Homework 5

Daniel Frank

May 23, 2017

1 Mountain Car Problem with Tabular $Q(\lambda)$

1a)

For the states there is p and $\dot{p}$ which are initialized in their range. P and Q are initialized by zero.

```
p_d = np.linspace(-1.2,0.5, d)
p_dot_d = np.linspace(-0.07,0.07,d)

P = np.zeros(d*d*2).reshape(d*d,2)
Q = np.zeros(d*d*env.action_space.n).reshape(20*20,env.
    action_space.n)
```

so $Q \in \mathbb{R}^{400x3}$ and $P \in \mathbb{R}^{400x2}$. So P represents the discrete states

$$P = \begin{pmatrix} \begin{bmatrix} p_0 & \dot{p}_0 \end{bmatrix} \\ \vdots \\ \begin{bmatrix} p_{19} & \dot{p}_0 \end{bmatrix} \\ \begin{bmatrix} p_0 & \dot{p}_1 \end{bmatrix} \\ \vdots \\ \begin{bmatrix} p_{19} & \dot{p}_{19} \end{bmatrix} \end{pmatrix}. \tag{1}$$

Q represents the Value function for each action

$$Q = \begin{pmatrix} s_1 - \begin{bmatrix} a_0 & a_1 & a_2 \end{bmatrix} \\ \vdots \\ s_{400} - \begin{bmatrix} a_0 & a_1 & a_2 \end{bmatrix} \end{pmatrix}. \tag{2}$$

After applying $Q(\lambda)$ with a fixed value of $\lambda = 0.8$ Q updates as shown in table 1. The values is taken as an absolute value to have the z axis positive. So one could interpret a lower Value of the following plot as better.

