NAME1: Mahsa Mojtahedi
STUDENTID1:12166510
MAIL1:

NAME2: Melika Ayoughi
STUDENTID2:12224855
MAIL2: melika.ayoughi@uva.student.nl

Homework Assignment 1
Reinforcement Learning, 19/20

2019-09-20

1 2.2 Homework: Dynamic Programming

1.1 Write the value, $v^{\pi}(s)$, of a state s under policy π, in terms of π and $q^{\pi}(s, a)$. Write down both the stochastic and the deterministic policy case.

In general case we have:

$$v_{\pi}(s) = \mathbb{E}_{\pi}[G_t|S_t = s] = \mathbb{E}_{a\sim\pi}q_{\pi}(s, a)$$

If we have a deterministic policy, $v_{\pi}(s)$ would be equal to:

$$v_{\pi}(s) = q_{\pi}(s, a)$$

If the policy is stochastic, we need to take all possible actions into account, thus:

$$v_{\pi}(s) = \sum_{a} \pi(a|s)q_{\pi}(s, a)$$

1.2 In Policy Iteration, we first evaluate the policy by computing its value function, and then update it using a Policy Improvement step. You will now change Policy Iteration as given on page 80 of the book to compute action-values. First give the new policy evaluation update in terms of $Q^{\pi}(s, a)$ instead of $v^{\pi}(s)$. Note that Policy Improvement uses deterministic policies.

```
repeat
$\Delta \leftarrow 0$
    for each $s \in S$: do
        for each $a \in A(s)$: do
            $q \leftarrow Q(s, a)$
            $Q(s, a) = \sum_{s',r} p(s', r|s, a)[r + \gamma \sum_{a'} \pi(a'|s')Q(s', a')]$
            $\Delta \leftarrow \max(\Delta, |q - Q(s, a)|)$
        end for
    end for
until $\Delta < \epsilon$
```

1.3 Now change the Policy Improvement update in terms of $Q^{\pi}(s, a)$ instead of $v^{\pi}(s)$.

```
$policystable \leftarrow TRUE$
for each $s \in S$: do
  $oldaction \leftarrow \pi(s)$
  $\pi(s) = argmax_a Q(s, a)$
    if $oldaction \neq \pi(s)$ then
        $policystable \leftarrow FALSE$
    end if
end for
if policystable then
  and return $V \approx v^*$ and $\pi \approx \pi^*$
else
      go to evaluation
end if
```

1.4 The Value Iteration update, given in the book in Eq. 4.10 can also be rewritten in terms of Q-values. Give the Q-value Iteration update.

$$Q_{k+1}(s,a) = \sum_{s',r} p(s',r|s,a)[r + \gamma \max_{a'} Q_k(s',a')]$$

2 3.1 Homework: Monte Carlo

2.1 Consider an MDP with a single state s0 that has a certain probability of transitioning back onto itself with a reward of 0, and will otherwise terminate with a reward of 5. Your agent has interacted with the environment and has gotten the following three trajectories: [0; 0; 5]; [0; 0; 0; 0; 5]; [0; 0; 0; 5].Use $\gamma = 0.9$

- **Estimate the value of s_0 using first-visit MC.**
  $v(s_0) = 5 * \frac{1}{3}(\gamma^2 + \gamma^4 + \gamma^3) = 3.6585$
- **Estimate the value of s_0 using every-visit MC.**
  $v(s_0) = 5 * \frac{1}{12}(\gamma^2 + \gamma + 1 + \gamma^4 + \gamma^3 + \gamma^2 + \gamma + 1 + \gamma^3 + \gamma^2 + \gamma + 1) = 4.268$

2.2 What is a disadvantage of using ordinary importance sampling in off-policy Monte Carlo?

In ordinary off-policy Monte Carlo the estimates are either very high or very low and the variance is high but unbiased, which makes the convergence of value estimation slower. (expected value equals to true value function)
In the case of having only 1 trajectory, the estimated value is equal to V_π but it can be extreme. If the ratio is very high (or very low), the observed trajectory is much more likely to happen under the target policy rather than the behaviour policy. Therefore, the ordinary importance sampling estimate would predict a much higher observed return than the actual value, thus the high variance.

2.3 What is a disadvantage of using weighted importance sampling in off-policy Monte Carlo?

Contrary to ordinary importance sampling, the weighted version is biased but the variance is low, thus lower errors.
In case of having only 1 trajectory (data-point), for the weighted average estimate, the ratio $\rho_{t:T(t)-1}$ cancels in the nominator and denominator so that the estimate is equal to the observed return independent of this ratio (non-zero ratio). The expectation of this value is V of the **observed** policy but we are looking for the V of the **target** policy, thus this sampling is biased. The bias goes to 0 as the number of points in the trajectory increases.

3 4.4 Homework: Maximization Bias

3.1 We repeatedly apply Q-learning and SARSA on the observed data until convergence. Give all final state-action values for Q-learning and SARSA respectively.

SARSA:

$$\begin{aligned} Q(A, left) &= 2(greedy/1random) \\ Q(A, right) &= 1.5 \\ Q(B, a1) &= 1 \\ Q(B, a2) &= 1 \\ Q(B, a3) &= 2 \\ Q(B, a4) &= 0 \end{aligned}$$

Q-learning:

$$\begin{aligned} Q(A, left) &= 2 \\ Q(A, right) &= 1.5 \\ Q(B, a1) &= 1 \\ Q(B, a2) &= 1 \\ Q(B, a3) &= 2 \\ Q(B, a4) &= 0 \end{aligned}$$

3.2 This problem suffers from maximization bias. Explain where this can be observed. Do both Q-learning and SARSA suffer from this bias? Why/why not?

Yes, both Q-learning and SARSA suffer from the maximization bias, which for both cases comes from choosing the maximum function. For greedy SARSA this comes from the greedy policy and for Q-learning, it involves a max over all the actions in the next state; Thus, it overestimates value and action-value functions. In this example, it overestimates the $Q(A, left)$ value.

3.3 To circumvent the issue of maximization bias, we can apply Double Q-learning. Use the given example to explain how Double Q-learning alleviates the problem of maximization bias.

The problem with Q-Learning is that the same samples are being used to decide which action is the best (highest expected reward), and the same samples are also being used to estimate that action-value. This way we are introducing the bias to maximum value in this example. To solve this problem, double Q-learning is introduced. In this approach, we can have two different action-value estimates and the best action can be chosen based on the values of the first action-value estimate and the target can be decided by the second action-value estimate.

3.4 What are the true state-action values that we would expect to get (after convergence) if we continued sampling episodes.

$$\begin{aligned} Q(A, left) &= 1 \\ Q(A, right) &= 1.5 \\ Q(B, a1) &= 1 \\ Q(B, a2) &= 1 \\ Q(B, a3) &= 1 \\ Q(B, a4) &= 1 \end{aligned}$$