CS 533 Intelligent Agents and Decision Making, Spring 2018

Homework #2: Optimizing Finite-Horizon Expected Total Reward

Yathartha Tuladhar

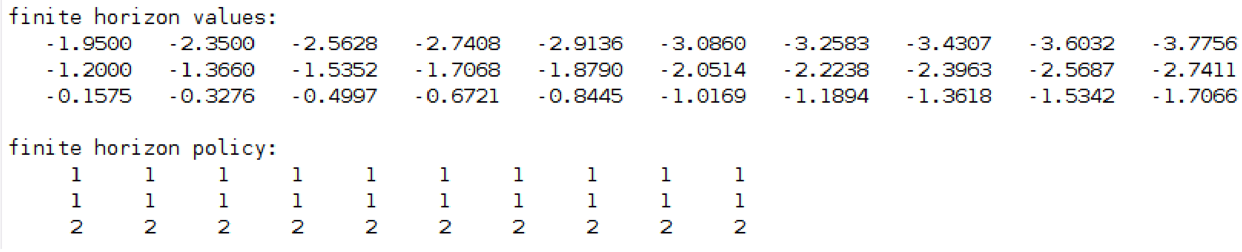
Instructions to run the code:

The main file to run the code is “Homework2_main.m”. The MDP class and relevant functions are defined in “MDP.m”.

- In “Homework2_main.m”, the user can set which part of the assignment to run, gamma(discount factor) and finite horizon time step (tStep).
- “MDPtest.txt”, is the input file for Part 1.
- “Part2_Own_MDP_Gridworld.txt” is the input file for Part 2, where I designed an MDP for a 5x5 gridworld.
- “MDP1.txt” and “MDP2.txt” are the input files for Part 3.

Part 1: Build a Planner

A planner as built using the instructions given in the assignment for part 1. The program takes in a text file, and parses it to extract the MDP and all the necessary information. Then it calculates the value function and the policy for this MDP. The results are shown below for a time horizon of 10.



```
finite horizon values:
   -1.9500   -2.3500   -2.5628   -2.7408   -2.9136   -3.0860   -3.2583   -3.4307   -3.6032   -3.7756
   -1.2000   -1.3660   -1.5352   -1.7068   -1.8790   -2.0514   -2.2238   -2.3963   -2.5687   -2.7411
   -0.1575   -0.3276   -0.4997   -0.6721   -0.8445   -1.0169   -1.1894   -1.3618   -1.5342   -1.7066

finite horizon policy:
     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1
     2     2     2     2     2     2     2     2     2     2
```

The finite horizon values towards the left are when k -> 0.
The finite horizon value at the far right is for k = 10.
As predicted state 3 has the highest value. Taking action 1 in states 1, and 2 have a greater probability to lead to state 3, and once it is in state 3, it will try to stay there as much as possible by taking action 2.

Part 2: Create Your Own MDP

The MDP that I created is a 5x5 gridword. The reward setup is shown below.

Reward for states:

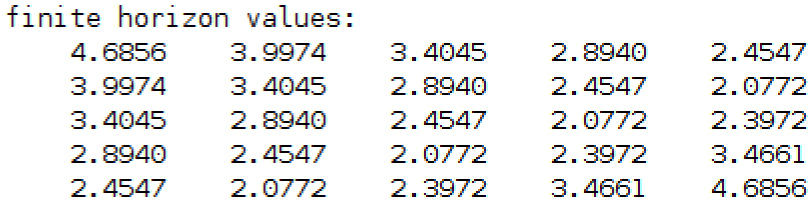
+1	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	+1

The transition function is set as:

