CP8319/CPS824 Reinforcement Learning

Winter 2021

Assignment 1

Rohaan Ahmed
PhD Student - Computer Science
rohaan.ahmed@ryerson.ca

February 9, 2021

Ryerson University

Instructor: Dr. Nariman Farsad
February 9, 2021

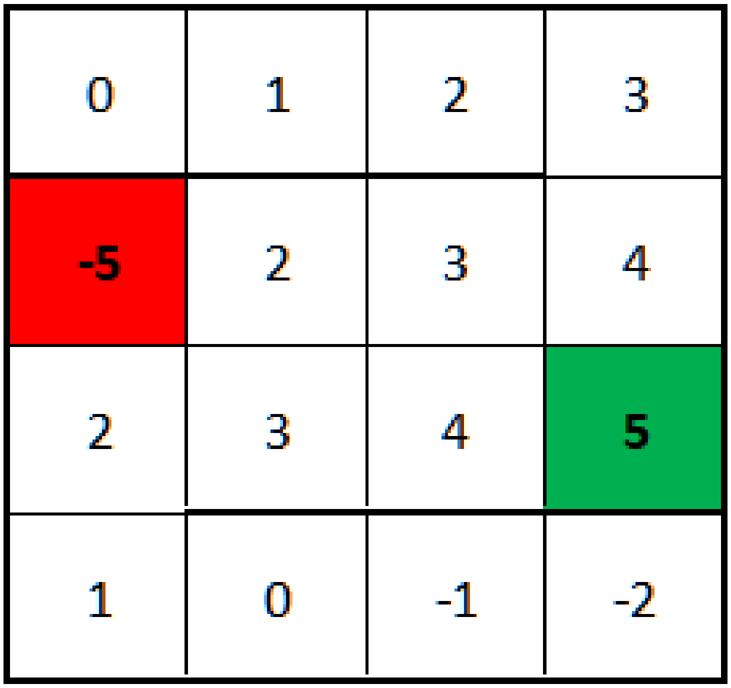
1. MDP Grid World

1.a.

The r_s that leads to the optimal solution is:

$r_s = -1$

Resulting in the following grid, where the values inside each square represent the optimal value for that square:



0	1	2	3
-5	2	3	4
2	3	4	5
1	0	-1	-2

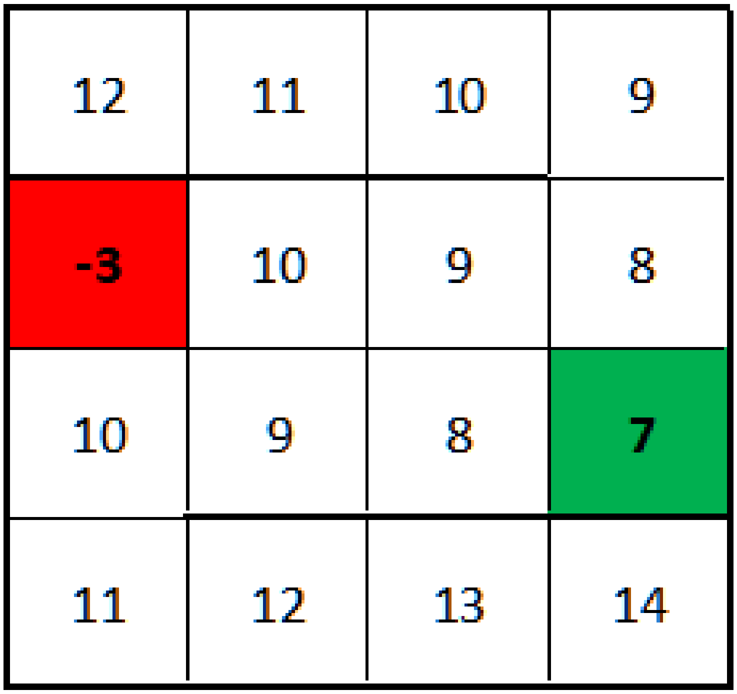
Figure 1:

1.b.

The new values for r_s, r_g, and r_r will be:

$r_s = 1$
$r_g = 7$
$r_r = -3$

Resulting in the following grid, where the values inside each square represent the optimal value for that square:



12	11	10	9
-3	10	9	8
10	9	8	7
11	12	13	14

Figure 2:

1.c.

The old value function is given by:

$$V^{\pi}_{old}(s) = E_{\pi}\left[\sum_{T=0}^{\infty} \gamma^T r_{(t+T)} | s_t = s\right] \tag{0.1}$$

The new value function is given by:

$$V^{\pi}_{new}(s) = E_{\pi}\left[\sum_{T=0}^{\infty} \gamma^T (r_{(t+T)} + c) | s_t = s\right] \tag{0.2}$$

$$\implies V^{\pi}_{new}(s) = E_{\pi}\left[\sum_{T=0}^{\infty} \gamma^T (r_{(t+T)}) | s_t = s\right] + c\sum_{T=0}^{\infty} \gamma^T \tag{0.3}$$

The new value function can then be written in terms of the old value function as:

$$V^{\pi}_{new}(s) = V^{\pi}_{old}(s) + \frac{c}{1-\gamma} \tag{0.4}$$

1.d.

For $r_s = +2$, $c = 3$. Thus, at the start:

$r_r = -2$

$r_g = 8$
$\gamma = 1$

Since the discount factpor is 1, the policy will continue to explore the grid indefinitely without convergence, resulting in infinite reward for each unshaded square and the target square.

1.e.

In general, if we set $0 < \gamma < 1$, there will be some threshold values $\gamma_{t,upper}$ and $\gamma_{t,lower}$ such that:

- For $\gamma \geq \gamma_{t,upper}$, the policy will continue to explore indefinitely. This is because we are still placing far greater importance on future rewards than the immediate reward. It is expected that for this particular problem, $\gamma_{t,upper} \approx 1$.
- For $\gamma_{t,lower} < \gamma < \gamma_{t,upper}$, the policy will converge to the optimal policy, i.e., finding the shortest path to the green target square.
- For $\gamma \leq \gamma_{t,lower}$, the policy will act like a "greedy policy", resulting in a sub-optimal policy. This is beacause we are incentivizing the agent to value immediate rewards much more than future rewards. It is expected that for this particular problem, $\gamma_{t,lower} \approx 0$.

1.f.

If we want the optimal policy to terminate at the red square, we must make is so that the reward in the red square is greater than the reward in its surrounding squares. Thus:

$r_s \leq -5$

2.c.

The stochastic case takes longer to find the optimal policy than the deterministic case. This is expected because there is an element of randomness in how the action affects the state in the stochastic case. When the agent selects an action to perform, it may not always result in the expected environmental state change.

Although stochasticity typically results in short-term unpredictability and slower convergence, it is not always a bad thing. Accounting for randomness in the environment's dynamics leads to many positive outcomes, including:

1. Better learning in dynamic environments
2. Greater exploration
3. Better representation of real-world scenarios