

209AS Lab 1

Alexie Pogue

October 15, 2019

1 $P_e = 0$

1.1 Value Iteration

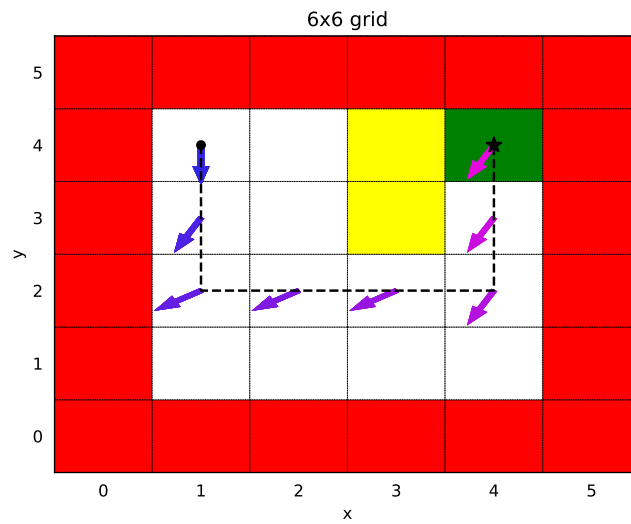


Figure 1: Value iteration trajectory, collection of R until convergence.

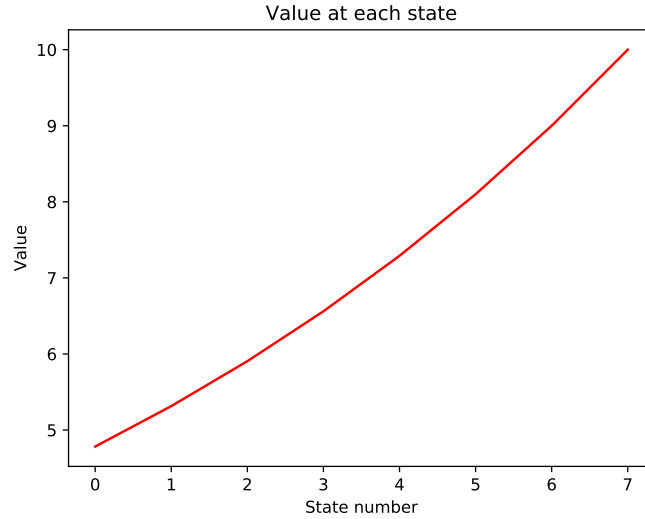


Figure 2: Value iteration values, collection of R until convergence.

The trajectory states are:

[(1, 4, 6), (1, 3, 7), (1, 2, 8), (2, 2, 8), (3, 2, 8), (4, 2, 7),
(4, 3, 7), (4, 4, 7)]

Respective values are:

```
[array([4.78296235]), array([5.31440261]), array([5.90489179]),  
array([6.56099179]), array([7.28999179]), array([8.09999179]),  
array([8.99999179]), array([9.99999179])]
```

Compute time:

```
value iteration timer 73.2994408607
```

If the Reward is changed to 10 for the green square, and the robots exit upon reaching the goal, then the corresponding data is:

The trajectory states are:

[(1, 4, 6), (1, 3, 7), (1, 2, 8), (2, 2, 8), (3, 2, 8), (4, 2, 7),
(4, 3, 7), (4, 4, 7)]

Respective values are:

```
[array([4.782969]), array([5.31441]), array([5.9049]),  
array([6.561]), array([7.29]), array([8.1]),  
array([9.]), array([10.])]
```

