Reinforcements Learning Assignment 3

Reinforcement Learning, H.W. -3

(Q.1) By the definition of the Q-function,

$\forall s \in S$, $\pi^*_{M'} \in \arg\max_{a \in A} \{ Q^*_{M'}(s,a) \} =$

$= \arg\max_{a \in A} \{ Q^*_{M}(s,a) + f(s) \} =$

~~$\arg\max_{a \in A} Q^*_M(s,a)$~~

$= \arg\max_{a \in A} \{ Q^*_M(s,a) \}$

$\Rightarrow$ And since $\forall s \in S$, $\pi^* \in \arg\max_{a \in A} \{ Q^*_M(s,a) \}$,

π^* is indeed an optimal policy for M'.

□

RL - H.W. - 3:

(Q.2) (a)

The claim is true. Proof:

$\forall s, a$:

$$Q_m^*(s,a) = \mathop{E}_{s' \sim P_m(\cdot|s,a)} \{ r(s,a,s') + \gamma \max_{a' \in A} \{ Q_m^*(s',a') \} \}$$

$$\Rightarrow c \cdot Q_m^*(s,a) = \mathop{E}_{s' \sim P_{m'}(\cdot|s,a)} \{ c \cdot r(s,a,s') + \gamma \max_{a' \in A} \{ c \cdot Q_m^*(s',a') \} \}$$

- $c \cdot Q_m^*(s,a)$: Mult. both sides by c
- $P_{m'}(\cdot|s,a) \equiv P_m(\cdot|s,a)$
- $\max_{a' \in A}$: $c > 0$

$$\Rightarrow c \cdot Q_m^*(s,a) + \frac{b}{1-\gamma} = \mathop{E}_{s' \sim P_{m'}(\cdot|s,a)} \{ c \cdot r(s,a,s') + b + \gamma \max_{a' \in A} \{ c \cdot Q_m^*(s',a') + \frac{b}{1-\gamma} \} \}$$

$\Rightarrow$ Denoting $g: S \times A \to \mathbb{R}$, $\boxed{g(s,a) = c \cdot Q_m^*(s,a) + \frac{b}{1-\gamma}}$ $\Rightarrow$

$\Rightarrow$ We have that $g(\cdot,\cdot)$ satisfies the Bellman-optimality, for M' with r', and from the uniqueness of the optimal-Q-func

$\Rightarrow g(s,a) = \boxed{Q_{m'}^*(s,a) = c \cdot Q_m^*(s,a) + \frac{b}{1-\gamma}}$.

$\Downarrow$

$\blacksquare$

- Now, noting that, $\forall s$:

$$\boxed{\arg\max_{a \in A} \{ Q^*_{M'}(s,a) \}} = \arg\max_{a \in A} \{ c \cdot Q^*_M(s,a) + \frac{b}{1-\gamma} \} =$$

$$= \arg\max_{a \in A} \{ c \cdot Q^*_M(s,a) \} = \boxed{\arg\max_{a \in A} \{ Q^*_M(s,a) \}}$$

↑ Trivial ↑ $c > 0$

$\Rightarrow$ And from here it follows trivially that the set of opt. policies for M and M' are identical.

□

(b) The claim is true. Proof:

$\forall s, a$:

$$Q^*_M(s,a) = \mathbb{E}_{s' \sim P_M(\cdot|s,a)} \{ r(s,a,s') + \gamma \cdot \max_{a' \in A} \{ Q^*_M(s',a') \} \}$$

$\Rightarrow$

$$Q^*_{M'}(s,a) + \phi(s) = \mathbb{E}_{s' \sim P_M(\cdot|s,a)} \{ r(s,a,s') + \phi(s) - \gamma\phi(s) + \gamma \max_{a' \in A} \{ Q^*_{M'}(s',a') + \phi(s') \} \}$$

$\Rightarrow$ following the same arguments as in (a) $\Rightarrow$

$\forall s, a$: $\boxed{Q^*_{M'}(s,a) = Q^*_M(s,a) + \phi(s)}$

$\Downarrow$

(2)

Hence, either from Q.1, or by taking the

$\arg\max_{a \in A}(\cdot)$ from both sides, the claim follows immediately.

