Advanced Concepts in Machine Learning

Assignment 4 Report

Maastricht University

Department of Data Science and Knowledge Engineering

Tomasz Stańczyk - i6209867

Ismail Alaoui Abdellaoui - i6204837

November 2019

Introduction

The purpose of this assignment is to use a reinforcement learning algorithm in an environment to plot the state-value function. The environment used is Open Ai Gym's "Mountain Car", which provides all the required properties to get started.

Reinforcement Learning algorithm

Q-learning with temporal difference

The algorithm we chose to implement is Q-learning with temporal difference. We first created a Q matrix of shape (P,S,A) where P is the number of possible positions, S is the number of possible speeds, and A is the number of possible actions. These numbers will be explained in more detail in the next section. This matrix was initialized with 0 values. Then in the main loop, the following steps are done:

- Select the best action: for a given position and speed, we select the action that corresponds to the highest Q value.
- Perform this action: the environment provides a function "step(action)" which performs the action and returns a 4-tuple: an observation made of the speed and position, the reward, whether the episode is done or not, and an information dictionary.
- Update the Q matrix with the new Q value: We update the matrix using the following formula:

$$Q(s,a) = (1-\alpha)\cdot Q(s,a) + \alpha\cdot(r + \gamma \max_{a'} Q(s',a'))$$

Where r = -1, and $\max_{a'} Q(s',a')$ corresponds to the maximum Q value for the given speed and position after executing the specific action.
s corresponds to the observation, which is made of the position and speed of the car.
a corresponds to the action performed (0 for going left, 1 for doing nothing, and 2 for going right).
r corresponds to the reward after executing an action. It is always -1 until an episode reaches its end.
Concerning the values of **α** and **γ**, they were set to different values in our experiments to visualize the state-value function.

It should be noted that the Q matrix is never reset: it is updated all the time, during each timestep and during each episode. We decided to set the number of episodes to 10, as originally set in the code provided. We also considered that this number of episodes was enough to have satisfactory results.

State space

In Gym's environment, the observation of the space is given by a tuple (P,S) where P is the value of the position and S is the value of the speed. The minimum and maximum values for the position are -1.2 and 0.6, respectively. The minimum and maximum values for the speed are -0.07 and 0.07, respectively. Since both of these values are continuous and that we need to store the Q values for a specific pair of position and speed, the values had to be discretized. We discretized both of the values through the following steps:

- Position: round the number to 1 decimal place, multiply by 10, and add 12,
- Speed: round the number to 2 decimal places, multiply by 100, and add 7.

The result is that we had 19 possible position values ranging from 0 and 18, and 15 possible speed values, ranging from 0 and 14. Therefore, every time we use Gym's function "step(action)", we first discretize the position and the speed, and the resulting values become indexes for the update of the Q matrix.

State Value Visualization

For State Value visualization, the following strategy has been applied. For each observation (meaning the position-speed pair), the smallest q-value has been extracted. By saying extracted, it corresponds to the fact that for each observation, there are 3 values, for each of the possible actions and the smallest value has been selected.
The reason behind taking the minimum values is the fact that for almost the entire duration of an episode, negative rewards (with values of -1) are applied. Therefore, if particular Q-value is zero, then it means that the action corresponding to this cell has never been executed for an observation (position and speed) related to that cell. For example, let us assume, that for certain observation, the following Q-values per action have been recorded:

- -9 for action 0,
- -5 for action 1,
- 0 for action 2.

In such a case, action 2 has never been performed for the given position-speed pair. Therefore, it was considered unreasonable to include Q-value corresponding to this action as a part of State Value Function.
On the contrary, the lowest Q-value for given observation indicates that the corresponding action has been performed the highest number of times there. Consequently, such Q-value was selected to represent the state value for given observation.

