CS6700 - Reinforcement Learning
Programming Assignment 1

February 10, 2023

1 Environment Description

This exercise aims to familiarize you with two popular Temporal Difference Learning algorithms: **SARSA and Q-Learning**. You will solve several variants of the Grid World problem (a sample world is shown in Figure 2).

This is a grid world with 4 deterministic actions ('up', 'down', 'left', 'right'). The agent transitions to the next state determined by the direction of the action chosen with a probability of $p \in [0, 1]$. We also define a parameter called $b \in [0, 1]$. Consider the direction of the action chosen as the agent's "North". For example, if the action is 'left', it is the agent's North, and the agent's East would be the direction of the action 'up'. Figure 1 provides an illustration of the same. The agent transitions to the state West of the chosen action with probability $(1-p) \times b$, and to the East of the chosen action with probability $(1-p) \times (1-b)$.

The environment may also have a wind blowing that can push the agent one **additional** cell to the right **after transitioning to the new state** with a probability of 0.4. An episode is terminated either when a goal is reached or when the timesteps exceed 100. Transitions that take you off the grid will not result in any change in state.

The dimensions of the grid are 10×10. The following types of states exist:

- ***Start state***: The agent starts from this state.
- ***Goal state***: The goal is to reach one of these states. There are 3 goal states in total.
- ***Obstructed state***: These are walls that prevent entry to the respective cells. Transition to these states will not result in any change.
- ***Bad state:*** Entry into these states will incur a higher penalty than a normal state.
- ***Restart state:*** Entry into these states will incur a very high penalty and will cause agent to teleport to the start state without the episode ending. Once the restart state is reached, no matter what action is chosen, it goes to the start state at the next step.
- ***Normal state:*** None of the above. Entry into these states will incur a small penalty.

Rewards: -1 for normal states, -100 for restart states, -6 for bad states, +10 for goal states.

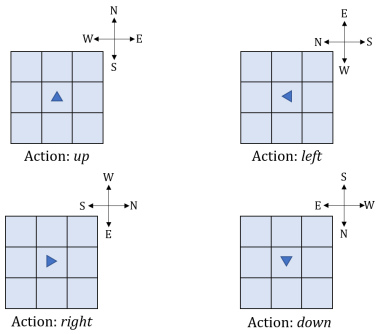

Figure 1: The intended direction of the action chosen is considered as 'North'

Additional Information

The code for the environment can be found here. The `env.step()` function takes as arguments the current state and action, and returns the reward and next state. The appropriate termination conditions have to be specified by the student in the code (as explained in section 1. `env.reset()` resets the environment. In the notebook containing the code, cell 1 contains the environment class, cell 2 contains the environment instantiation, and cell 3 lists some environment variables. For each experiment, the start state is fixed and does not change. Different experiments may have different start states.

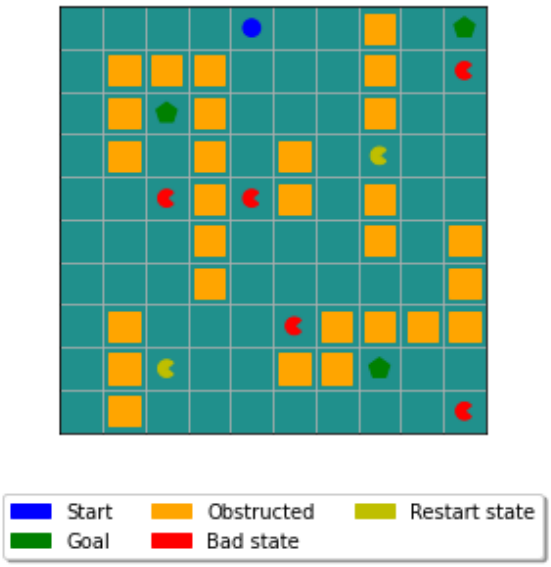

Figure 2: An example grid world with start position $(0, 4)$

2 Tasks

- Implement SARSA and Q-Learning.
- For each algorithm, run experiments with wind=False and wind=True; two different start states: $(0, 4)$, $(3, 6)$; two values of p (1.0, 0.7); and two types of exploration

strategies (ϵ-greedy and softmax), making it **16 different configurations** in total.

- For each of the 16 configurations, determine the best set of hyperparameters (ϵ in ϵ-greedy exploration, temperature β in softmax exploration, learning rate α, and discount factor γ) and plot the following:

    1. Reward curves and the number of steps to reach the goal in each episode (during the training phase with the best hyperparameters).
    2. Heatmap of the grid with state visit counts, i.e., the number of times each state was visited throughout the training phase.
    3. Heatmap of the grid with Q values after training is complete, and optimal actions for the best policy.

For each of the algorithm, provide a written description of the policy learnt, explaining the behavior of the agent, and your choice of hyperparameters. This description should also provide information about how the behavior changes with and without the wind, for different levels of stochasticity and for different start states.

Figure 3 summarizes how you can structure your report.

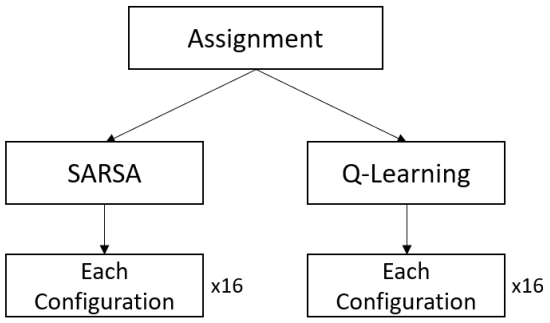

Figure 3: The report can be organized in this manner

3 Submission Instructions

You are required to submit both your report and your code (details will be announced on Moodle). Please submit in **teams of two**. One submission per team will suffice. The due date for this programming assignment is **11:59 pm on Friday, Feb 24.**