Artificial Intelligence

Assignment 2 Report

Akhil Singh (20171210)

Faizan Farooq(20171209)

Part B.)

1 a.)

Step cost = -3.5, discount factor = 0.1.

Final utilities :-

3.5000 0.0000 2.4729 35.0000

-0.0742 -0.3630 -0.1310 2.4732

-0.3630 -0.3857 0.0000 -0.1552

-0.3851 -0.3885 -7.0000 -0.3832

o o E o

N W E N

N E o N

W W o E

(T marks terminal states)

Iterations till convergence = 2.

The algorithm converges rapidly, and the final utilities are all mostly negative. This is because our discount factor is much less, and as such, the farther rewards are given much less favor. So, the immediate negative step cost dominates the utility values.

1 b.)

Step cost = -3.5, discount factor = 0.99

Final Utilities:-

3.5000 0.0000 29.7277 35.0000

13.9850 20.2783 25.5520 29.7277

10.5363 15.1371 0.0000 25.0366

6.3453 8.4588 -7.0000 17.3929

Final Policy :-

o o E o

E E E N

E N o N

N N o N

Iterations till convergence = 16

Since the discount factor ~1, there is not much decay in the rewards of the farther states. As such, the policy always tends to direct us to the goal state at (0,3), as it has a very high value.

2 a.)

Step Cost = 35, discount factor = 0.99

Final utilities :-

-3.5000 0.0000 3464.9950 35.0000

3464.9954 3464.9955 3464.9955 3464.9954

3464.9955 3464.9955 0.0000 3464.9954

3464.9955 3464.9956 -7.0000 3464.9950

Final Policy :-

o o W o

S S W S

E W o N

N W o E

Iterations till convergence = 691

Since the step cost is very large, larger than the reward of the best goal state, the utilities are skewed and the policy sends us in circles. The cell at (1,1) has the highest utility, so it assumes that we cannot move from there.

2 b.) Step cost = -7, discount factor = 0.99

Final utilities :-

3.5000 0.0000 24.9391 35.0000

-2.0631 6.9324 16.9708 24.9391

-9.7988 -2.6743 0.0000 15.9872

-17.5663 -11.4801 -7.0000 5.5925

Final policy :-

o o E o

E E E N

N N o N

N N o N

Iterations till convergence = 15

Here, the negative factor of the step cost initially outweighs all positive benefits, as they occur much farther away. However, as we approach the goal state at (0,3), the utilities become larger as it has a very high reward. So, the policy directs us towards that state.

2 c.)

Step cost = -8.75, discount factor = 0.99

Final utilities:-

3.5000 0.0000 22.5448 35.0000

-6.5042 0.3050 12.6802 22.5448

-16.5586 -11.1656 0.0000 11.4625

-26.0370 -16.9943 -7.0000 -0.3076

Final Policy :-

o o E o

N E E N

N N o N

N E o N

Iterations till convergence = 13

2 d.)

Step cost = -35, discount factor = 0.99

Final utilities :-

3.5000 0.0000 -13.3698 35.0000

-45.1339 -88.7640 -51.6787 -13.3698

-88.7640 -96.7707 0.0000 -56.4076

-96.7707 -55.2434 -7.0000 -50.8084

Final Policy :-

o o E o

N W E N

N S o N

E E o W

Iterations till convergence = 13

Detailed analysis on Q1 below:

INPUT MATRIX

0.0 0.0 0.0 35.0

3.5 0.0 0.0 0.0

0.0 0.0 0.0 0.0

0.0 0.0 -7.0 0.0

Step Reward = -X/10 = -3.5

Wall Position: 0 1 , 2 2

End Positions: 0 0 , 0 3 , 3 2

Start Position: 3, 0

In this test case, we substituted x = 35 corresponding to our team number. When we run with the value iteration algorithm, we got total 16 iterations before our terminal case is reached. In the shown iterations below, we have also given the paths adopted to reach the terminal state.

Iteration: 1

3.5000 0.0000 24.2550 35.0000

-0.6930 -3.4650 15.4019 25.7798

-3.4650 -3.8080 0.0000 16.9526

-3.4650 -4.5010 -7.0000 9.2685

o o E o

N E N N

S S o N

S S o N

Since the values are being updated from the top-left corner, the states acquire positive values which cascade to the neighboring states that are remaining to be calculated. However, there is a negative value attached with some states which are close to the bottom-right positive sink and there is a change in the direction of moment for some of the grid-positions with just one-step reward check. But, since they were calculated late, their positive value doesn't affect other grid-positions.