A series of experiments was performed with number of episodes set to 10 and the values of the discount rate γ and the learning rate α being altered. Starting with γ = 0.9, the visualizations of State Value functions for the following values of α have been performed: [0.25, 0.5, 0.75, 1.0]. The corresponding State Value Function visualizations are presented in Figures 1-4 below:

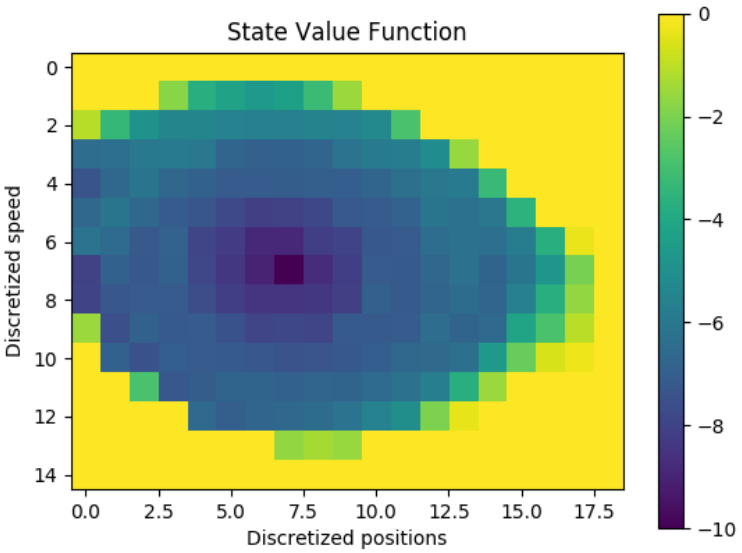

Figure 1 - State Value Function visualization for α = 0.25, γ=0.9

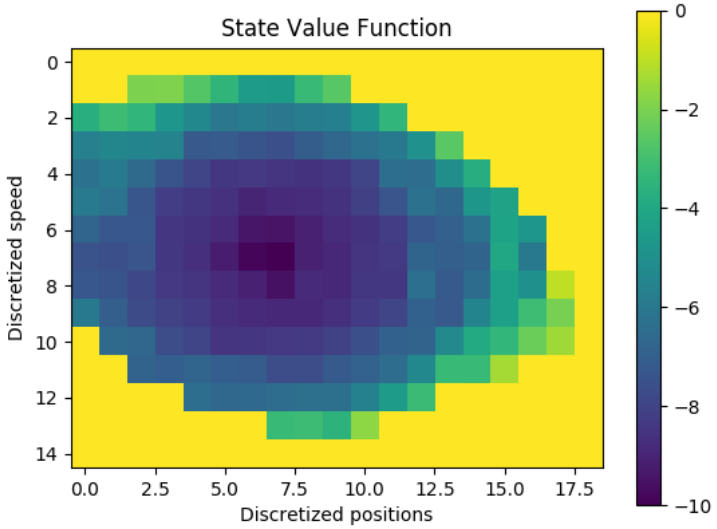

Figure 2 - State Value Function visualization for α = 0.5, γ = 0.9

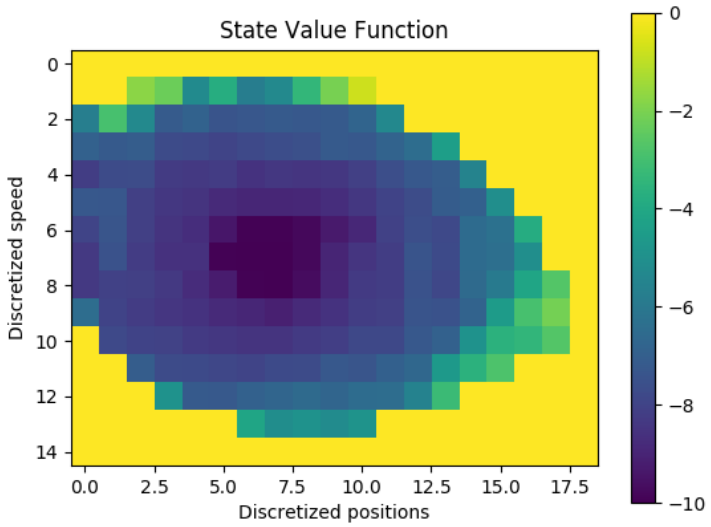

Figure 3 - State Value Function visualization for $\alpha = 0.75$, $\gamma = 0.9$

