Project 3
Artificial Intelligence

Fall 2024

[Solutions to this assignment must be submitted via CANVAS prior to midnight on the due date which is 20 November 2024.]

This project may be undertaken in pairs or individually. If working in a pair, state the ***names*** of the two people undertaking the project in your report. Only ONE submission should be made per group.

Purpose: To gain a thorough understanding of the working of a robot that is implemented as a reinforcement learning agent. The robot needs to navigate a grid that contains obstacles and hazards. The robot can start in any position along the grid that is either not a position occupied by an obstacle or is not the destination. Both obstacles and the destination position are fixed and indicated in the grid below. The grid is modelled on the scenario that was discussed in class under Exercise 1.

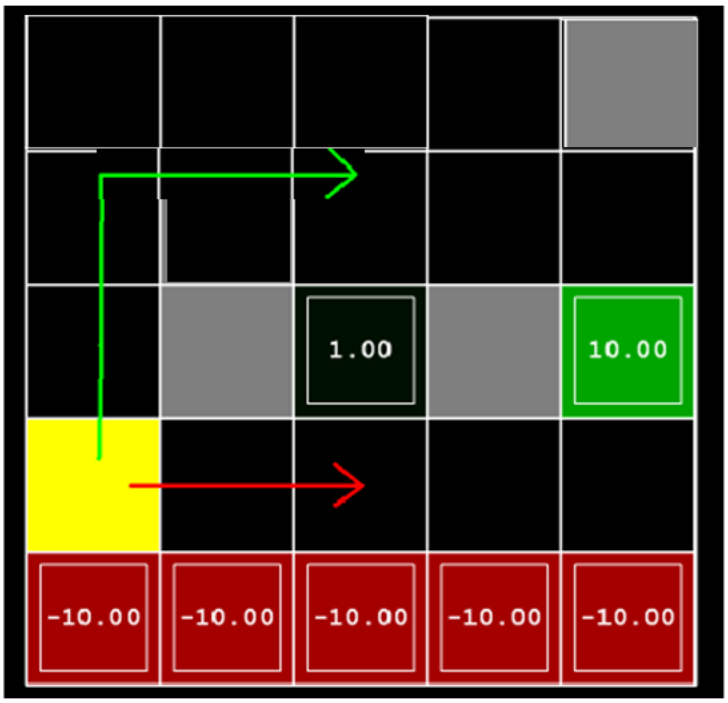

Figure 1 Original environment

Environment Description: The environment in which the robot navigates is a 5 by 5 grid. Hazards are represented by the grid positions in row 1 (rows are numbered from bottom to top while columns are numbered from left to right) as given in Figure 1. All hazards are given a negative reward of -10. In addition to hazards, 3 obstacles exist at points (3,2), (3,4) and (5,5) on the grid (indexing by row and then by column). When the robot *collides with an obstacle it will need to return to the grid position that it occupied before the collision*. An episode consists of the robot moving from its starting position and then attempting to reach the higher reward of +10 at position (3,5). The starting position of the robot varies randomly between episodes and can take any values (i.e., grid position) as long as it does not coincide with an obstacle, hazard or one of the exit positions (positions with positive or negative rewards).

The navigation rule (from any position on the grid) is as follows. A maximum of four actions, Up, Down, Left and Right are possible from any given position. Navigation in the intended direction occurs with (1-p) % probability and movement in unintended directions (taken as a sum) add up to p%, where p is the degree of noise. Note that movements never take place opposite (at 180 degrees) to the intended direction. The noise level is 0.1 and the discount factor $\gamma = 0.99$.

Jot down any questions/doubts that you may have and feel free to ask me questions in class or in person. Together with your partner, ***work out a strategy before you start coding the solution in Python.*** Note that this project, unlike the previous two projects, has only one milestone and hence we expect that you on your own will carry on the good practice of designing a solution before implementing it.

Given the limited timeframe for the project, some simplifications have been applied.

Your task in this project is to implement the following requirements. The project has only ONE milestone (one submission) which has the following requirements. See Tutorial 5 for Project 3 for a discussion on setting rewards and transition probabilities.

Requirements

Your Python code in Collab should meet the following requirements. All the outputs required in R1-R3 must be displayed by your Python program. The outputs should also be included in the pdf report file that you hand in.

