Shane Toma

RBE 595

2/25/2024

# Assignment 4: Temporal Difference Learning

1- Between Dynamic Programming, Monte-Carlo and Temporal Difference, which one of these algorithms use bootstrapping? Explain.

Of the three algorithms we have studied thus far, both Dynamic Programming and Temporal Difference methods use bootstrapping. This is because both these methods base their incremental updates on estimates from the previous states, or update at each time step, whereas Monte-Carlo methods only update at the end of each episode.

2- We mentioned that the target value for TD is [$Rt$+1 + $\gamma$($st$+1)]. What is the target value for Monte-Carlo, Q-learning, SARSA and Expected-SARSA.

Monte-Carlo: $G_t$

Q-Learning: $R_{t+1} + \gamma \max_a Q(S_{t+1}, a)$

SARSA: $R_{t+1} + \gamma Q(S_{t+1}, A_{t+1})$

Expected-SARSA: $R_{t+1} + \gamma \sum_a \pi(a|S_{t+1})\, Q(S_{t+1}, a)$

3- What are the similarities of TD and MC

The primary similarity between the TD and MC methods are their use of sampling in there update process. Monte-Carlo methods sample action values in order to estimate a return value in place of using expected values like in DP. TD uses this sampling method while also using current estimate to combine DP and MC strategies.

4- Assume that we have two states $x$ and $y$ with the current value of V($x$) = 10, Y($y$) = 1. We run an episode of {$x$, 3, $y$, 0, $y$, 5, $T$}. What's the new estimate of V($x$), V($y$) using TD (assume step size $\alpha$ = 0.1 and discount rate $\gamma$ = 0.9)

Using the Tabular TD(0) method, new estimates of V(x) and V(y) were made using the following method:

$$V(S) = V(S) + \alpha[R + \gamma V(S') - V(S)$$

Step 1, state = x, reward = 3:

$$V(x) = 9.3$$

Step 2, state =y, reward =0:

$$V(y) = 0.99$$

Step 3, state = y, reward = 5:

$$V(y) = 1.391$$

Step 4, state = T, and our final estimates are:

$$V(x) = 9.3, V(y) = 1.391$$

5- Can we consider TD an online (real-time) method and MC an offline method? Why?

TD can be considered an online, or real-time method due to the fact that it updates its return function throughout each iteration it makes. Monte-Carlo methods, on the other hand, can be considered offline because they do not return a value function until the end of each episode.

6- Does Q-learning learn the outcome of exploratory actions?

Q-learning does not learn the outcome of exploratory actions, as its algorithm choses the optimal approximation of the action-value function for every iteration. In the context of the cliff example, this means that the Q-learning algorithm will always choose the 'optimal' path closest to the cliff, even though this means a lower average reward as it will occasionally fall off the cliff.

7- What is the advantage of Double Q-learning over Q-learning?

Unlike Q-learning, Double Q-learning uses two approximate value functions which it can chose to update at random. This effectively eliminate the negative effects of maximization bias, which is the overestimation of action values and future rewards.