CLASSTIME | Page No.
Date / /

RL HW 3

Vishaal Udandarao
20161179.

1) Ex. 5.4

Assume that we track the returns for each state-action pair (s,a) by G_i.

For updating the action value function, we need to average out over G_is.

Considering first-visit MC ⇒

$$Q_{n+1}(s,a) = \frac{1}{n}\left(\sum_{i=1}^{n} G_i\right)$$

where Q_1 is initialized arbitrarily for every state-action pair.

Now, we know ⇒

$$Q_{n+1}(s,a) = \frac{1}{n}\left(\sum_{i=1}^{n-1} G_i + G_n\right)$$

$$= \frac{1}{n}\left(\frac{n-1}{n-1}\sum_{i=1}^{n-1} G_i + G_n\right)$$

$$= \frac{1}{n}\left((n-1)\cdot Q_n(s,a) + G_n\right)$$

CLASSTIME | Page No.
Date / /

$$\Rightarrow \boxed{Q_{n+1}(S,a) = Q_n(S,a) + \frac{1}{n}(G_n - Q_n(S,a))}$$

We clearly see from this modified update rule that the count of number of times the particular state action pair has been encountered and the previous action value function suffice to form the updated action

The updated pseudo code will be ⇒

Initialize:
$\pi(s) \in A(s)$ (arbitrarily) for all $s \in S$.
$Q(s,a) \in R$ (arbitrarily) for all $s \in S, a \in A(s)$
Returns $(s,a) = 0$ for all $s \in S, a \in A(s)$

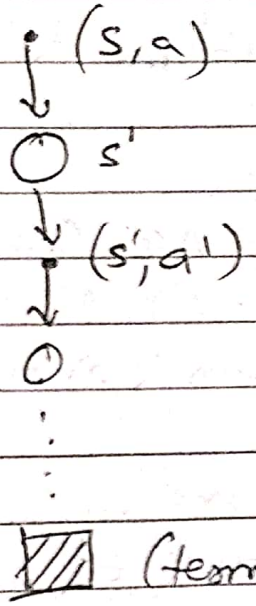
Loop Forever: (for each episode)
Choose $S_0 \in S, A_0 \in A(s_0)$ randomly s.t all pairs have non-0 probability.
Generate an episode from S_0, A_0 following π : $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$.
$G \leftarrow 0$
Loop for each step of episode, $t = T-1, T-2, \ldots 0$:
$G \leftarrow \gamma G + R_{t+1}$
Returns $(s,a) += 1$
Unless pair s,a appears in $S_0, A_0, \ldots, S_{t-1}, A_{t-1}$

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

$\pi(s) \leftarrow \arg\max_a Q(s, a)$

The above pseudo code is the same as the MC control ES pseudocode as we have previously shown that both the update rules are equivalent.

2) The backup diagram for q_π is ⇒



The topmost node shows the initial state-action pair of a particular episode, the rest of the nodes are state nodes and state-action pairs based on the episode.
Black dots depict state-action pair nodes while circles represent state nodes.

CLASSTIME | Page No.
Date / /

3) Assume the set of all time-steps in which the state-action pair (s,a) is visited to be denoted as ⇒ $\mathcal{T}(s,a)$

Then, we can represent $Q(s,a)$ as ⇒

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

where G_t is the return at steps t to $T(t)$.
and $\rho_{t:T(t)-1}$ are the importance sampling ratios.

5) In this particular case TD updates would perform better due to ⇒
i) As TD bootstraps, the original state values are retained.
ii) As only initial route in the problem has changed, there will be many common states in the new problem and old problem.
iii) State-value function of TD for original problem would have converged to optimal values since there was a lot of prior experience while driving home from work.

CLASSTIME | Page No.
Date / /

iv) This state value function can be used in the new problem as they would be very close to the true state values of the new problem. Hence, there will be faster convergence to true state values.

Even, in the original scenario if the initial value estimates are close to true values, then there would be faster convergence.

8) If action-selection procedure is greedy, q-learning is not the same as the sarsa algo. This is because sarsa picks the next action from a given next state according to the q-values. The update to the q-values is then based on this next action.
However, in Q-learning, first the q-values are updated and then the next action is picked according to the updated q-value. This is why the q-learning is different from sarsa.

6) Exercise 6.3

We are given the following hyperparameters ⇒

$\alpha = 0.1$
$\gamma = 1$ (undiscounted)

We know, update to state-value is ⇒

$$V(S_t) = V(S_t) + \alpha(R_{t+1} + \gamma V(S_{t+1}) - V(S_t))$$

Since all rewards are 0 except for transitions to terminal state ⇒

$$V(S_t) = V(S_t) + 0.1(0 + V(S_{t+1}) - V(S_t))$$

Now, all $V(S_t)$ initially were 0.5

$$\Rightarrow V(S_t) = V(S_t) + 0.1(0.5 - 0.5)$$
$$\Rightarrow V(S_t) = V(S_t)$$

For terminal states ⇒

$$\boxed{V(S_t) = 0}$$

In the first episode, when we go to A that is the left terminal state, and reward is 0.

CLASSTIME | Page No.
Date / /

$$\Rightarrow V(A) = V(A) + 0.1(0 + 0 - V(A))$$

$$\Rightarrow V(A) = V(A) - 0.1\,V(A)$$
$$= 0.9\,V(A)$$
$$= 0.9 \times 0.5$$
$$= 0.45$$

Therefore, we conclude that only state value of A changed in first episode i.e it decreased to 0.45.

Therefore the amount of change was 0.9 times the original value = 0.5

Exercise 6.4

Yes, the conclusions about the better performing algo will be affected if we use a wider range of alpha values. Suppose we take large values of alpha, then even TD(0) will start performing poorly since oscillations are likely to happen as the step size α being large causes larger jumps and hence it will be hard for the convergence to be achieved.
Similar is the case with MC algo if we pick a wider set of alpha

Yes, there will be a specific alpha at which the algos might perform better. This can be found out by doing an alpha-tuning to find out the best alpha at which RMSE is reduced.

Exercise 6.5

For high α values, there will be oscillations in the convergence (which can be seen as a gradient descent update rule). Therefore, for high α values, when convergence is achieved, oscillations will occur.

NO, this is not a function of how the value function estimates are initialised. At all high α values, oscillations will occur.