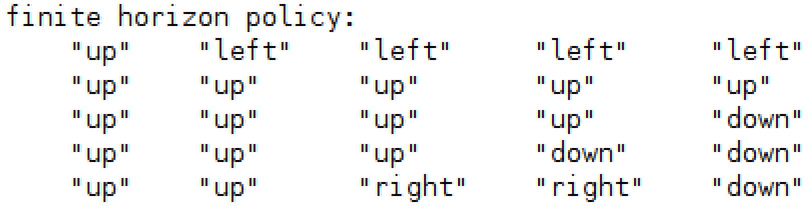
- There are four actions {up, down, left, right}
- If the action leads to a wall, probability P=1, for transitioning into the same state
- If the action is ‘left’, there is P = 0.8 for transitioning into the state to the left, and P = 0.2 for transitioning to the same position
- Similarly, for all actions (unless it hits a wall) P = 0.8 for transitioning into the intended direction, and P = 0.2 for transitioning into its current state.
- The episode does not end when it reaches the reward (+1) states. Thus it can stay there to collect rewards

The value function and policy for this MDP is shown below for time horizons of 5 and 10:

For time horizon = 5:

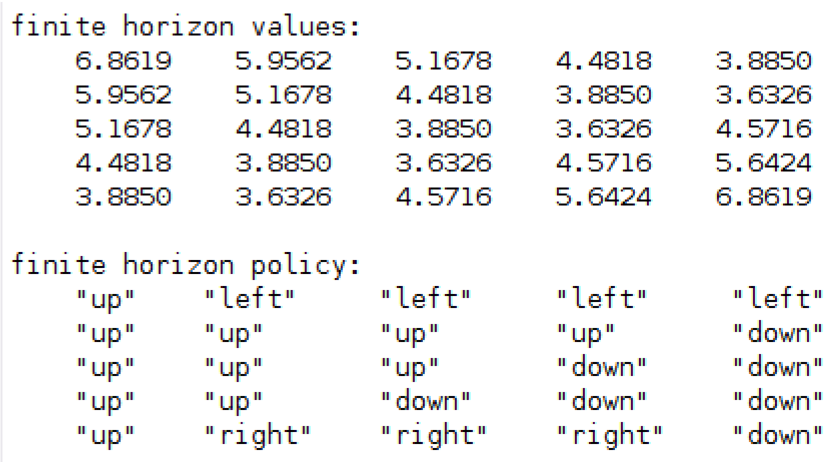


```
finite horizon values:
    4.6856    3.9974    3.4045    2.8940    2.4547
    3.9974    3.4045    2.8940    2.4547    2.0772
    3.4045    2.8940    2.4547    2.0772    2.3972
    2.8940    2.4547    2.0772    2.3972    3.4661
    2.4547    2.0772    2.3972    3.4661    4.6856
```



```
finite horizon policy:
    "up"    "left"    "left"     "left"     "left"
    "up"    "up"      "up"       "up"       "up"
    "up"    "up"      "up"       "up"       "down"
    "up"    "up"      "up"       "down"     "down"
    "up"    "up"      "right"    "right"    "down"
```

For time horizon = 10:



```
finite horizon values:
    6.8619    5.9562    5.1678    4.4818    3.8850
    5.9562    5.1678    4.4818    3.8850    3.6326
    5.1678    4.4818    3.8850    3.6326    4.5716
    4.4818    3.8850    3.6326    4.5716    5.6424
    3.8850    3.6326    4.5716    5.6424    6.8619

finite horizon policy:
    "up"    "left"     "left"     "left"     "left"
    "up"    "up"       "up"       "up"       "down"
    "up"    "up"       "up"       "down"     "down"
    "up"    "up"       "down"     "down"     "down"
    "up"    "right"    "right"    "right"    "down"
```

As one can see, the policy always points towards the top-left, or the bottom-right corner. Moreover, the value function values for horizon of 10 is greater than that of a horizon of 5. This is because once the agent reaches the rewarding state, it can choose to stay there and receive rewards until the horizon is over.