The number of steps each episode takes is shown in figure 1

Table 1: $|Q_{max}|_i$ where $i \in \mathbb{S}$

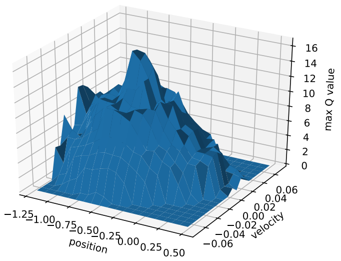

Table 2: $episode = 10$

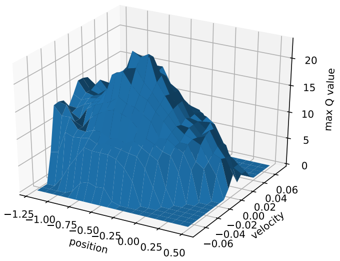

Table 3: $episode = 30$

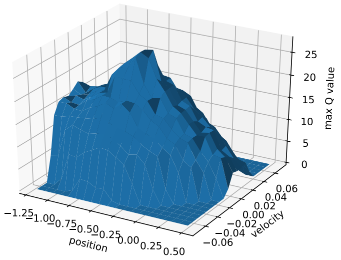

Table 4: $episode = 50$

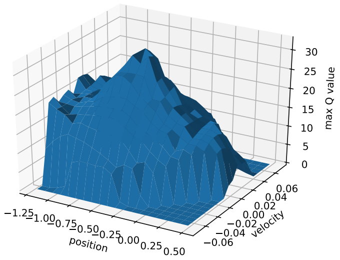

Table 5: $episode = 70$

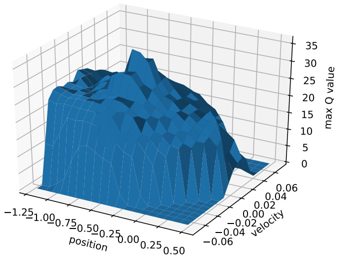

Table 6: $episode = 90$

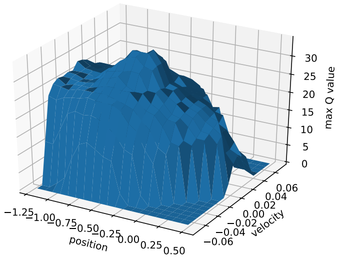

Table 7: $episode = 100$

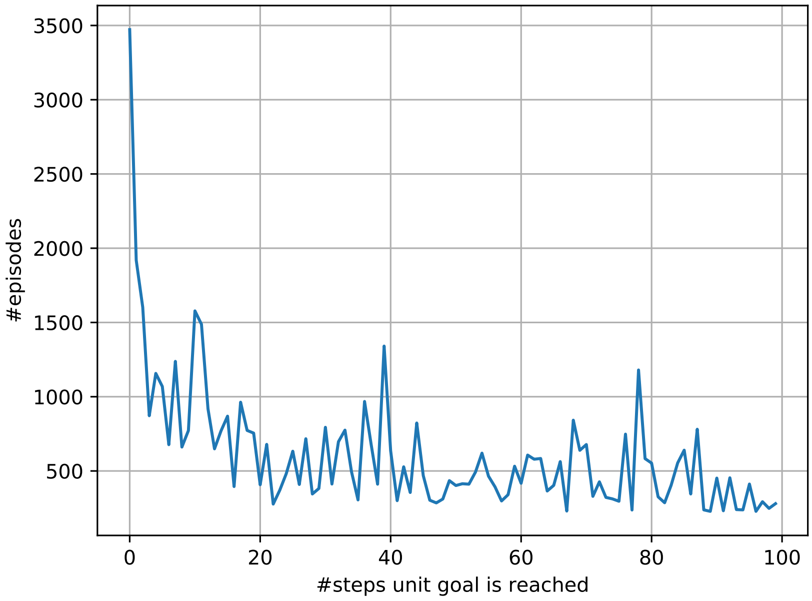

Figure 1: Number of steps needed per episode, for $\lambda = 0.8$.

1b)

How should the process be repeated? Using a initial Q or using the already learned Q for the next ten times?
Figure 4 shows the result of the following code.

```
step_til_end = np.zeros(episode*10).reshape(10,episode)
cumulative = []
sum_average = []
sum_steps = 0
for n in range(10):
    st,Qnew = TabularQ(Q,P,episode, steps, epsilon, gamma
        , alpha, lamb)
    step_til_end[n,:]=st
    sum_average.append(1/episode * sum(step_til_end[n,:])
        )
    sum_steps += 1/episode * sum(step_til_end[n,:])
    cumulative.append(sum_steps)
```

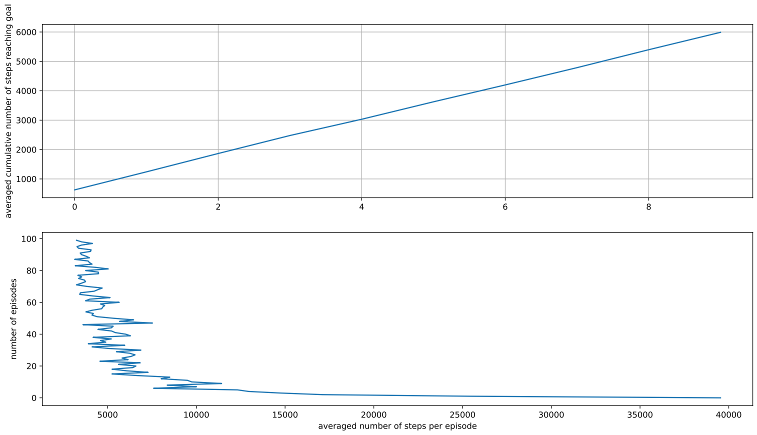

Figure 2: Averaged number of steps needed per episode, for $\lambda = 0.8$.

1c)

When changing the discretization intervals the time of calculation increases. In figure 3 the plot on top shows the performance of the different intervals. The plot on the bottom shows the time until it calculates 100 episodes in seconds.

The intervals are not as high as in the exercise since it was running smoothly anymore when the intervals were set to 200.

