MDL Assignment 2
Part 1

Submitted by
T. H. ARJUN
CSD
2019111012

Parameters

γ (Discount factor) = 0.20
ϵ (Bellman error) = 0.01
R (Reward) = Arr [(2019111012) % 15] = Arr [12]
R = 18

Questions

1) Write the Transition Table

Current State	Action	Next State	Probability	Reward.
A	Move Right	B	0.8	−1
A	Move Right	A	0.2	−1
A	Move Up	C	0.8	−1
A	Move Up	A	0.2	−1
B	Move left	A	0.8	−1
B	Move left	B	0.2	−1
B	Move Up	R	0.8	−4
B	Move Up	B	0.2	−1
C	Move Right	R	0.25	−3
C	Move Right	C	0.75	−1
C	Move Down	A	0.8	−1
C	Move Down	C	0.2	−1

2) According to me the best option will be to Take the following steps :-

> At square A, Move Right and then at B move up. If you end up at C Move Right

The reason for these choices is the discount factor γ. Here $\gamma = 0.2$. So each time our rewards in future become 2/10th. So there is high depreciation. So we care less about the future and care more about the current choice. Hence at A Moving Right is most rewarding as it has high chances of Reaching B from where there is high chance of reaching Terminal State. At B moving Up is the best Policy as we reach terminal state of high reward with high probability And at C Move Right as we have a high reward waiting and we care more about current reward due to low value of γ (discount factor

3) Value Iteration

Expectation of Rewards :-

$R(A, \text{Move Right}) = 0.8 \times -1 + 0.2 \times -1 = -1$

$R(A, \text{Move Up}) = 0.8 \times -1 + 0.2 \times -1 = -1$

$R(B, \text{Move Left}) = 0.8 \times -1 + 0.2 \times -1 = -1$

$R(B, \text{Move Up}) = 0.8 \times -4 + 0.2 \times -1 = -3.4$

$R(C, \text{Move Right}) = 0.25 \times -3 + 0.75 \times -1 = -1.5$

$R(C, \text{Move Down}) = 0.8 \times -1 + 0.2 \times -1 = -1$

Given,

$R = 18$

$\gamma = 0.20$

$\epsilon = 0.01$

Value Iteration Algorithm.

Initialize $U_0(I) = 0$

Iterate :

$$U_{t+1}(I) = \max_A \left[R(I,A) + \gamma \sum_j P(J \mid I, A) * U_t(J) \right]$$

Until $\max(|U_{t+1} - U_t|) < \epsilon$

So in our case the updates become following

State	U_t	U_{t+1}
A	a	$\max\{-1 + 0.2(0.8b + 0.2a),\ -1 + 0.2(0.8c + 0.2a)\}$
B	b	$\max\{-1 + 0.2(0.8a + 0.2b),\ -3.4 + 0.2(0.8r + 0.2b)\}$
C	c	$\max\{-1.5 + 0.2(0.25r + 0.75c),\ -1 + 0.2(0.8a + 0.2c)\}$
R	r	r

Iteration 1

State	U_1	U_2 expression	U_2
A	0	$\max\{-1+0.2\times(0.8\times0+0.2\times0),\ -1+0.2\times(0.8\times0+0.2\times0)\}$	−1
B	0	$\max\{-1+0.2\times(0.8\times0+0.2\times0),\ -3.4+0.2\times(0.8\times18+0.2\times0)\}$	−0.52
C	0	$\max\{-1.5+0.2\times(0.25\times18+0.75\times0),\ -1+0.2\times(0.8\times0+0.2\times0)\}$	−0.6
R	18	18	