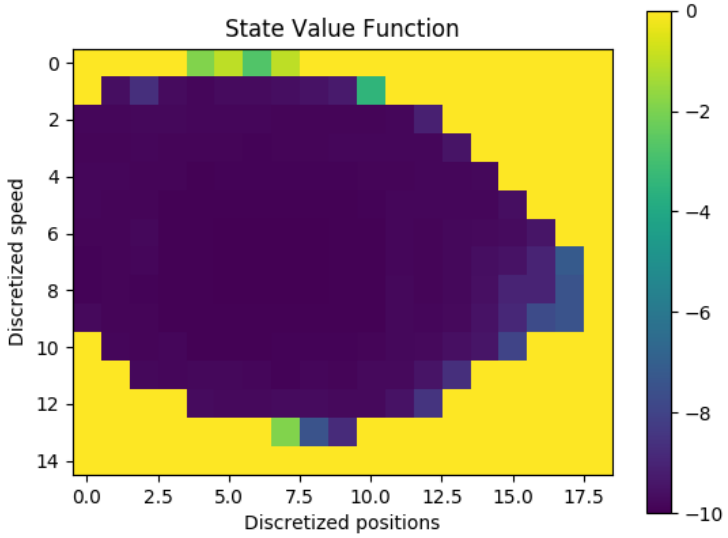

Figure 4 - State Value Function visualization for $\alpha = 1$, $\gamma = 0.9$

Bright colors on the figures above correspond to higher values. More specifically, when particular value is negative (so different than zero), the lower its absolute value is, the brighter representation color it receives. Moreover, in all the figures above (constant γ and α being altered), the ranges of the registered values are the same (from 0 to -10).
It was assumed that the brighter the plot in general is (considering the number of bright/dark cells in the visualization), the less steps needed to be performed in total to reach the goal.

Therefore, for altering the values of γ, α was set to 0.25, as the visualization for such value looked the most promising based on the stated assumptions. The following values of γ have been tested: [0.3, 0.6, 0.85 0.9, 0.95, 0.99]. The State Value Function visualizations are displayed in the Figures 5-10 below.

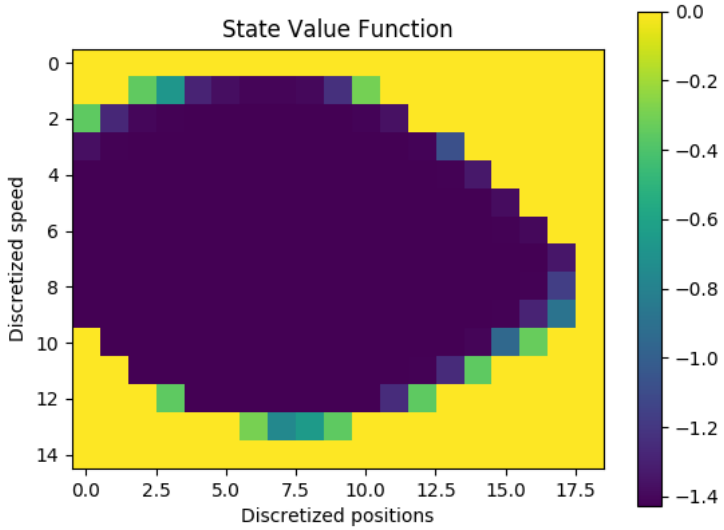

Figure 5 - State Value Function visualization for α = 0,25, γ = 0.3

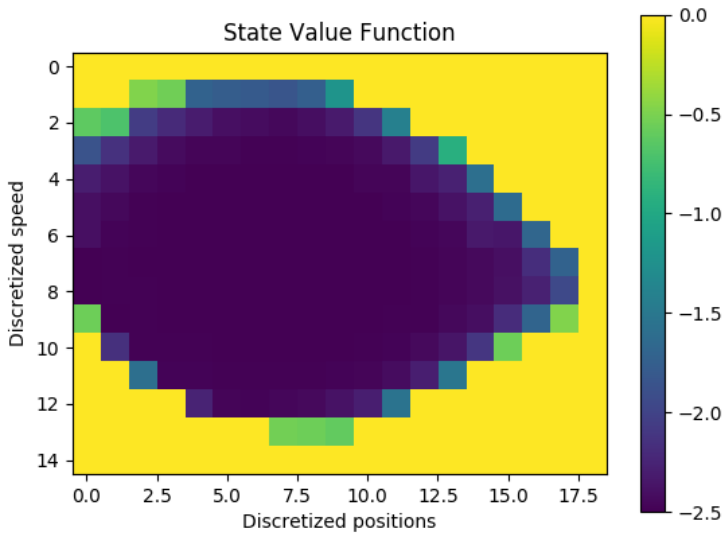

Figure 6 - State Value Function visualization for $\alpha = 0{,}25$, $\gamma = 0.6$