Compute time:

```
value iteration timer 5.96084094048
```

1.2 Policy Iteration

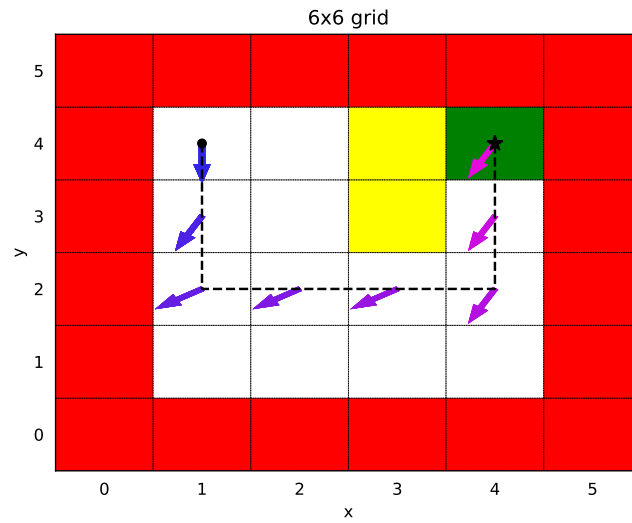


Figure 3: Policy iteration trajectory, collection of R until convergence.

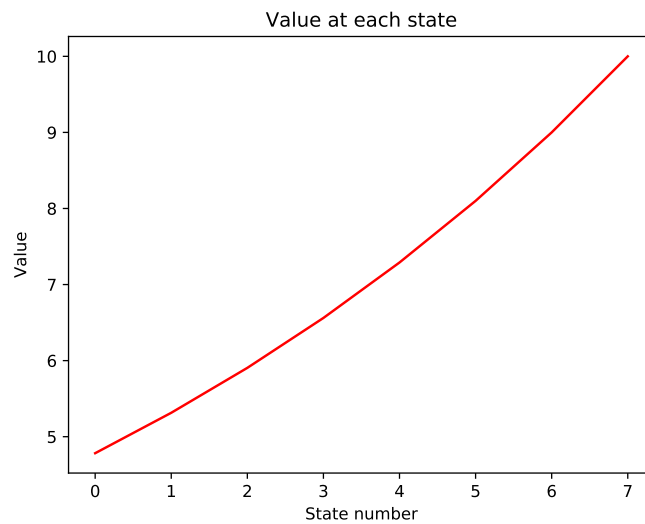


Figure 4: Policy iteration values, collection of R until convergence.

The trajectory states are:

[(1, 4, 6), (1, 3, 7), (1, 2, 8), (2, 2, 8), (3, 2, 8), (4, 2, 7),
(4, 3, 7), (4, 4, 7)]

Respective values are:

[array([4.78296235]), array([5.31440261]), array([5.90489179]),
array([6.56099179]), array([7.28999179]), array([8.09999179]),
array([8.99999179]), array([9.99999179])]

Compute time:

policy iteration timer 110.596468925

If the Reward is changed to 10 for the green square, and the robots exit
upon reaching the goal, then the corresponding data is:

The trajectory states are:

[(1, 4, 6), (1, 3, 7), (1, 2, 8), (2, 2, 8), (3, 2, 8), (4, 2, 7),
(4, 3, 7), (4, 4, 7)]

Respective values are:

[array([4.782969]), array([5.31441]), array([5.9049]),
array([6.561]), array([7.29]), array([8.1]),
array([9.]), array([10.])]

Compute time:

policy iteration timer 20.302341938

2 $P_e = .1$, Robot points down at goal (4,4,6)

2.1 Value Iteration

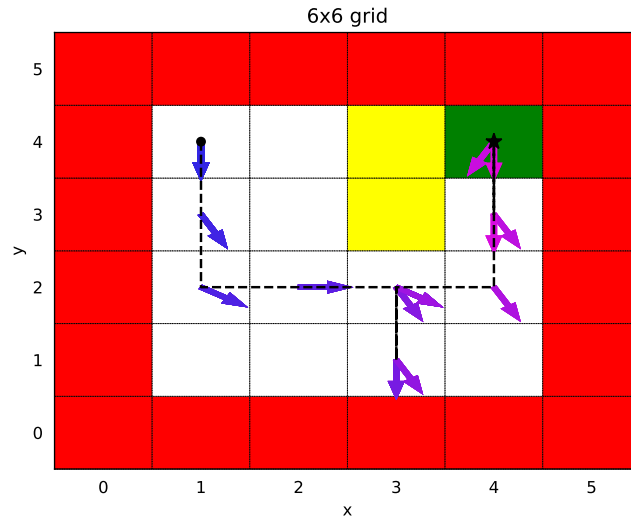


Figure 5: Value iteration trajectory, collection of R until convergence.

