Reinforcement Learning

Homework #4

Q-Learning

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 (https://github.com/openai/gym/blob/master/gym/envs/toy_text/taxi.py) (Taxi-v3). The Taxi problem was introduced by Dietterich 1998 (https://www.jair.org/index.php/jair/article/download/10266/24463) 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.

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.

Procedure

- You should return the optimal Q-value for a specific state-action pair of the Taxi environment.
- 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.

Resources

The concepts explored in this homework are covered by:

- Lesson 4: Convergence
- Lesson 7: Exploring Exploration
- Chapter 6 (6.5 Q-learning: Off-policy TD Control) of http://incompleteideas.net/book/the-book-2nd.html
- Chapter 2 (2.6.1 Q-learning) of 'Algorithms for Sequential Decision Making', M. Littman, 1996

Submission

- The due date is indicated on the Syllabus page for this assignment.
- Use the template code to implement your work.
- Please use *python 3.6.x, gym==0.17.2, numpy==1.18.0* or their more recent versions, and you can use any core library (i.e., anything in the Python standard library). No other library can be used.