Reinforcement Learning - Week 2

Antonio Chiappetta

October 2019

1 Sailor gridworld

1.1 Question 1

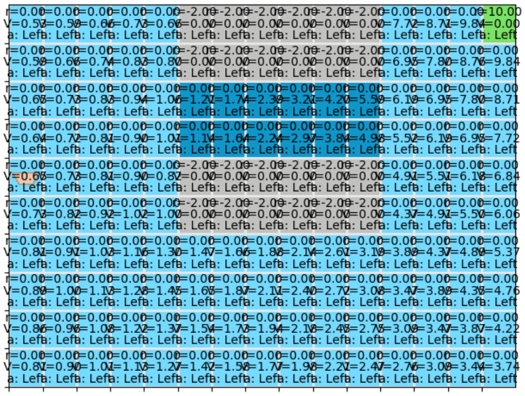
The agent is the sailor. He is the one that has to make decisions to perform actions, taking into consideration the environment around him and the goal he wants to reach, i.e., the harbour.

The environment is everything that is outside of the agent's absolute control: the sea, the rocks, the wind, the harbour, and the way these elements influence the outcome of the actions taken by the sailor.

2 Value iteration

2.1 Task 1

The following image shows the final render after 100 iterations of the value iteration algorithm. In this image only state values are computed, not the policy.



r=0.00 V=0.57 a: Left	r=0.00 V=0.59 a: Left	r=0.00 V=0.66 a: Left	r=0.00 V=0.73 a: Left	r=0.00 V=0.66 a: Left	r=-2.00 V=0.00 a: Left	r=-2.00 V=0.00 a: Left	r=-2.00 V=0.00 a: Left	r=-2.00 V=0.00 a: Left	r=-2.00 V=0.00 a: Left	r=-2.00 V=0.00 a: Left	r=0.00 V=7.72 a: Left	r=0.00 V=8.71 a: Left	r=0.00 V=9.84 a: Left	r=10.00 V=0.00 a: Left
r=0.00 V=0.59 a: Left	r=0.00 V=0.66 a: Left	r=0.00 V=0.74 a: Left	r=0.00 V=0.83 a: Left	r=0.00 V=0.80 a: Left	r=-2.00 V=0.00 a: Left	r=-2.00 V=0.00 a: Left	r=-2.00 V=0.00 a: Left	r=-2.00 V=0.00 a: Left	r=-2.00 V=0.00 a: Left	r=-2.00 V=0.00 a: Left	r=0.00 V=6.95 a: Left	r=0.00 V=7.80 a: Left	r=0.00 V=8.76 a: Left	r=0.00 V=9.84 a: Left
r=0.00 V=0.65 a: Left	r=0.00 V=0.73 a: Left	r=0.00 V=0.83 a: Left	r=0.00 V=0.94 a: Left	r=0.00 V=1.06 a: Left	r=0.00 V=1.21 a: Left	r=0.00 V=1.74 a: Left	r=0.00 V=2.39 a: Left	r=0.00 V=3.21 a: Left	r=0.00 V=4.20 a: Left	r=0.00 V=5.59 a: Left	r=0.00 V=6.19 a: Left	r=0.00 V=6.95 a: Left	r=0.00 V=7.80 a: Left	r=0.00 V=8.71 a: Left
r=0.00 V=0.64 a: Left	r=0.00 V=0.72 a: Left	r=0.00 V=0.81 a: Left	r=0.00 V=0.90 a: Left	r=0.00 V=1.01 a: Left	r=0.00 V=1.14 a: Left	r=0.00 V=1.64 a: Left	r=0.00 V=2.24 a: Left	r=0.00 V=2.97 a: Left	r=0.00 V=3.84 a: Left	r=0.00 V=4.98 a: Left	r=0.00 V=5.52 a: Left	r=0.00 V=6.19 a: Left	r=0.00 V=6.95 a: Left	r=0.00 V=7.72 a: Left
r=0.00 V=0.65 a: Left	r=0.00 V=0.73 a: Left	r=0.00 V=0.81 a: Left	r=0.00 V=0.90 a: Left	r=0.00 V=0.82 a: Left	r=-2.00 V=0.00 a: Left	r=-2.00 V=0.00 a: Left	r=-2.00 V=0.00 a: Left	r=-2.00 V=0.00 a: Left	r=-2.00 V=0.00 a: Left	r=-2.00 V=0.00 a: Left	r=0.00 V=4.91 a: Left	r=0.00 V=5.51 a: Left	r=0.00 V=6.18 a: Left	r=0.00 V=6.84 a: Left
r=0.00 V=0.73 a: Left	r=0.00 V=0.82 a: Left	r=0.00 V=0.92 a: Left	r=0.00 V=1.02 a: Left	r=0.00 V=1.00 a: Left	r=-2.00 V=0.00 a: Left	r=-2.00 V=0.00 a: Left	r=-2.00 V=0.00 a: Left	r=-2.00 V=0.00 a: Left	r=-2.00 V=0.00 a: Left	r=-2.00 V=0.00 a: Left	r=0.00 V=4.37 a: Left	r=0.00 V=4.91 a: Left	r=0.00 V=5.50 a: Left	r=0.00 V=6.06 a: Left
r=0.00 V=0.81 a: Left	r=0.00 V=0.91 a: Left	r=0.00 V=1.03 a: Left	r=0.00 V=1.16 a: Left	r=0.00 V=1.30 a: Left	r=0.00 V=1.47 a: Left	r=0.00 V=1.66 a: Left	r=0.00 V=1.88 a: Left	r=0.00 V=2.14 a: Left	r=0.00 V=2.61 a: Left	r=0.00 V=3.19 a: Left	r=0.00 V=3.89 a: Left	r=0.00 V=4.37 a: Left	r=0.00 V=4.89 a: Left	r=0.00 V=5.37 a: Left
r=0.00 V=0.89 a: Left	r=0.00 V=1.00 a: Left	r=0.00 V=1.13 a: Left	r=0.00 V=1.28 a: Left	r=0.00 V=1.45 a: Left	r=0.00 V=1.65 a: Left	r=0.00 V=1.87 a: Left	r=0.00 V=2.11 a: Left	r=0.00 V=2.40 a: Left	r=0.00 V=2.72 a: Left	r=0.00 V=3.08 a: Left	r=0.00 V=3.47 a: Left	r=0.00 V=3.89 a: Left	r=0.00 V=4.35 a: Left	r=0.00 V=4.76 a: Left
r=0.00 V=0.86 a: Left	r=0.00 V=0.96 a: Left	r=0.00 V=1.08 a: Left	r=0.00 V=1.22 a: Left	r=0.00 V=1.37 a: Left	r=0.00 V=1.54 a: Left	r=0.00 V=1.73 a: Left	r=0.00 V=1.94 a: Left	r=0.00 V=2.18 a: Left	r=0.00 V=2.45 a: Left	r=0.00 V=2.75 a: Left	r=0.00 V=3.09 a: Left	r=0.00 V=3.47 a: Left	r=0.00 V=3.87 a: Left	r=0.00 V=4.22 a: Left
r=0.00 V=0.81 a: Left	r=0.00 V=0.90 a: Left	r=0.00 V=1.01 a: Left	r=0.00 V=1.13 a: Left	r=0.00 V=1.27 a: Left	r=0.00 V=1.42 a: Left	r=0.00 V=1.58 a: Left	r=0.00 V=1.77 a: Left	r=0.00 V=1.98 a: Left	r=0.00 V=2.21 a: Left	r=0.00 V=2.47 a: Left	r=0.00 V=2.76 a: Left	r=0.00 V=3.09 a: Left	r=0.00 V=3.44 a: Left	r=0.00 V=3.74 a: Left