□

Question 3:

a) Assume towards contradiction that there exists an iteration t that does not determinates. In particular, there exists a pair (s, a) which are unknown (otherwise the algorithm will stop). The iteration does not stop meaning it gets only to a known pairs, by the definition of the algorithm we get zero rewards on the infinite trajectory, so the optimal policy based on r_t gives zero reward. Since the graph is strongly connected, there exists a path from s_{start} to s which is unknown, so $r_t(s, a) = 1$. So there exists a policy π_t based on r_t that gives a positive reward. This cause a contradiction since the policy π_t gives higher reward that the optimal policy computed based on r_t. Thus, the iteration must terminate.

b) At each iteration there exists some unknown pairs (s, a), otherwise the algorithm has stopped already. From section a, together with the fact that the model is deterministic, we know that no iteration got to an infinite loop. So the optimal policy which computed at the beginning of the iteration will get to some state s with unknown pair (s, a) in order to get a positive reward by choosing action a (since the graph is strongly connected there exists such a path). The number of steps which takes for the algorithm to reach a state s with unknown pair (s, a) is no longer than $|S|$, otherwise due to the determinism of the model and the algorithm we reached some state twice and stuck in infinite loop. So we can conclude that in order to reach an unknown pair the algorithm will run $O(|S|)$ steps and by the definition of the algorithm the iteration will terminate when reached to it.

c) The algorithm will make $O(|S||A|)$ iterations. In each iteration the optimal policy we will reach to exactly one unknown pair (s, a) in order to have a positive reward. The algorithm terminates the iteration after reaching such a pair. So there will be $O(|S||A|)$ updates of the optimal policy before the algorithm will stop.

d) Since every iteration is $O(|S|)$ steps and there are $O(|S||A|)$ iterations we can conclude that:

$$T_{exploration} = O(|S|^2|A|)$$

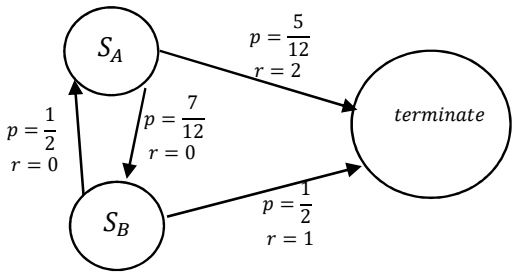
Question 4:

a) We observed state S_A 12 times in the sample. After state S_A we reached S_B 7 times and $terminate$ 5 times. So we will estimate: $(S_B|S_A) = \frac{7}{12}$, $P(terminate|S_A) = \frac{5}{12}$.

We observed state S_B 14 times in the sample. After state S_B we reached S_A 7 times and $terminate$ 7 times. So we will estimate: $(S_A|S_B) = \frac{1}{2}$, $P(terminate|S_B) = \frac{1}{2}$.

From the sample we can see the following:
$S_A, 0 \to S_B$
$S_B, 0 \to S_A$
$S_A, 2 \to terminate$
$S_B, 1 \to terminate$

From here we can estimate the empirical model to be:



b) S_A:

First we count the number of trajectories where we see S_A: $N(S_A) = 9$
Then we sum the rewards we get after visiting S_A at first: $\Gamma(S_A) = 14$
So $V(S_A) = \frac{9}{14}$

S_B:

First we count the number of trajectories where we see S_B: $N(S_B) = 11$
Then we sum the rewards we get after visiting S_B at first: $\Gamma(S_B) = 15$
So $V(S_B) = \frac{11}{15}$

Question 5:

a) Recall that for the true value of $V^\pi(S_t)$ we have $V^\pi(S_t) = r_t + \gamma E[V^\pi(S_{t+1})]$. So if Δ_t uses the true value of $V^\pi(S_t)$ we get:

$$E[\Delta_t | S_t = s] = E[r_t + \gamma V(S_{t+1}) - V(S_t) | S_t = s] = 0$$

It means that in expectation there is no difference between our estimation of V^π to its true value.

b) Calculating:

$$E[\Delta_t | S_t = s, A_t = a] = E[r_t + \gamma V^\pi(S_{t+1}) - V^\pi(S_t) | S_t = s, A_t = a] =$$
$$= r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a) V^\pi(s') - V^\pi(s) = Q^\pi(s,a) - V^\pi(s)$$

