Reinforcment Learning Exercise 2

Franceschini Riccardo 769697

October 9, 2019

1 Sailor Gridworld

1) The agent in the Sailor Gridworld environment setup is the one that controls and decide which is the best direction to take for the boat. In this case, the agent can control only the boat and decide which is the best action to take (up, down, left, right) in order to reach the harbour as fast as possible and without crashing. The environment is the representation of the world. Thus, in the environment are included the probabilities of the wind the probabilities of the calm water, the position of both rocks and harbour and the rewards positive of negative of hitting them. In general, anything that cannot be controlled by the agent (in this case the boat) is part of the environment.

2 Value Iteration

2.1 Task 1

In order to implement the value iteration function I implemented four nested loop which are fundamental for exploring all the options (rows, column and actions) and selecting the best one. The values of the cells start to increase before in the ones near the harbour and continuing iterating, the values spread around neighbour cells until it covers the entire map and converges

2.2 Q2

After the implementation of the value function, the resulting state values of rocks and harbour are zero (fig1), this happen because they are both ending state with a reward (positive or negative) and for this reason their state value are always zero.

2.3 Q3

With the reward for hitting the rocks at -2, the sailor tries to pass between the rock and sometimes is able to reach the harbour while sometimes it crashes in the rocks. However, setting the reward for hitting the rocks at -10 the sailor chooses the longer and safer path around the rocks.

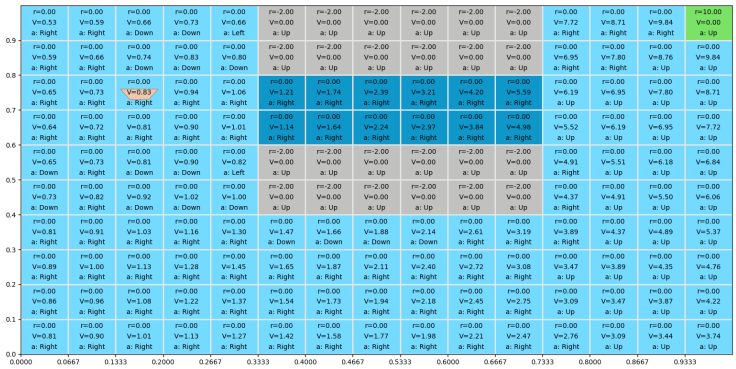

Figure 1: State values gridworld (rocks reward -2)

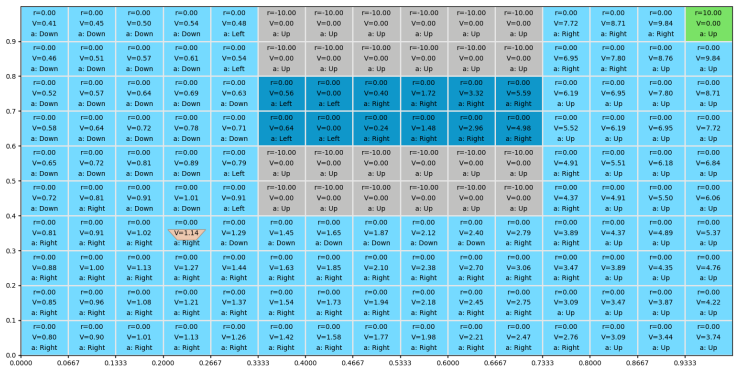

Figure 2: Rocks reward -10

2.4 Q4

After running the value iteration with different amount of iterations it was found that the value function converges after 46 iterations while the policy converges few iterations before. The policy converges faster because, in order to make the right decision it only needs that the state value of the right action has to be greater than the other ones, the difference in values is not relevant for the policy.

2.5 Task 3

In order to change the termination condition, the external loop has changed and as a termination condition, I checked if the differences between the previous state values and the current one are less then the threshold. If the difference is less it means that there is no value that has changed more than the threshold, so the algorithm has converged.

2.6 Task 4

In order to implement the discounted return of the inital state, I tested the policy for 1000 episodes, in each episodes, I saved the cumulative reward which is in this case (because all the rewards are 0 except for the last one) is

$$discret = \gamma^{N_S - 1} * r \tag{1}$$

where r is the final reward and N_S is the number of step of the episode.
Otherwise the standard formula should be :

$$G = \sum_{k=0}^{N} \gamma^k r_k \tag{2}$$

where N is the number of reward received, γ is the discount factor and r is the reward received after each action.
The standard deviation and the mean are :

```
mean :0.5585047933954507
std dev :1.1143632990925 5
```

Figure 3: mean and standard deviation

2.7 Q5

The relationship between the discounted return and the value function is that in the value function we have the expected value of the expected reward. In fact, we use the value function because we want to approximate the discounted

reward in order to pick the best decision.

$$G = \sum_{k=0}^{N} \gamma^k r_k \tag{3}$$

$$V = \mathbf{E}(G) \tag{4}$$

2.8 Q6

In a real application where the robot has to explore the unknown environment, the value iteration method cannot be used. In fact, in order to use the value iteration we should know how the environment behave. For example, in the case of the sailor we already know how that the environment is structured in a certain way (ex. wind probabilty), so, using the value iteration is possible to explore in order to find the best path. On the other hand, a robot without knowing the environment cannot find the best path simply because it has to explore before use the value iteration.