CS 234: Assignment #3

Due date: 5/24 13:30 PM PST

These questions require thought, but do not require long answers. Please be as concise as possible.

We encourage students to discuss in groups for assignments. **However, each student must finish the problem set and programming assignment individually, and must turn in her/his assignment.** We ask that you abide by the university Honor Code and that of the Computer Science department, and make sure that all of your submitted work is done by yourself. If you have discussed the problems with others, please include a statement saying who you discussed problems with. Failure to follow these instructions will be reported to the Office of Community Standards. We reserve the right to run a fraud-detection software on your code.

Please review any additional instructions posted on the assignment page at http://cs234.stanford.edu/assignment3. When you are ready to submit, please follow the instructions on the course website.

1 Frozen Lake

Now we are going to implement algorithms with smart exploration for the Frozen Lake environment. If you recall the Frozen Lake environment, the layout of the environment is described using a grid. The layout is as following:

S	*H*	*H*	*H*
F	*H*	*H*	*H*
F	*H*	*H*	*H*
F	*F*	*F*	*G*

In this environment, the agent start from top left. It needs to reach the goal at bottom right. The max steps it could take in this 4×4 layout is 6. The agent cannot move to the hole's location. We have provided custom versions of this environment in the starter code.

(a) (coding) Implement Rmax in q1.py. With default hyper-parameters, it should reach the goal quickly.

(b) (Written) How would the threshold m influence the performance? Please plot the running average of the first 10,000 training episodes to support your argument.The x-axis should be the episode number, and the y-axis should be the average score over all training episodes so far. Include the plot in your writeup.

(c) (coding and Written) Implement $\epsilon - greedy$ Q-learning (minimal changes from assignment 1) in q2.py. Experiment with different values of ϵ and choose one that works well. Compare Rmax (with the best performing m) with $\epsilon - greedy$. What do you observe? Plot performance on the same axis as part (b).

2 Expected Regret Bounds

Assume a reinforcement learning algorithm A for discounted infinite-horizon MDPs has expected regret

$$\mathbb{E}_* \left[\sum_{t=1}^{T} r_t \right] - \mathbb{E}_A \left[\sum_{t=1}^{T} r_t \right] = f(T)$$

for all $T > 0$, where $\mathbb{E}_*$ is over the probability distribution with respect to the optimal policy π_* and $\mathbb{E}_A$ is the expectation with respect to the algorithm's behavior. We assume that $\gamma \in [0, 1)$ is the discount factor and that rewards are normalized, i.e., $r_t \in [0, 1]$.

(a) Let π be an arbitrary policy or algorithm. Show that for any $\epsilon' > 0$ and $T' \geq \log_{\frac{1}{\gamma}} \frac{H}{\epsilon'}$ where $H = 1/(1-\gamma)$, we have

$$\left| V_\pi(s) - \sum_{t=1}^{T'} \gamma^{t-1} \mathbb{E}_\pi[r_t | s_1 = s] \right| \leq \epsilon' \,, \text{ for all state } s.$$

Note V_π is the value function associated with π and $\mathbb{E}_\pi$ is the expectation with respect to the randomization of π.

(b) From the regret guarantee of algorithm A and Part a), it follows that for any $\epsilon' > 0$ and $T' \geq \log_{\frac{1}{\gamma}} \frac{H}{\epsilon'}$, we have

$$\mathbb{E}_*[V_*(s_{T+1})] - \mathbb{E}_A[V_A(s_{T+1})] \leq f(T' + T) - f(T) + 2\epsilon' \,, \text{ for } T > 0,$$

where V_A is the value function of the (possibly nonstationary) policy that algorithm A follows. Assume now $f(T) = \sqrt{T}$. Show that for any $\epsilon > 0$ and $t \geq 1 + \frac{1}{\epsilon^2} \left(\log_{\frac{1}{\gamma}} \frac{4H}{\epsilon} \right)^2$, we have

$$\mathbb{E}_*[V_*(s_t)] - \mathbb{E}_A[V_A(s_t)] \leq \epsilon \,.$$

Hint: It may be helpful to set ϵ' to be some function of ϵ.