CS 7642: Reinforcement Learning and Decision Making **Homework #3**

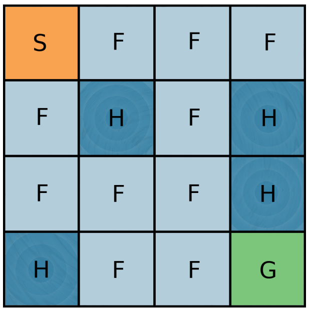
SARSA

1 Problem

1.1 Description

For this assignment, you will build a SARSA agent which will learn policies in the OpenAI Gym Frozen Lake environment. You will be given randomized Frozen Lake maps, with corresponding sets of parameters to train your SARSA agent; if your agent is implemented correctly, after training for the specified number of episodes it will produce the same policies (which are not necessarily optimal policies) as the grader.

Frozen Lake is a grid world environment that is highly stochastic, where the agent must cross a slippery frozen lake which has deadly holes to fall through. The agent begins in the starting state S and is given a reward of 1 if it reaches the goal state G. The agent can take one of four possible moves at each state (left, down, right, or up). The frozen cells F are slippery, so the agent's actions succeed only $\frac{1}{3}$ of the time, while the other $\frac{2}{3}$ are split evenly in orthogonal directions. If the agent lands in a hole H, then the episode terminates, and the agent is given a reward of 0.



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

Figure 1: Example Frozen Lake Map

1.2 SARSA $(s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1})$

SARSA is a model-free on-policy reinforcement learning algorithm that solves the control problem through trial-and-error learning. It is model-free because, unlike with value iteration and policy iteration, it does not need or use an MDP. It is on-policy because it learns about the same policy that generates behaviors. The algorithm estimates the action-value function Q_π of the behavior policy π, and uses an exploration strategy to improve π while increasing the policy's greediness.

1.3 Procedure

1. Install OpenAI Gym **0.14.0** with `pip install gym==0.14.0`
2. Implement a **SARSA** agent
3. Train your agent given the specified parameters
4. Export the policy and enter it into Canvas

1.4 Notes

- You must use Python, NumPy, and OpenAI Gym 0.14.0 for this homework
- Use the **provided seed** for both Gym and NumPy

- Initialize the agent's Q-table to zeros
- To avoid any unexpected behavior, setup the Gym environment with `gym.envs.toy_text.frozen_lake.FrozenLakeEnv().unwrapped`
- To set up the environment with a custom map, use the environment's `desc` attribute. Note that the environment expects a square map so you will need to reshape it.
- You must train your agent with an epsilon-greedy exploration strategy, using NumPy's `numpy.random.randint` function to select random actions

1.5 Examples

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

- Input: amap='SFFFHFFFFFFFFFFG', gamma=1.0, alpha=0.25, epsilon=0.29, n_episodes=14697, seed=741684, Output:

$$\wedge, \vee, \vee, >, <, >, >, \vee, \vee, \vee, >, \vee, >, >, >, <$$

- Input: amap='SFFFFHFFFFFFFFFFFFFFFFFFG', gamma=0.91, alpha=0.12, epsilon=0.13, n_episodes=42271, seed=983459, Output:

$$\wedge, >, >, >, >, <, >, >, >, \vee, \vee, \vee, >, >, \vee, \vee, >, >, >, >, \vee, >, >, \wedge, <$$

- Input: amap='SFFG', gamma=1.0, alpha=0.24, epsilon=0.09, n_episodes=49553, seed=202404, Output:

$$<, <, \vee, <$$

1.6 Resources

1.6.1 Lectures

- Lesson 4: Convergence

1.6.2 Readings

- Chapter 6 (6.4 Sarsa: On-policy TD Control) of Sutton and Barto 2018

1.7 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. To complete the assignment calculate answers to the specific problems given and submit results to Canvas

References

[SB18] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction.* MIT press, 2018.