CS 7642: Reinforcement Learning and Decision Making **Homework #4**

Q-Learning

1 Problem

1.1 Description

In this homework, you will have the complete reinforcement-learning experience: training an agent from scratch to solve a simple domain using Q-learning.

The environment you will be applying Q-learning to is called Taxi (Taxi-v3). The Taxi problem was introduced by Dietterich 1998 and has been used for reinforcement-learning research in the past. It is a grid-based environment where the goal of the agent is to pick up a passenger at one location and drop them off at another.

The map is fixed and the environment has deterministic transitions. However, the distinct pickup and drop-off points are chosen randomly from 4 fixed locations in the grid, each assigned a different letter. The starting location of the taxicab is also chosen randomly.

The agent has 6 actions: 4 for movement, 1 for pickup, and 1 for drop-off. Attempting a pickup when there is no passenger at the location incurs a reward of -10. Dropping off a passenger outside one of the four designated zones is prohibited, and attempting it also incurs a reward of -10. Dropping the passenger off at the correct destination provides the agent with a reward of 20. Otherwise, the agent incurs a reward of -1 per time step.

Your job is to train your agent until it converges to the optimal state-action value function. You will have to think carefully about algorithm implementation, especially exploration parameters.

1.2 Q-learning

Q-learning is a fundamental reinforcement-learning algorithm that has been successfully used to solve a variety of decision-making problems. Like Sarsa, it is a model-free method based on temporal-difference learning. However, unlike Sarsa, Q-learning is *off-policy*, which means the policy it learns about can be different than the policy it uses to generate its behavior. In Q-learning, this *target* policy is the greedy policy with respect to the current value-function estimate.

1.3 Procedure

- The answer you provide should be the optimal Q-value for a specific state-action pair of the Taxi environment.

  Provide answers for the specific problems you are given on Canvas. Your answers must be correct to 3 decimal places, truncated (e.g., 3.14159265 becomes 3.141).

- To solve this problem you should implement the Q-learning algorithm and use it to solve the Taxi environment. The agent should explore the MDP, collect data to learn an optimal policy and also the optimal Q-value function. Be mindful of how you handle terminal states: if S_t is a terminal state, then $V(S_t)$ should always be 0. Use $\gamma = 0.90$—this is important, as the optimal value function depends on the discount rate. Also, note that an ϵ-greedy strategy can find an optimal policy despite finding sub-optimal Q-values. As we are looking for optimal Q-values you will have to carefully consider your exploration strategy.

2 Examples

The following examples can be used to verify that your agent is implemented correctly.

- $Q(462, 4) = -11.374$
- $Q(398, 3) = 4.348$
- $Q(253, 0) = -0.585$
- $Q(377, 1) = 9.683$
- $Q(83, 5) = -13.996$

3 Resources

3.1 Lectures

- Lesson 4: Convergence
- Lesson 7: Exploring Exploration

3.2 Readings

- Chapter 6 (6.5 Q-learning: Off-policy TD Control) of Sutton and Barto 2020
- Chapter 2 (2.6.1 Q-learning) of Littman 1996

3.3 Documentation

- `http://gym.openai.com/docs/`
- `https://github.com/openai/gym/blob/master/gym/envs/toy_text/taxi.py`

4 Submission Details

The due date is indicated on the Canvas page for this assignment.
Make sure you have set your timezone in Canvas to ensure the deadline is accurate.
Submit your answers on Canvas, as outlined in section 1.3. You will have a total of 10 submission attempts - only the highest score is kept.

References

[Die98] Thomas G Dietterich. “The MAXQ Method for Hierarchical Reinforcement Learning.” In: *ICML*. Vol. 98. Citeseer. 1998, pp. 118–126.

[Lit96] Michael Lederman Littman. *Algorithms for Sequential Decision Making*. 1996.

[SB20] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. 2nd Ed. MIT press, 2020. URL: `http://incompleteideas.net/book/the-book-2nd.html`.