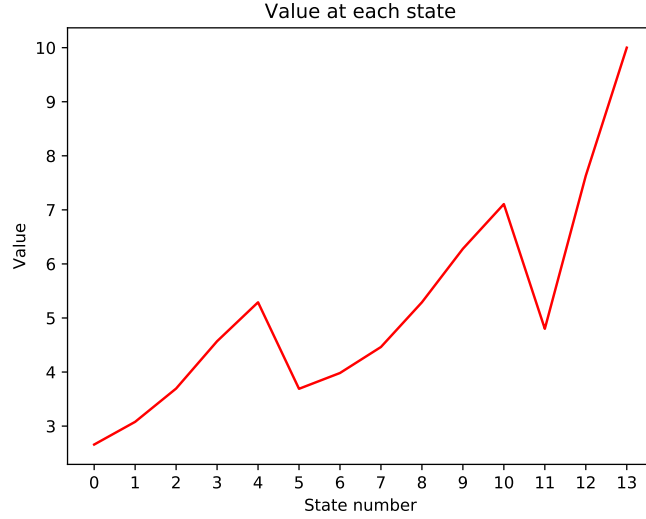


Figure 6: Value iteration values, collection of R until convergence.

The trajectory states are:

[(1, 4, 6), (1, 3, 5), (1, 2, 4), (2, 2, 3), (3, 2, 4), (3, 1, 6),
 (3, 2, 5), (3, 1, 5), (3, 2, 4), (4, 2, 5), (4, 3, 5),
 (4, 4, 7), (4, 3, 6), (4, 4, 6)]

Respective values are:

[array([2.65875227]), array([3.07896706]), array([3.69456097]),
 array([4.5696727]), array([5.29033789]), array([3.69002005]),
 array([3.98232236]), array([4.4633886]), array([5.29033789]),
 array([6.28007767]), array([7.10704236]), array([4.79975875]),
 array([7.63197166]), array([9.9999179])]

Compute time:

value iteration timer 71.6555588245

If the Reward is changed to 10 for the green square, and the robots exit upon reaching the goal, then the corresponding data is:

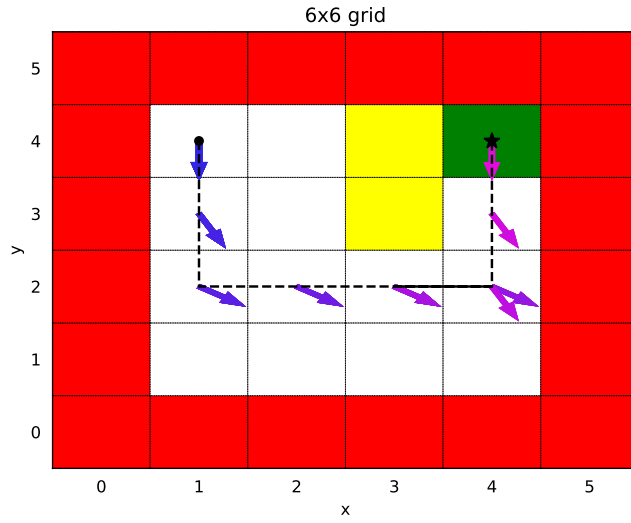


Figure 7: Value iteration trajectory, exit at goal.

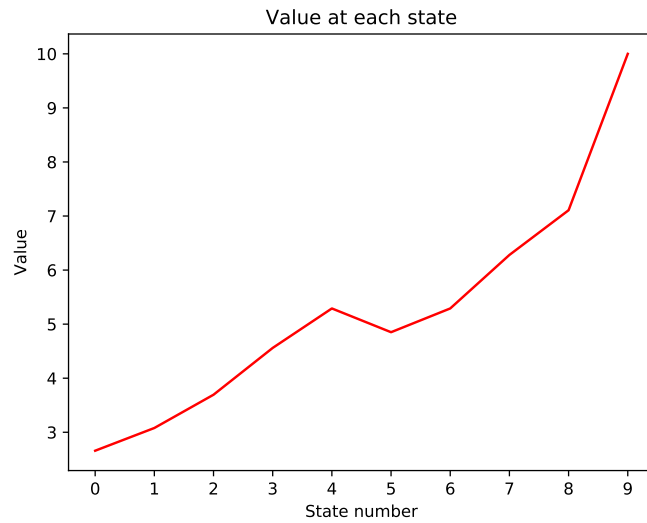


Figure 8: Value iteration values, exit at goal.