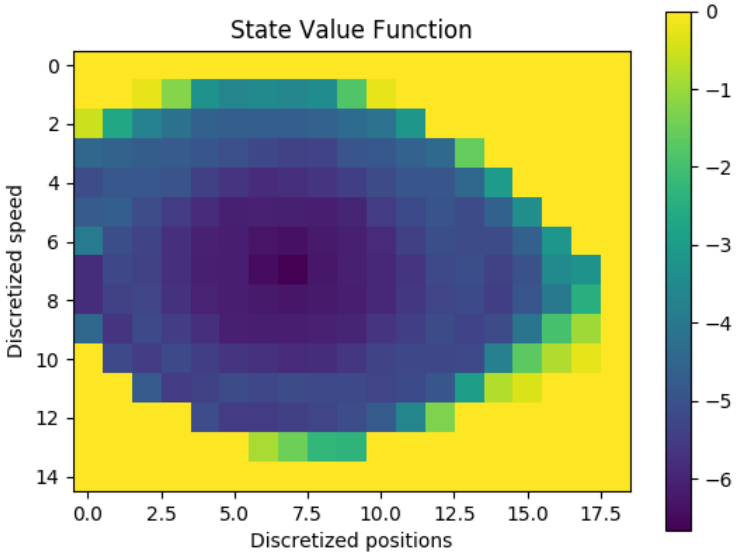

Figure 7 - State Value Function visualization for $\alpha = 0{,}25$, $\gamma = 0.85$

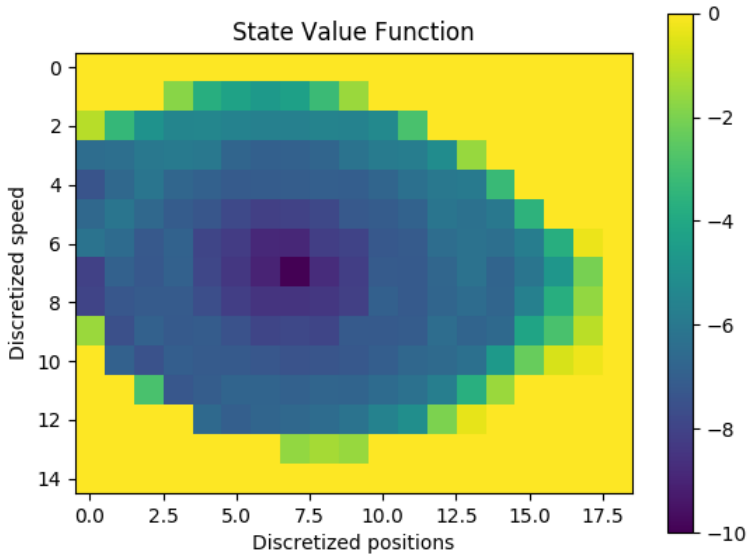

Figure 8 - State Value Function visualization for α = 0,25, γ = 0.9 (the same as Figure 1)