Iteration: 2

3.5000 0.0000 28.1810 35.0000

-1.1046 8.0133 22.1999 29.0050

-5.0599 2.0036 0.0000 22.8636

-6.9979 -3.2639 -7.0000 14.8675

o o E o

N E N N

N N o N

S N o N

In this iteration, the policy for the position (marked in red) changes because it analyzes that it gets more reward on going in the new direction rather than the old one. They clearly prefer the positive reward.

Iteration: 3

3.5000 0.0000 29.2427 35.0000

2.7271 15.1090 24.5998 29.5619

-1.6077 8.5405 0.0000 24.4750

-5.7542 2.0364 -7.0000 16.6981

o o E o

E E E N

N N o N

N N o N

The policy at two positions changes from moving WEST and NORTH to moving EAST as it analyzes that moving East is more rewarding. This is because the positive-sink affects the grid-positions at this level and they prefer to move towards the positive sink.

Iteration: 4

3.5000 0.0000 29.5854 35.0000

8.6887 18.3593 25.3123 29.6875

4.1028 12.3273 0.0000 24.8936

-0.5837 5.5474 -7.0000 17.2108

o o E o

E E E N

N N o N

N N o N

In this iteration, even the states close to the negative sink start getting less negative and all the states point towards the positive-sink.

Iteration: 5

3.5000 0.0000 29.6899 35.0000

11.8283 19.6203 25.4928 29.7178

7.5295 14.0401 0.0000 25.0005

2.9898 7.2578 -7.0000 17.3462

o o E o

E E E N

N N o N

N N o N

Iteration: 6

3.5000 0.0000 29.7181 35.0000

13.1662 20.0576 25.5374 29.7253

9.2542 14.7268 0.0000 25.0275

4.8789 7.9886 -7.0000 17.3811

o o E o

E E E N

E N o N

N N o N

Iteration: 7

3.5000 0.0000 29.7253 35.0000

13.6833 20.2043 25.5484 29.7271

10.0363 14.9883 0.0000 25.0343

5.7576 8.2828 -7.0000 17.3899

o o E o

E E E N

E N o N

N N o N

Iteration: 8

3.5000 0.0000 29.7271 35.0000

13.8769 20.2534 25.5511 29.7275

10.3496 15.0842 0.0000 25.0360

6.1219 8.3947 -7.0000 17.3921

o o E o

E E E N

E N o N

N N o N

Iteration: 9

3.5000 0.0000 29.7275 35.0000

13.9468 20.2699 25.5518 29.7277

10.4685 15.1185 0.0000 25.0364

6.2632 8.4359 -7.0000 17.3927

o o E o

E E E N

E N o N

N N o N

Iteration: 10

3.5000 0.0000 29.7277 35.0000

13.9717 20.2755 25.5520 29.7277

10.5121 15.1306 0.0000 25.0365

6.3158 8.4507 -7.0000 17.3928

o o E o

E E E N

E N o N

N N o N

Iteration: 11

3.5000 0.0000 29.7277 35.0000

13.9804 20.2774 25.5520 29.7277

10.5278 15.1349 0.0000 25.0366

6.3349 8.4560 -7.0000 17.3928

o o E o

E E E N

E N o N

N N o N

Iteration: 12

3.5000 0.0000 29.7277 35.0000

13.9834 20.2780 25.5520 29.7277

10.5333 15.1363 0.0000 25.0366

6.3417 8.4578 -7.0000 17.3929

o o E o

E E E N

E N o N

N N o N

Iteration: 13

3.5000 0.0000 29.7277 35.0000

13.9845 20.2782 25.5520 29.7277

10.5353 15.1368 0.0000 25.0366

6.3441 8.4584 -7.0000 17.3929

o o E o

E E E N

E N o N

N N o N

Iteration: 14

0.0000 0.0000 29.7277 35.0000

17.4272 20.3135 25.5520 29.7277

13.1716 15.4576 0.0000 25.0366

8.7155 8.9473 -7.0000 17.3929

Move Matrix :

o o E o

E E E N

N N o N

N N o N

Iteration: 15

0.0000 0.0000 29.7277 35.0000

17.4273 20.3135 25.5520 29.7277

13.1717 15.4576 0.0000 25.0366

8.7156 8.9473 -7.0000 17.3929

Move Matrix :

o o E o

E E E N

N N o N

N N o N

Iteration: 16

0.0000 0.0000 29.7277 35.0000

17.4273 20.3135 25.5520 29.7277

13.1717 15.4576 0.0000 25.0366

8.7157 8.9473 -7.0000 17.3929

Move Matrix:

o o E o

E E E N

N N o N

N N o N