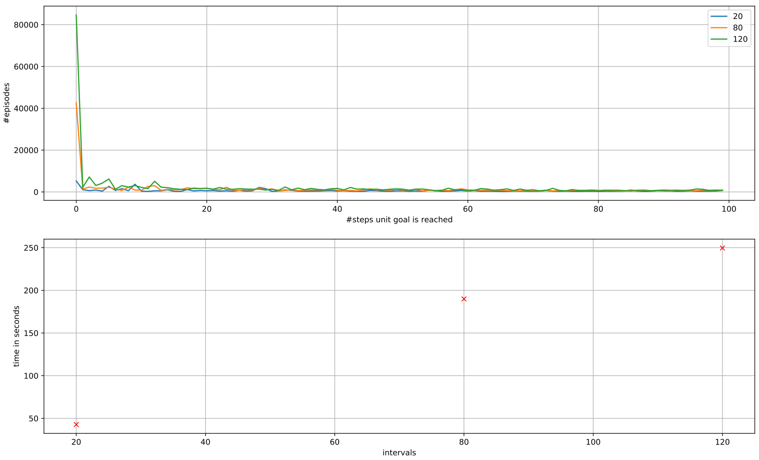

Figure 3: Changing the intervals $d = [20, 80, 120]$.

2 Mountain Car Problem with linear $Q(\lambda)$ and RBFs

2.1 Problem

First there is an introduction about the notation needed

$$\Theta = \begin{pmatrix} (i=1)\begin{bmatrix}\theta_{a_0} & \theta_{a_1} & \theta_{a_2}\end{bmatrix} \\ \vdots \\ (i=n)\begin{bmatrix}\theta_{a_0} & \theta_{a_1} & \theta_{a_2}\end{bmatrix} \end{pmatrix} \tag{3}$$

so $\Theta \in \mathbb{R}^{32\text{x}3}$.

$$\phi(s) = \begin{pmatrix} \exp\left(-\frac{\left(\begin{pmatrix}s_p\\s_v\end{pmatrix} - \begin{pmatrix}c_{p_1}\\c_{v_1}\end{pmatrix}\right)^T \begin{pmatrix}\sigma_p^2 & 0\\ 0 & \sigma_v^2\end{pmatrix} \begin{pmatrix}s_p\\s_v\end{pmatrix} - \begin{pmatrix}c_{p_1}\\c_{v_1}\end{pmatrix}}{2} \right) \\ \vdots \\ \exp\left(-\frac{\left(\begin{pmatrix}s_p\\s_v\end{pmatrix} - \begin{pmatrix}c_{p_n}\\c_{v_n}\end{pmatrix}\right)^T \begin{pmatrix}\sigma_p^2 & 0\\ 0 & \sigma_v^2\end{pmatrix} \begin{pmatrix}s_p\\s_v\end{pmatrix} - \begin{pmatrix}c_{p_n}\\c_{v_n}\end{pmatrix}}{2} \right) \end{pmatrix} \tag{4}$$

That means $\theta(s) \in \mathbb{R}^{32x1}$. The centerpoints of the RBFs are set to discrete sates in statespace, discretized by $n_p = 4$ and $n_v = 8$.

$$c = \begin{pmatrix} \begin{bmatrix} c_{p_1} & c_{v_1} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} c_{p_n} & c_{v_1} \end{bmatrix} \\ \begin{bmatrix} c_{p_1} & c_{v_2} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} c_{p_n} & c_{v_n} \end{bmatrix} \end{pmatrix} \tag{5}$$

so it follows $c \in \mathbb{R}^{32x2}$. After implementing Linear $Q(\lambda)$ from the slides I do have the problem that the values for $Q(s,a)$ are the same for all actions. That comes from updating Θ. In the algorithm it says

$$\Theta \leftarrow \Theta + \alpha e[r + \gamma \mathrm{max}_a Q(s', a; \Theta) - Q(s, a; \Theta)] \tag{6}$$

now the implementation for the update works like

```
for i in range(len(Theta[0,:])):
    for j in range(len(Theta[:,0])):
        Theta[j,i] = Theta[j,i] + alpha*e[j]*(rnew+gamma*
            Qstar-Q[i])
```

where $Qstar$ is $max_a Q(s', a; \Theta)$ this value will not change for all Θ values. The only part that will change for the action is $Q[i]$, but since it is initialized by zero the values will be the same in every iteration.

So it follows that $[\theta_{a0}\theta_{a1}\theta_{a2}]_i$ will have all the same values $\theta_{a0} = \theta_{a1} = \theta_{a2}$. Can you help me here? I do not know how to fix this problem.

2.2 Fixed

After fixing the size of e the algorithm worked. Size of e is the same as size of Θ. The evolution of Q for linear $Q(\lambda)$ with RBFs and $\alpha = 0.001$ is shown in 2.2.

