_1\ a_2\ r_1\ S_2\ a_1\ r_2\ S_3\ a_2\ r_3\ S_1\ a_1\ r_2 \longrightarrow$

⋮

so on, to fill the table.

	a_1	a_2	a_3
S_1	r_2	r_1	
S_2	r_2		
S_3		r_3	

✱ Note:- The MC prediction can also be performed to get "state-value" function instead of "action-value" function.

Q. What if same state-action pair is seen in an episode again?

A. episode 1:- $S_1\ a_2\ r_1\ S_2\ a_1\ r_2\ S_3\ a_2\ r_3\ S_1\ a_2\ r_2$

episode 2:- $S_1\ a_2\ r_3\ S_2\ a_2\ r_1\ S_3\ a_1\ r_1$

This can happen if there is some randomness in the environment resulting different reward for same (s, a).

There are 2 methods to deal with this situation:-

(i) First-Visit MC:- In this the Q-table is updated when the (s,a) pair is first seen in the episode, later visits are ignored.

First-Visit

	a_1	a_2	a_3
S_1		X	
S_2	r_2	r_1	
S_3	r_1	r_3	

$X = \frac{(r_1 + r_3)}{2}$

Every-Visit

	a_1	a_2	a_3
S_1		Z	
S_2	r_2	r_1	
S_3	r_1	r_3	

$Z = \frac{(r_1 + r_2 + r_3)}{3}$

(ii) Every-Visit MC:- In this the Q-table is updated with the average reward seen for the episode on that (s,a) pair visits.

✱ Note:- Given sufficient number of episodes, both the methods converge to give π_*

› Every-visit MC is biased.

› Initially Every-visit MC has lower Mean Squared Error, but as more episodes are collected, First-visit MC attains better MSE.

"If" condition can be removed below to get Every-Visit MC

Algorithm 8: First-Visit MC Prediction *(for state values)*

Input: policy π, positive integer *num_episodes*
Output: value function V ($\approx v_\pi$ if *num_episodes* is large enough)
Initialize $N(s) = 0$ for all $s \in \mathcal{S}$
Initialize $returns\_sum(s) = 0$ for all $s \in \mathcal{S}$
for $i \leftarrow 1$ ***to*** *num_episodes* **do**
  Generate an episode $S_0, A_0, R_1, \ldots, S_T$ using π
  for $t \leftarrow 0$ ***to*** $T-1$ **do**
    if S_t *is a first visit (with return* G_t*)* **then**
      $N(S_t) \leftarrow N(S_t) + 1$
      $returns\_sum(S_t) \leftarrow returns\_sum(S_t) + G_t$
  end
end
$V(s) \leftarrow returns\_sum(s)/N(s)$ for all $s \in \mathcal{S}$
return V

Algorithm 9: First-Visit MC Prediction *(for action values)*

Input: policy π, positive integer *num_episodes*
Output: value function Q ($\approx q_\pi$ if *num_episodes* is large enough)
Initialize $N(s,a) = 0$ for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$
Initialize $returns\_sum(s,a) = 0$ for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$
for $i \leftarrow 1$ ***to*** *num_episodes* **do**
  Generate an episode $S_0, A_0, R_1, \ldots, S_T$ using π
  for $t \leftarrow 0$ ***to*** $T-1$ **do**
    if (S_t, A_t) *is a first visit (with return* G_t*)* **then**
      $N(S_t, A_t) \leftarrow N(S_t, A_t) + 1$
      $returns\_sum(S_t, A_t) \leftarrow returns\_sum(S_t, A_t) + G_t$
  end
end
$Q(s,a) \leftarrow returns\_sum(s,a)/N(s,a)$ for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$
return Q

Epsilon Greedy Policy (ϵ) :-

Instead of always selecting the greedy action, we make a decision based on the probability (ϵ).

The agent randomly selects an action with probability (ϵ) & selects the greedy action with prob. $(1-\epsilon)$ where ϵ lies between $[0, 1]$.

This allows the agent to move greedily but also explore the environment.

$\epsilon \uparrow$ Exploration $\uparrow$ Exploitation $\downarrow$

MC Control Problem :-

Estimating the optimal policy :-

(i) Policy - Evaluation :- Use initial policy π to construct the Q-table

(ii) Policy - Improvement :- Improving the policy by changing it to be ϵ-greedy w.r.t the Q-table i.e. $\pi' \leftarrow \epsilon\text{-greedy}(Q)$

$\pi \leftarrow \pi'$

Repeat (i) & (ii) to get the optimal policy. (π_*)

Exploration vs. Exploitation :-

The value of epsilon governs whether the agent explores / exploits.