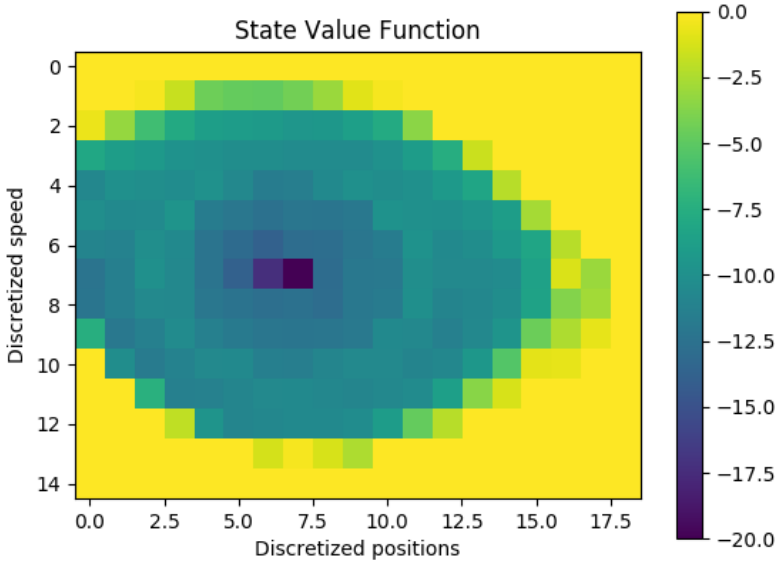

Figure 9 - State Value Function visualization for α = 0,25, γ = 0.95

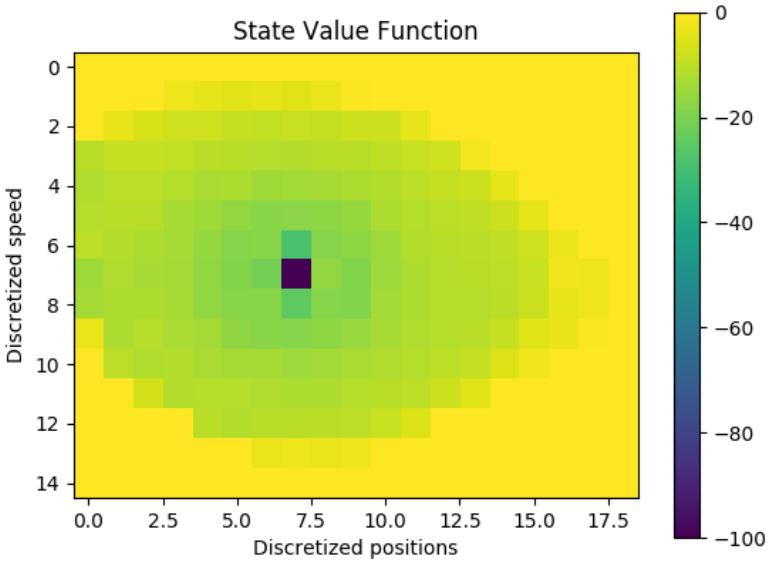

Figure 10 - State Value Function visualization for $\alpha = 0{,}25$, $\gamma = 0.99$

In the figures above, it seems that the higher the value of γ is, the less steps needed to be performed so as to reach the goal. However, the value ranges of the State Value Functions visualized in figures 5-10 are not the same. Therefore, it cannot be directly inferred which of the setting is the best based on the plot brightness here. However, what could be observed, is the fact that the growing values of γ cause the values represented by the darker colors to be more focused. Meaning that smaller and smaller cluster of similar observation values (meaning observations with similar/adjacent values of corresponding positions and speeds) was visited for the (relatively) high number of times.
Furthermore, higher γ values cause lower values (meaning higher absolute values of the negative scores) obtained at the State Value Function.

Finally, it was considered that the best representative plot of State Value Function comes from Figure 1, so with values of α set to 0.25 and γ set to 0.9.
The following list presents the number of steps required to finish each episode with these, i.e. to reach the goal defined within the score of the moving car problem. The values of α and γ were set as in visualization from Figure 1.

Episode 1 finished after 13857 timesteps.
Episode 2 finished after 1202 timesteps.
Episode 3 finished after 5485 timesteps.
Episode 4 finished after 1219 timesteps.
Episode 5 finished after 332 timesteps.
Episode 6 finished after 389 timesteps.
Episode 7 finished after 4300 timesteps.
Episode 8 finished after 624 timesteps.
Episode 9 finished after 613 timesteps.
Episode 10 finished after 803 timesteps.

It can be seen that each episode has successfully been finished, the goal was reached every time. Based on the number of steps required to finish each episode, it can be observed that significantly more steps were required to finish the first episode than to finish 3 last episodes. Interestingly, the number of steps required to finished episodes 5 and 6 was even lower than for the last episodes (8, 9 and 10).
This as a whole could suggest that indeed the Q-values were representing the learnt knowledge. However, it seemed most beneficial to stop after episode 5. Nevertheless, this could be an effect of various factors like the values of position and observation returned by the environment variable from external library (Gym) and their rounding.
Ultimately, the learning could be acknowledged, because not only was the goal reached after each episode, but the number of steps required to finish the episode in the last episodes was considerably lower than the one corresponding to the first episode.