Figure 2.2 shows the the evolution of the states that are needed to reach the goal state.

Table 8: $|Q_{max}|_i$ where $i \in \mathbb{S}$

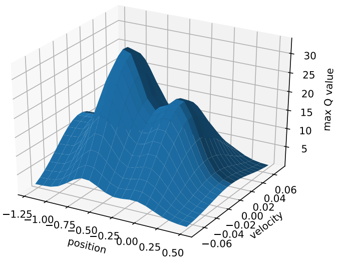

Table 9: $episode = 10$

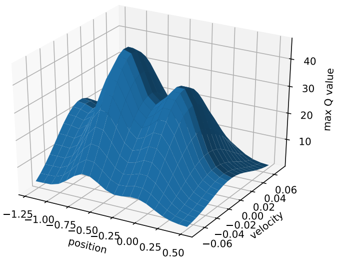

Table 10: $episode = 30$

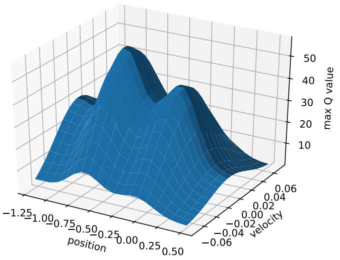

Table 11: $episode = 50$

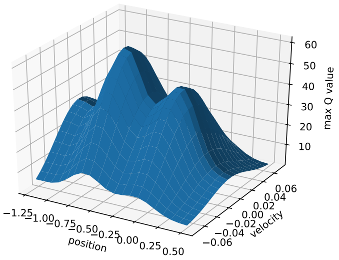

Table 12: $episode = 70$

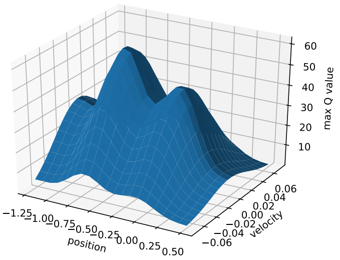

Table 13: $episode = 90$

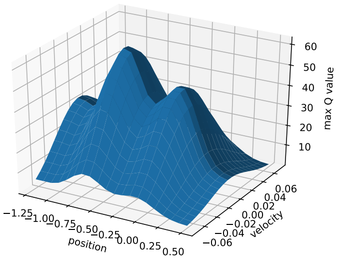

Table 14: $episode = 100$

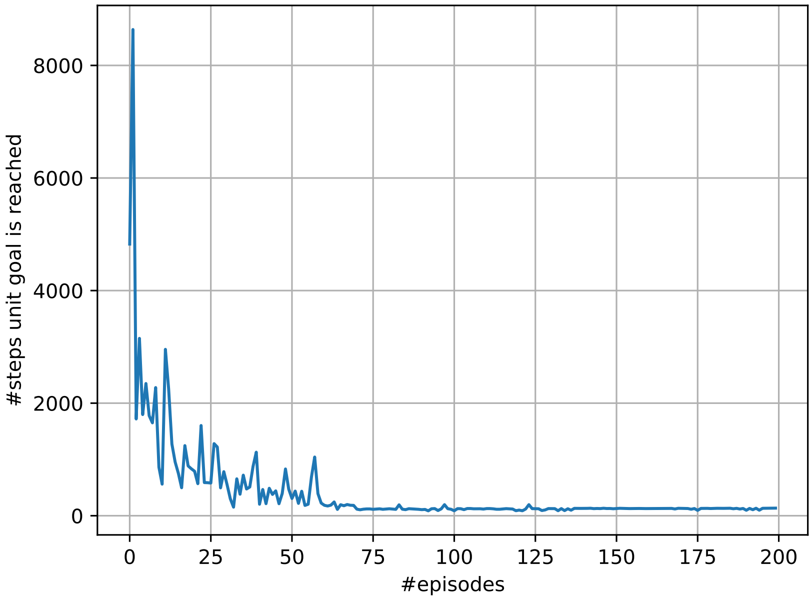

Figure 4: Number of steps needed per episode, for $\lambda = 0.9$ and $\alpha = 0.001$.