Figure 1: State values after 100 value iterations

2.2 Question 2

The state value of the harbour and rock states is 0. Both are termination states, the episode finishes in both cases: in the former case because the objective has been reached, and in the latter because the sailor dies of hitting the rock. The termination implies there are no other rewards following those states, so the discounted return provided in that location is obviously null.

2.3 Task 2

The following images shows the final render in 3 executions of the program where 100 iterations of the value iteration algorithm were used both to update state values and to determine an optimal policy. The sailor always managed to find his way to the goal.

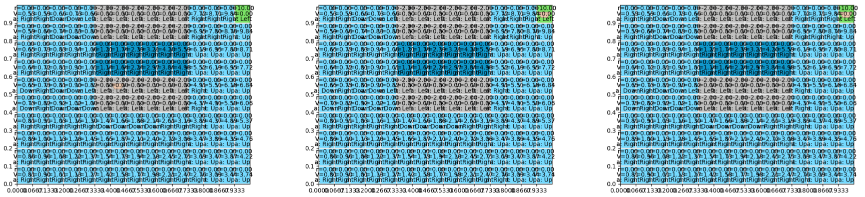


Figure 2: State values after 100 value iterations

2.4 Question 3

During the previous executions, the sailor was always trying to use the short path between the rocks to reach the goal. Despite the possible penalty resulting in hitting the rocks, the great difference in distance to the goal compared to the path going around the goal made this solution still optimal.

But if we change the penalty for hitting the rocks to -10, making the sailor value his life more, he will absolutely try to avoid being closer to the rocks and will choose a different path, going towards the bottom of the grid and then right towards the column of the harbour. The behaviour we saw showed that the sailor was always keeping himself at least 2 steps away from the rocks.