$$\max_{\in S}(|U_2-U_1|) = 1$$

−0.6 C	18 R
−1 A	−0.52 B

Iteration 2

State	U_2	U_3 Expression	U_3
A	−1	$\max\{-1+0.2(0.8\times-0.52+0.2\times-1),\ -1+0.2\times(0.8\times0.6+0.2\times-1)\}$	−1.12
B	−0.52	$\max\{-1+0.2(0.8\times-1+0.2\times-0.52),\ -3.4+0.2\times(0.8\times18+0.2\times-0.52)\}$	−0.54
C	−0.6	$\max\{-1.5+0.2(0.25\times18+0.75\times-0.6),\ -1+0.2(0.8\times-1+0.2\times-0.6)\}$	−0.69
R	18	18	

$$\max_{\in S}(U_3-U_2) = 0.123$$

−0.69 C	18 R
−1.12 A	−0.54 B

Iteration 3

State	U_3	U_4 Expression	U_4
A	-1.12	$\max\{-1+0.2(0.8\times -0.54 + 0.2\times -1.12),$ $-1+0.2(0.8\times$ [illegible] $+0.2\times$ [illegible] $)\}$	-1.131
B	-0.54	$\max\{-1+0.2(0.8\times -1.12+0.2\times -0.54),$ $-3.4+0.2(0.8\times 18+0.2\times -0.54)\}$	-0.541
C	-0.69	$\max\{-1.5+0.2(0.25\times 18+0.75\times -0.69),$ $-1+0.2(0.8\times -1.12+0.2\times -0.69)\}$	-0.703
R	18	18	

$$\max_{\in \text{States}} |U_4 - U_3| = 0.0135$$

-0.703 C	18 R
-1.1314 A	-0.541 B

Iteration 4

State	U_4	U_5 Expression	U_5
A	-1.13	$\max\{-1+0.2\times(0.8\times -0.54+0.2\times -1.13),$ $-1+0.2\times(0.8\times -0.7+0.2\times -1.13)\}$	-1.131
B	-0.54	$\max\{-1+0.2\times(0.8\times -1.13+0.2\times -0.54),$ $-3.4+0.2\times(0.8\times 18+0.2\times -0.54)\}$	-0.5416
C	-0.7	$\max\{-1.5+0.2\times(0.25\times 18+0.75\times -0.7),$ $-1+0.2\times(0.8\times -1.131+0.2\times -0.7)\}$	-0.705
R	18	18	

$$\max_{\in \text{States}} |U_5 - U_4| = 0.002$$

-0.70 C	18 R
-1.1319 A	-0.5416 B

4) For finding the optimal path we need to find the optimal policy at each state. To find the optimal policy we need to find utility of each state, action pair available and choose best according to Maximum expected utility principle i.e choosing

$$\pi^*(S(I)) = \underset{action}{\operatorname{argmax}} \left(\sum P_{Ij}^{action} \times U(S(J)) \right)$$

$U(A, \text{Move Right}) = 0.8 \times -0.54 + 0.2 \times -1.13 = -0.658$

$U(A, \text{Move Up}) = 0.8 \times -0.7 + 0.2 \times -1.13 = -0.786$

So Best Policy at A is
Move Right

$U(B, \text{Move left}) = 0.8 \times -1.13 + 0.2 \times -0.54 = -1.012$

$U'(B, \text{Move Up}) = 0.8 \times 18 + 0.2 \times -0.54 = 3.6$

So Best Policy at B is
Move Up

$U(C, \text{Move Right}) = 0.25 \times 18 + 0.75 \times -0.7 = 3.975$

$U(C, \text{Move Down}) = 0.8 \times -1.13 + 0.2 \times -0.7 = -1.044$

So Best Policy at C is
Move Right

So Best path is

A → Move Right
B → move Up
C → Move Right.

⇒ My Initial guess is correct.

5) The two states from where reaching terminal state is possible is B & C due to high values of Reward/Utility for them and high depreciation of future rewards due to low gamma (discount factor) we care more about current reward and hence it is best in our interest to Move Up at B and Move Right at C. From A it is better to move to B by choosing Move Right as we have high probability of reaching B from where we can reach R with high probability and reward. So the things that sets the trend here are the high rewards/Utilities of some transitions They weigh in so much that it is very clear what the best policy is. Changing the high value or moving them around will change the best policy.