R1

Your task in this requirement is to use the MDP procedure with the objective of creating a policy that will enable the robot to retrieve the reward of +10 *without risking the cliff* (i.e., not walking parallel to the cliff). The cliff is represented by the bottom row consisting of negative reward cells. With a realistic noise factor of 0.1 (instead of the unrealistic value of 0.5 used in the solution discussed in class), your strategy will be to set a live-in reward r for all cells which are not obstacles or exit states). Experiment with values of r in the range [-0.3,-0.1] in increments of 0.05 and determine the first value of r in the range [-0.3,-0.1] that enables you to create the best possible policy P1. P1 is the policy that maximizes the sum of rewards and is given by:

$P1 = \arg max_P \left\{\sum_{p \in P} \sum_{c \in p} value(c)\right\}$ where p is a path supported by policy P, c is a cell along path p and value(c) is the value of the cell c along path p.

Once you have created policy P1 you need to:

a) State the first value of r in the range [-0.3,-0.1] that enables you to create policy P1.
b) Visualize the policy P1 in one of two ways: (1) Draw it with arrows showing the path in the same style as discussed in the lectures or if you prefer using (2) Populate each cell in the grid with numeric values produced after running the Value Iteration algorithm. **(8 marks)**

R2

This task requires no programming but requires you to test how robust your policy is to changes in the environment. Note that the only change to the grid is that the obstacle at (5,5) is now at (4.4) as shown in Figure 2 below.

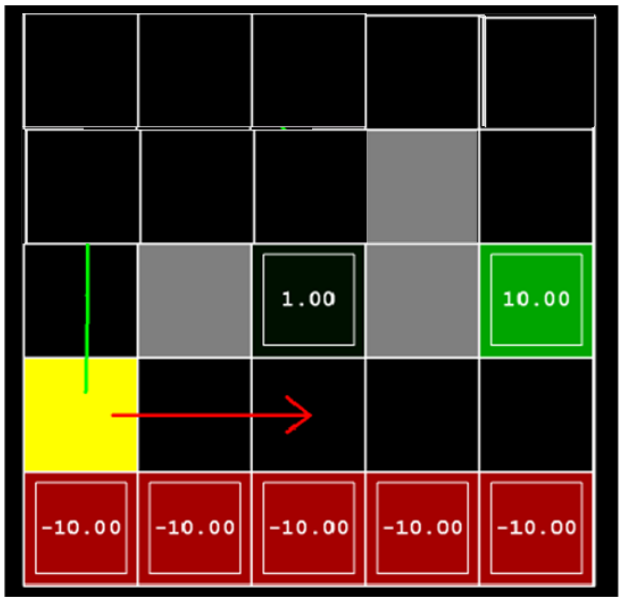

Figure 2: New environment with sifting of obstacle from (5,5) to (4.4)

Apply the MDP procedure to this new environment and determine the best policy P2 using the same criteria that you used in R1 above.

Is P2 the same as P1? If yes, why does P1 produce the same results in the new environment? If not, what are the differences and why do these differences exist? **(3 marks)**

R3

One limitation of the system implemented is that it is restricted to a single agent (robot). In practice many agents may be required to navigate at the same time while avoiding collisions with not just walls but also colliding with each other. An example of this is an automated warehouse which uses multiple robots to service different customer orders in parallel with each other. The challenge here is to cope with obstacles that are mobile. The MDP algorithm only accommodates a static environment.

Suggest a suitable modification (without an actual implementation) to the MDP procedure that will enable it to be applied to this type of environment. Hint: Positions that are free of fixed obstacles now may be occupied by robots with a certain probability. Assume that you have a formula to compute this probability. How will you make use of this probability in the MDP procedure? Your explanation needs to be at most 2 paragraphs in length but must be convincing. You may use pseudo code to present your answer if you so wish.

(3 marks)

Notes:

Produce ONE pdf document that contains Python code. If you submit an image version of your pdf code file, it will not be graded.
Also submit a separate pdf (report pdf) that contains answers to the questions. Do NOT bury answers to questions as comments in your code.
Also submit a publicly accessible link to your Collab code file.

End of project specification