CS533-Assignment2-Report

Chunxiao Wang

1 Part I: Build a Planner

I build planner for finite-horizon MDP problems using dynamic programming for value function iteration. To test my algorithm, I use the example in the homework problem with 3 states and 2 actions as example. The non-stationary value function is

```
-1.95 -2.35 -2.53 -2.67 -2.83 -2.99 -3.14 -3.30 -3.46 -3.62
-1.20 -1.32 -1.46 -1.62 -1.78 -1.93 -2.09 -2.25 -2.41 -2.57
-0.10 -0.25 -0.41 -0.57 -0.72 -0.88 -1.04 -1.20 -1.35 -1.51,
```

and the non-stationary policy is

```
1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1,
```

with 0 indicating action 1 and 1 indicating action 2. The only difference compared with intuition is if there is 1 step to go, at state 1, we will take action 2 instead of action 1. Through simple computation, the value function will be -2 after taking action 1and -1.95 after taking action 2. So the result is right, there should be modification in the intuition expression.

2 Part II: Create Your Own MDP

2.1 MDP

We create a simple MDP for test purpose. The simple MDP has 20 states $1, ..., 20$ and 2 actions $0, 1$. At each state i, action 0 has equal probability 0.5 to go to $i + 1$ or $i - 1$(in 20, go to 19 or 1 and in 1, go to 20 or 2), action 1 has probability 1 to stay at the current state. Therefore the MDP states form a circle. For this circle, in the stochastic process scenario, action 0 leads to random walk and action 1 leads to absorption. The reward function is 0 at state $2, ..., 20$ and 1 at state 1. Therefore the goal is to go to state 20 and stay there as much as possible. Intuitively, if we are at state $2, ..., 20$, we will take action 0 to get the chance to go to state 1 and if we are in state 1, we will take action 1 to stay there.

2.2 Result

The non-stationary value function matrix for our MDP with column i giving value function with i steps-to-go is

```
2.00 3.00 4.00 5.00 6.00 7.00 8.00 9.00 10.00 11.00
0.50 1.00 1.62 2.25 2.94 3.62 4.35 5.08 5.83 6.59
0.00 0.25 0.50 0.88 1.25 1.70 2.16 2.66 3.17 3.72
0.00 0.00 0.12 0.25 0.47 0.69 0.98 1.27 1.61 1.95
0.00 0.00 0.00 0.06 0.12 0.25 0.38 0.55 0.73 0.96
0.00 0.00 0.00 0.00 0.03 0.06 0.13 0.20 0.31 0.42
0.00 0.00 0.00 0.00 0.00 0.02 0.03 0.07 0.11 0.17
0.00 0.00 0.00 0.00 0.00 0.00 0.01 0.02 0.04 0.06
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.01 0.02
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.01 0.02
0.00 0.00 0.00 0.00 0.00 0.00 0.01 0.02 0.04 0.06
0.00 0.00 0.00 0.00 0.00 0.02 0.03 0.07 0.11 0.17
0.00 0.00 0.00 0.00 0.03 0.06 0.13 0.20 0.31 0.42
0.00 0.00 0.00 0.06 0.12 0.25 0.38 0.55 0.73 0.96
0.00 0.00 0.12 0.25 0.47 0.69 0.98 1.27 1.61 1.95
0.00 0.25 0.50 0.88 1.25 1.70 2.16 2.66 3.17 3.72
0.50 1.00 1.62 2.25 2.94 3.62 4.35 5.08 5.83 6.59,
```

and the non-stationary policy matrix for our MDP with column i giving policy with i steps-to-go is,

```
1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
```

```
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
```

2.3 Analysis

From the result, we could see action 0 is taken when the state is in $2, ..., 20$ and action 1 is taken when the state is 1., which matches our intuition. And also when the state is closer to 1, the value function is larger and when the state is far from 1, the value function is small.

3 Part III: More Testing

For both test MDPs, there are 10 states and 4 actions. For $H = 10$, the output includes two 10×10 matrices, one is value function matrix with column i giving value function with i steps-to-go and the other one is policy matrix with column i giving policy with i steps-to-go. In the policy, 0 for action 1, 1 for action 2, 2 for action 3 and 3 for action 4
For the first MDP, the value function matrix is

```
1.00 1.00 1.03 2.00 2.00 2.03 3.00 3.00 3.03 4.00
0.00 0.89 1.00 1.02 1.89 2.00 2.02 2.89 3.00 3.02
0.01 0.80 0.90 1.01 1.80 1.90 2.01 2.80 2.90 3.01
0.00 0.89 0.90 1.02 1.89 1.90 2.02 2.89 2.90 3.02
1.00 1.03 2.00 2.00 2.03 3.00 3.00 3.03 4.00 4.00
0.00 0.66 0.89 1.00 1.65 1.89 2.00 2.65 2.89 3.00
0.66 0.88 0.98 1.65 1.88 1.98 2.65 2.88 2.98 3.65
0.00 0.85 0.90 1.01 1.85 1.90 2.01 2.85 2.90 3.01
0.03 1.00 1.00 1.03 2.00 2.00 2.03 3.00 3.00 3.03
0.89 0.90 1.02 1.89 1.90 2.02 2.89 2.90 3.02 3.89,
```

the policy matrix is

```
3 3 3 3 3 3 3 3 3 3
0 3 1 3 3 1 3 3 1 3
2 2 1 2 2 1 2 2 1 2
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 2 2 0 2 2 0 2 2
1 1 1 1 1 1 1 1 1 1
0 2 2 2 2 2 2 2 2 2
3 1 1 1 1 1 1 1 1 1
3 0 3 3 0 3 3 0 3 3.
```

For the second MDP, the value function matrix is

```
0.06 1.00 1.06 2.00 2.06 3.00 3.06 4.00 4.06 5.00
0.00 0.99 0.99 1.99 2.00 2.99 3.00 3.99 4.00 4.99
```

```
1.25 1.60 2.30 2.62 3.31 3.62 4.31 4.62 5.31 5.62
0.00 0.07 0.99 1.04 1.99 2.04 2.99 3.04 3.99 4.04
0.57 1.00 1.57 2.00 2.57 3.00 3.57 4.00 4.57 5.00
1.00 1.00 2.00 2.00 3.00 3.00 4.00 4.00 5.00 5.00
1.00 2.00 2.00 3.00 3.00 4.00 4.00 5.00 5.00 6.00
0.08 0.08 0.97 1.08 1.96 2.08 2.96 3.08 3.96 4.08
0.01 0.57 1.00 1.57 2.00 2.57 3.00 3.57 4.00 4.57
1.00 1.00 2.00 2.00 3.00 3.00 4.00 4.00 5.00 5.00,
```

the policy matrix is

```
3 0 3 0 3 0 3 0 3 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
1 2 1 1 1 1 1 1 1 1
2 1 2 1 2 1 2 1 2 1
2 0 2 0 2 0 2 0 2 0
0 2 2 2 2 2 2 2 2 2
3 1 1 1 1 1 1 1 1 1
3 1 0 1 0 1 0 1 0 1
2 2 2 2 2 2 2 2 2 2.
```