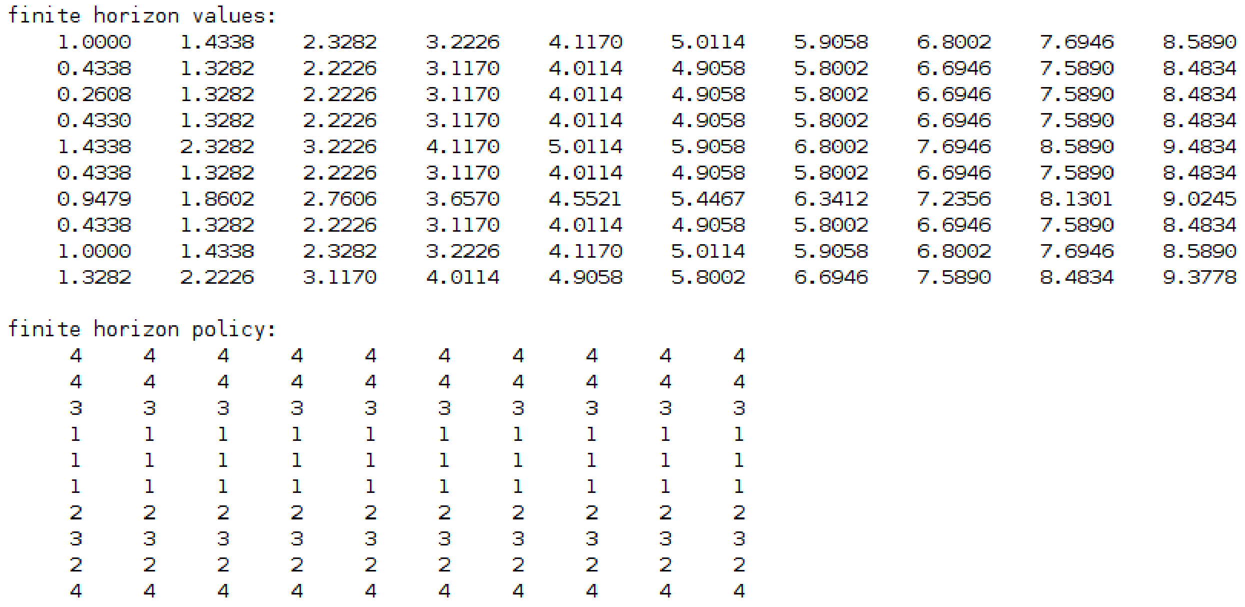
The policy for the time horizon of 10 is an optimal policy (but not necessarily unique). The value function calculated above, is the optimal value function.

Note: Eventhough this is a finite horizon case, for this part I used a discounting factor 0.9.

Part 3: More Testing

The results for the value iteration, and it’s corresponding policy for Part 3 are shown below for a horizon of 10.

