3460:460/560 AI, Project 2 – MDP

Problem Description: MDP is a sequential decision problem for a fully observable, stochastic environment with a Markovian transition model and additive rewards. In this project, you will implement policy evaluation and value iteration for an MDP in a grid-world like the one given in the book (Figure 17.1).

You are strongly encouraged to complete the project in Python although C++ is permitted.

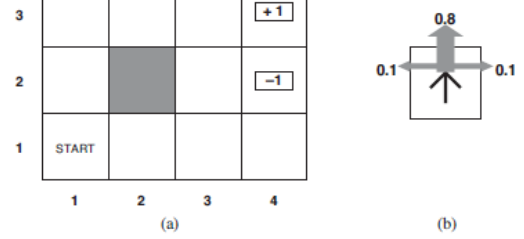

Figure 17.1 (a) A simple 4×3 environment that presents the agent with a sequential decision problem. (b) Illustration of the transition model of the environment: the "intended" outcome occurs with probability 0.8, but with probability 0.2 the agent moves at right angles to the intended direction. A collision with a wall results in no movement. The two terminal states have reward +1 and –1, respectively, and all other states have a reward of –0.04.

Part I: Policy evaluation (assume max number of iterations = 20, discount factor γ = 0.95).

- Given the MDP same as described in Figure 17.1, except the reward function. For your project, the reward to the two terminal states (3,4) = 1 and (2,4) = -1 but the reward to all other states will be a variable although the values are the same. (for example, a reward of -0.04 to all other states. Another example, a reward of -0.1 to all other states.) The reward value will be passed to your program as a command-line argument.
- Policy (a .csv file): a policy for the MDP, a 3x4 matrix of integers, 1 for up, -1 for down, 2 for right, -2 for left (a test case : https://www.cs.uakron.edu/~duan/classes/460/projects/project2/case1.csv). Your program will take the policy filename from command-line, i.e. pass the filename as a command-line argument.
- Output: expected utility of the given policy, output to the console.

  Note: When we grade your project, the command we will use will be something like: python3 -0.05 case1.csv
  If you've never used command-line arguments before, it's a good time to learn it.
  https://www.geeksforgeeks.org/command-line-arguments-in-python/

Part II: Value iteration (assume max number of iterations = 20, discount factor γ = 0.95).

- Given a MDP same as described in Figure 17.1, except the transition function and reward function. For your project, the transition probability is a variable for the intended direction, for example 0.8 as outlined in Figure 17.1, and 0.1 to the left/right of the intended direction; another example 0.6 and 0.2 to the left/right of the intended direction. The reward function is the same as outlined in Part I.
- Output: optimal policy (.cvs file). Save your best policy for the given MDP to a file. Name your file "expectimax.csv".

  Note: if there is a tie, here is the precedence: up > right > down > left (clock wide order, starting with up).

  Note: we will grade your part2 using "python3 0.7 -0.05" (1st number represents the transition probability to the intended direction, 2nd number for rewards to states other than the two terminal states).

Submission instructions:

Submit an electronic copy of the program using project2 dropbox at Brightplace. You are required to zip your submission inside an archive. Follow these steps:

1. Create a folder named zippy_2 (but use your name/uanetID).
2. Place just the source files inside the folder.
3. Right-click on the folder and choose Send To... Compressed Folder (or use some other Zip utility to archive the entire folder). I think two files should be sufficient for this project, project2_part1.py, project2_part2.py.
4. Drop this single zipped file in your dropbox at Brightplace.

Be sure to submit your working solution before the due date! Do not submit non-working programs. The submission time will be used to assess late penalties.