$\epsilon = 1 \Rightarrow$ Complete exploration (random)

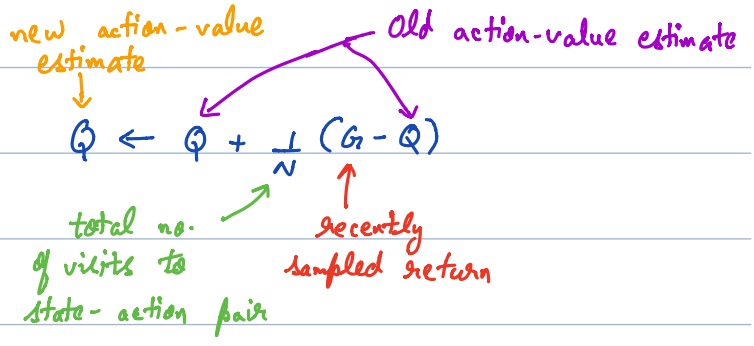
$\epsilon = 0 \Rightarrow$ Complete exploitation (greedy)

As initially the Agent knows very less about the environment, we want Exploration & in a later stage it can exploit on the learned knowledge.

This is achieved by gradually decaying ϵ upto a certain threshold say ($\epsilon = 0.1$)

Incremental Mean :-

Instead of waiting for many episodes to converge our Q-table, we update it after each episode by taking the incremental mean for each (s, a) pair seen in the episode.



Algorithm 10: First-Visit GLIE MC Control

Input: positive integer *num_episodes*, GLIE $\{\epsilon_i\}$
Output: policy π ($\approx \pi_*$ if *num_episodes* is large enough)
Initialize $Q(s,a) = 0$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$
Initialize $N(s,a) = 0$ for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$
for $i \leftarrow 1$ ***to*** *num_episodes* **do**
  $\epsilon \leftarrow \epsilon_i$
  $\pi \leftarrow \epsilon$-greedy(Q)
  Generate an episode $S_0, A_0, R_1, \ldots, S_T$ using π
  for $t \leftarrow 0$ ***to*** $T-1$ **do**
    if (S_t, A_t) *is a first visit (with return* G_t*)* **then**
      $N(S_t, A_t) \leftarrow N(S_t, A_t) + 1$
      $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \frac{1}{N(S_t, A_t)}(G_t - Q(S_t, A_t))$
  end
end
return π

Constant Alpha (α) :-

In the previous algorithm, to update Q_{t+1} we add $\delta_t = (G_t - Q_t)$.

If $\delta_t > 0$ we increase Q_{t+1} value

$\delta_t < 0$ we decrease Q_{t+1} value

The amount of increase / decrease depends on the value of N. As N grows with time, the updates become very small & insignificant.

To handle this instead of $\frac{1}{N}$, we use a constant (α) which allows to update Q-table uniformly overtime.

How to choose value of α?

$Q \leftarrow Q + \alpha(G - Q)$

$Q \leftarrow (1-\alpha)Q + \alpha G$

if $\alpha = 0 \Rightarrow$ Use the old table

$\alpha = 1 \Rightarrow$ Use only the latest return & discard previous knowledge.

Algorithm 11: First-Visit Constant-α (GLIE) MC Control

Input: positive integer *num_episodes*, small positive fraction α, GLIE $\{\epsilon_i\}$
Output: policy π ($\approx \pi_*$ if *num_episodes* is large enough)
Initialize Q arbitrarily (e.g., $Q(s,a) = 0$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$)
for $i \leftarrow 1$ ***to*** *num_episodes* **do**
  $\epsilon \leftarrow \epsilon_i$
  $\pi \leftarrow \epsilon$-greedy(Q)
  Generate an episode $S_0, A_0, R_1, \ldots, S_T$ using π
  for $t \leftarrow 0$ ***to*** $T-1$ **do**
    if (S_t, A_t) *is a first visit (with return* G_t*)* **then**
      $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(G_t - Q(S_t, A_t))$
  end
end
return π

We need to set 'α' such that it learns from new knowledge.

Increasing α ensures the agent focuses on the most recent returns.