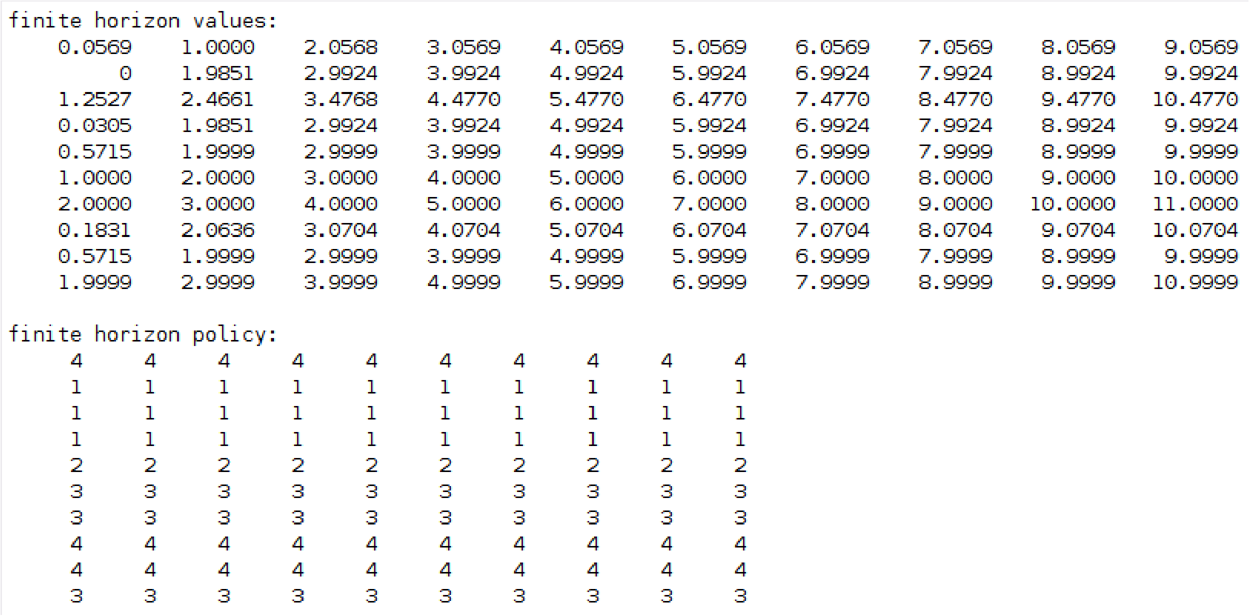
Results for MDP1.txt:



```
finite horizon values:
    1.0000    1.4338    2.3282    3.2226    4.1170    5.0114    5.9058    6.8002    7.6946    8.5890
    0.4338    1.3282    2.2226    3.1170    4.0114    4.9058    5.8002    6.6946    7.5890    8.4834
    0.2608    1.3282    2.2226    3.1170    4.0114    4.9058    5.8002    6.6946    7.5890    8.4834
    0.4330    1.3282    2.2226    3.1170    4.0114    4.9058    5.8002    6.6946    7.5890    8.4834
    1.4338    2.3282    3.2226    4.1170    5.0114    5.9058    6.8002    7.6946    8.5890    9.4834
    0.4338    1.3282    2.2226    3.1170    4.0114    4.9058    5.8002    6.6946    7.5890    8.4834
    0.9479    1.8602    2.7606    3.6570    4.5521    5.4467    6.3412    7.2356    8.1301    9.0245
    0.4338    1.3282    2.2226    3.1170    4.0114    4.9058    5.8002    6.6946    7.5890    8.4834
    1.0000    1.4338    2.3282    3.2226    4.1170    5.0114    5.9058    6.8002    7.6946    8.5890
    1.3282    2.2226    3.1170    4.0114    4.9058    5.8002    6.6946    7.5890    8.4834    9.3778

finite horizon policy:
     4     4     4     4     4     4     4     4     4     4
     4     4     4     4     4     4     4     4     4     4
     3     3     3     3     3     3     3     3     3     3
     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1
     2     2     2     2     2     2     2     2     2     2
     3     3     3     3     3     3     3     3     3     3
     2     2     2     2     2     2     2     2     2     2
     4     4     4     4     4     4     4     4     4     4
```

Results for MDP2.txt:



```
finite horizon values:
    0.0569    1.0000    2.0568    3.0569    4.0569    5.0569    6.0569    7.0569    8.0569    9.0569
         0    1.9851    2.9924    3.9924    4.9924    5.9924    6.9924    7.9924    8.9924    9.9924
    1.2527    2.4661    3.4768    4.4770    5.4770    6.4770    7.4770    8.4770    9.4770   10.4770
    0.0305    1.9851    2.9924    3.9924    4.9924    5.9924    6.9924    7.9924    8.9924    9.9924
    0.5715    1.9999    2.9999    3.9999    4.9999    5.9999    6.9999    7.9999    8.9999    9.9999
    1.0000    2.0000    3.0000    4.0000    5.0000    6.0000    7.0000    8.0000    9.0000   10.0000
    2.0000    3.0000    4.0000    5.0000    6.0000    7.0000    8.0000    9.0000   10.0000   11.0000
    0.1831    2.0636    3.0704    4.0704    5.0704    6.0704    7.0704    8.0704    9.0704   10.0704
    0.5715    1.9999    2.9999    3.9999    4.9999    5.9999    6.9999    7.9999    8.9999    9.9999
    1.9999    2.9999    3.9999    4.9999    5.9999    6.9999    7.9999    8.9999    9.9999   10.9999

finite horizon policy:
     4     4     4     4     4     4     4     4     4     4
     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1
     2     2     2     2     2     2     2     2     2     2
     3     3     3     3     3     3     3     3     3     3
     3     3     3     3     3     3     3     3     3     3
     4     4     4     4     4     4     4     4     4     4
     4     4     4     4     4     4     4     4     4     4
     3     3     3     3     3     3     3     3     3     3
```