The trajectory states are:

$[(1, 4, 6), (1, 3, 5), (1, 2, 4), (2, 2, 4), (3, 2, 4), (4, 2, 4), (3, 2, 4), (4, 2, 5), (4, 3, 5), (4, 4, 6)]$

Respective values are:

```
[array([2.65875724]), array([3.07897261]), array([3.69456737]),  
array([4.56013359]), array([5.29034517]), array([4.85090475]),  
array([5.29034517]), array([6.28008522]), array([7.10705009]),  
array([10.])]
```

Compute time:

value iteration timer 21.66061306

2.2 Policy Iteration

Policy iteration doesn't always converge.

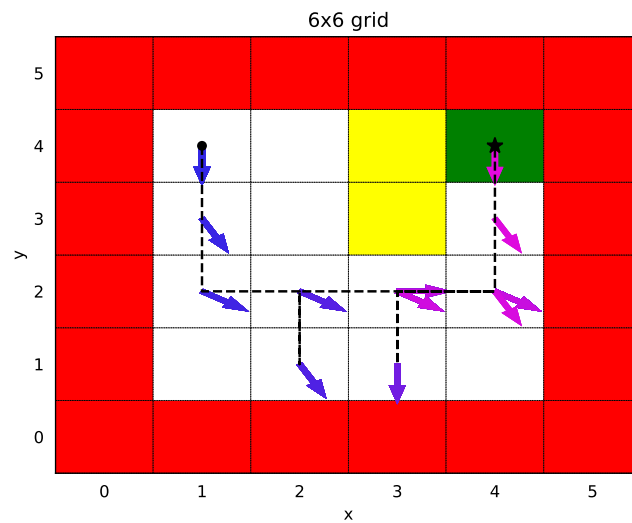


Figure 9: Policy iteration trajectory, collection of R until convergence.

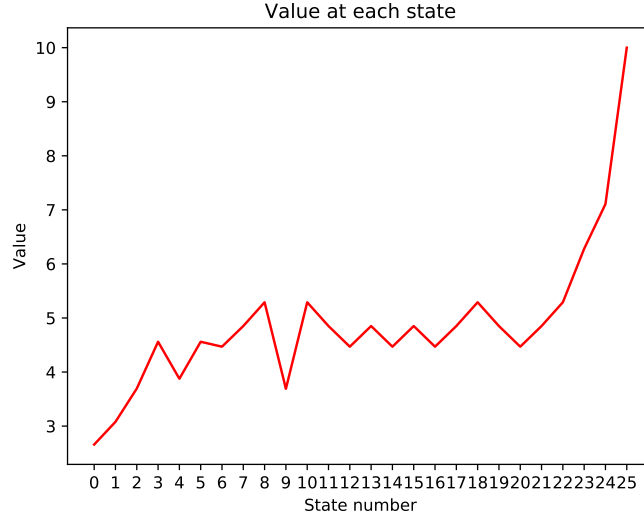


Figure 10: Policy iteration values, collection of R until convergence.