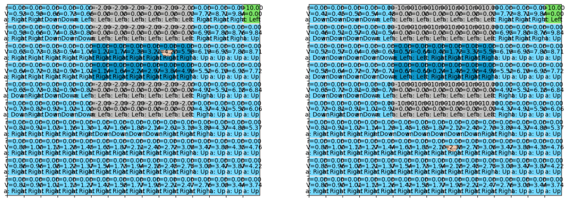


Figure 3: Different behaviour depending on rocks penalty

2.5 Question 4

I tried running the algorithm for 50, 25, 20 and 10 iterations. The final grid with value and policy functions is shown in the next figures for all these cases.

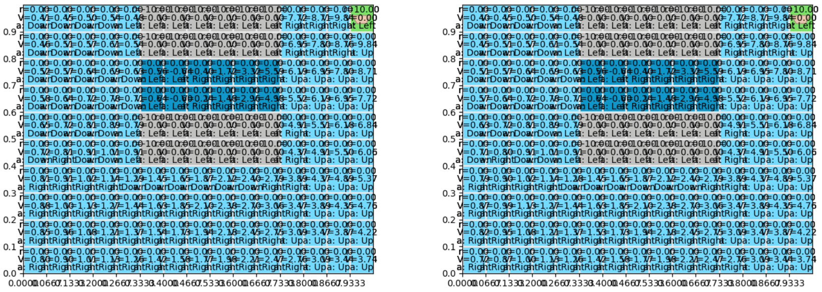


Figure 4: Value and policy functions convergence with 50 and 25 iterations

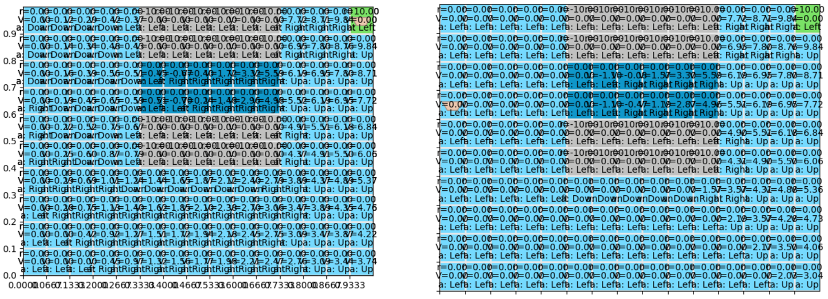


Figure 5: Value and policy functions convergence with 20 and 10 iterations

It's immediate to notice that the policy function converges before the value function: once the relative order among the state values is fixed, the optimal policy has been found; from that moment on, state values will keep changing and adjusting to the optimal values, but the policy will stay the same.

With 50 iterations, both functions have already converged as we can see from the state and action values that are the same as in the last figure of the previous exercise, since we ran the computation again with a penalty of -10 for hitting the rocks .

With 25 iterations the value function has slightly different values, meaning it did not converge yet to the values of 50 iterations (at least up to 2 decimal digits). But the relative order of states it's the same and we can see that the policy function has converged.

With 20 iterations the policy function has not converged yet but the sailor is still able to find his way to the goal.

With 10 iterations the sailor is not even able to find a path to reach the harbour.

2.6 Task 3

Changing the termination condition with the use of a small threshold, a new run of the algorithm (with rock penalty back to -2) terminated the value iteration in 31 steps. The following figure shows the results of an episode after this execution.

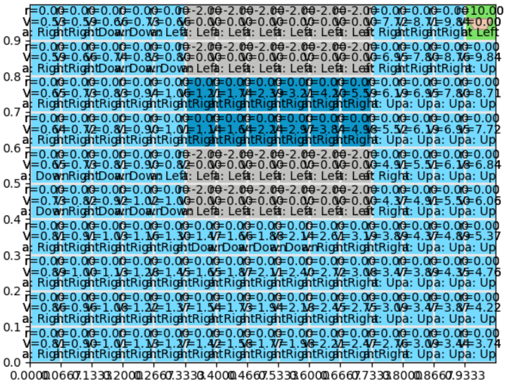


Figure 6: Value iteration with threshold as termination condition

2.7 Task 4

Running the program for $N = 1000$ episodes, the discounted return of the initial state has a mean $\mu(R_0) = 0.67$ and a standard deviation $\sigma(R_0) = 1.36$.

2.8 Question 5

Since we are executing the optimal policy, the result regarding the discounted return is coherent with the optimal value function calculated, that says that the value of the initial state is $V^*(s_0) = 0.66$. The value function tells the value of a state in terms of expected return when starting in that state and following the chosen policy thereafter; that's why, over a great number of episodes, the average discounted return is expected to tend to this value.

2.9 Question 6

The value iteration approach used in the previous tasks assumes complete knowledge of the environment, i.e., having the complete probability distributions of all possible transitions between states. Given a state and an action, with this information it's possible to generate predictions of the resultant next state and next reward, thus simulating the environment to calculate value and policy functions. [1]

However, in an unknown environment, this information is not available and real experience generated by the environment is needed to learn these probability distributions.

References

[1] R. S. Sutton and G. Barto. *Reinforcement Learning.* The MIT Press, 2015.