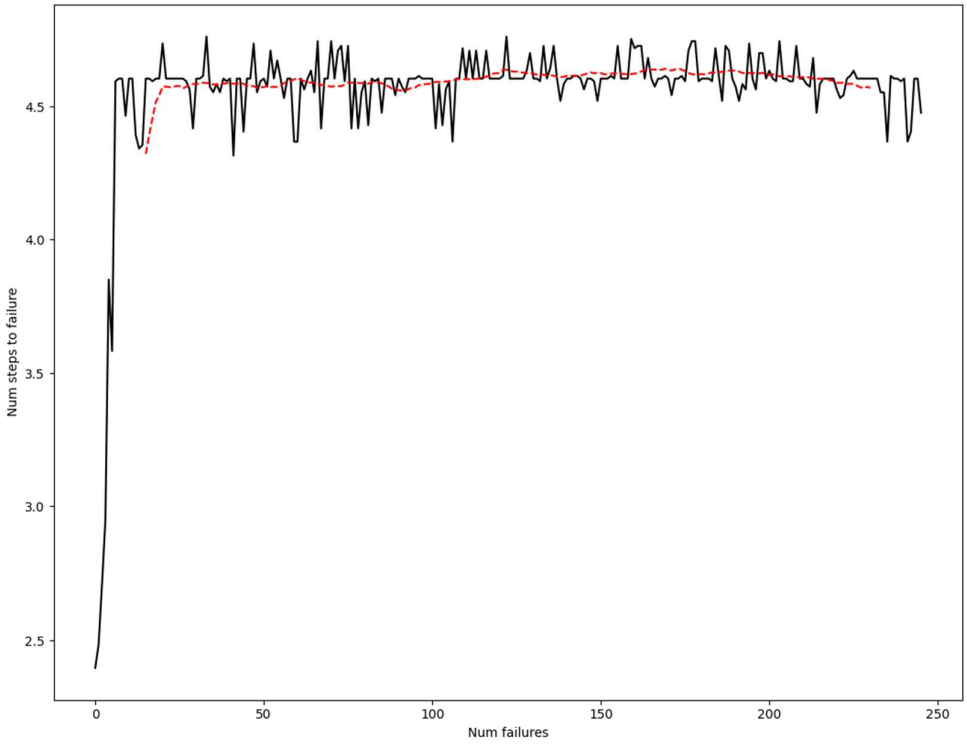
The result means that given a state $s \in S$ and action $a \in A$, a step of $\mathrm{TD}(0)$ is towards $Q^\pi(s,a)$ from $V^\pi(s)$ with step size depended on the learning rate.

Programming:

Question 1:

a) Due to some randomness on the algorithm we got different result at each run. The plot below show convergence after 246 failures.

b) Plot of the learning curve:



Question 2:

a) The percent of successful episodes was between 30-60%. In the run of the attached Q table it was 51.6%. This is the Q table:

```
Final Q-Table Values
[[1.88233127e-01 1.60772962e-01 1.72983054e-01 9.18197184e-02]
 [1.72371710e-04 8.83685923e-05 9.09437756e-04 1.88298644e-01]
 [8.86425829e-04 3.46473081e-04 2.94075026e-04 6.95836057e-02]
 [3.09079705e-05 1.52326524e-04 1.39499168e-04 3.13047660e-02]
 [1.32784861e-01 4.35333973e-04 3.81183799e-03 7.51734097e-04]
 [0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00]
 [7.06803534e-05 1.71821247e-04 5.70270432e-05 4.16538625e-04]
 [0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00]
 [3.57001992e-03 1.41601500e-04 3.48644525e-03 4.02401056e-01]
 [0.00000000e+00 2.54617833e-01 0.00000000e+00 0.00000000e+00]
 [5.59964321e-04 1.19640275e-01 2.20865705e-04 0.00000000e+00]
 [0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00]
 [0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00]
 [0.00000000e+00 0.00000000e+00 5.61329169e-01 0.00000000e+00]
 [0.00000000e+00 9.16909569e-01 0.00000000e+00 0.00000000e+00]
 [0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00]]
```

2.

The percent of successful episodes was between 10-45%.

The expected value of successful episodes in this method seems lower than the tabular method.