The trajectory states are:

```
[(1, 4, 6), (1, 3, 5), (1, 2, 4), (2, 2, 4), (2, 1, 5), (2, 2, 4),
(3, 2, 3), (4, 2, 4), (3, 2, 4), (3, 1, 6), (3, 2, 4), (4, 2, 4),
(3, 2, 3), (4, 2, 4), (3, 2, 3), (4, 2, 4), (3, 2, 3), (4, 2, 4),
(3, 2, 4), (4, 2, 4), (3, 2, 3), (4, 2, 4), (3, 2, 4), (4, 2, 5),
(4, 3, 5), (4, 4, 6)]
```

Respective values are:

```
[array([2.65875227]), array([3.07896706]), array([3.69456097]),
array([4.56012648]), array([3.87625252]), array([4.56012648]),
array([4.46912359]), array([4.85089814]), array([5.29033789]),
array([3.69002005]), array([5.29033789]), array([4.85089814]),
array([4.46912359]), array([4.85089814]), array([4.46912359]),
array([4.85089814]), array([4.46912359]), array([4.85089814]),
array([5.29033789]), array([4.85089814]), array([4.46912359]),
array([4.85089814]), array([5.29033789]), array([6.28007767]),
array([7.10704236]), array([9.99999179])]
```

Compute time:

```
policy iteration timer 103.498069048
```

If the Reward is changed to 10 for the green square, and the robots exit upon reaching the goal, then the corresponding data is:

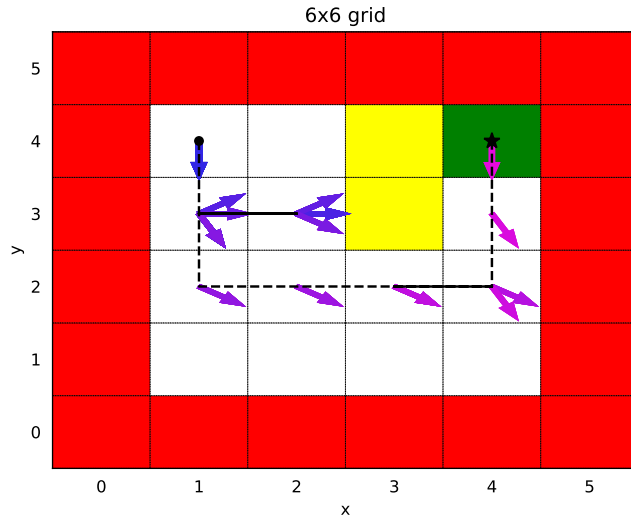


Figure 11: Policy iteration trajectory, exit at goal.

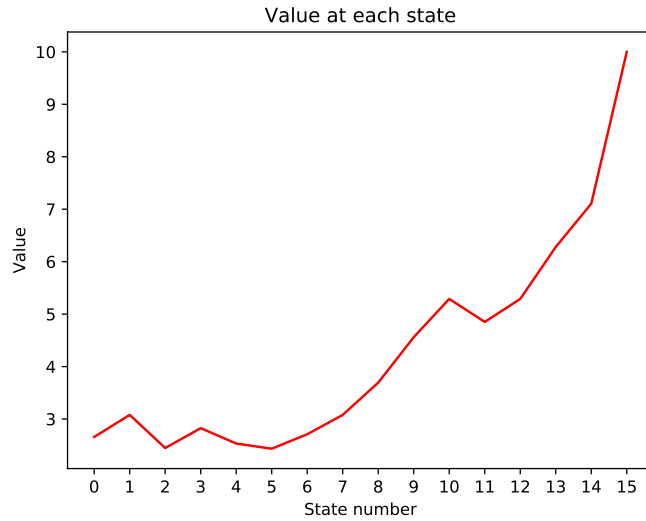


Figure 12: Policy iteration values, exit at goal.

The trajectory states are:

$[(1, 4, 6), (1, 3, 5), (2, 3, 3), (1, 3, 2), (2, 3, 2), (1, 3, 3), (2, 3, 4), (1, 3, 5), (1, 2, 4), (2, 2, 4), (3, 2, 4), (4, 2, 4),$

(3, 2, 4), (4, 2, 5), (4, 3, 5), (4, 4, 6)]

Respective values are:

```
[array([2.65875724]), array([3.07897261]), array([2.44709846]),  
array([2.82615543]), array([2.53250241]), array([2.43417618]),  
array([2.70954544]), array([3.07897261]), array([3.69456737]),  
array([4.56013359]), array([5.29034517]), array([4.85090475]),  
array([5.29034517]), array([6.28008522]), array([7.10705009]),  
array([10.] )]
```

Compute time:

policy iteration timer 43.4269220829