IN4320 Machine Learning

Reinforcement Learning

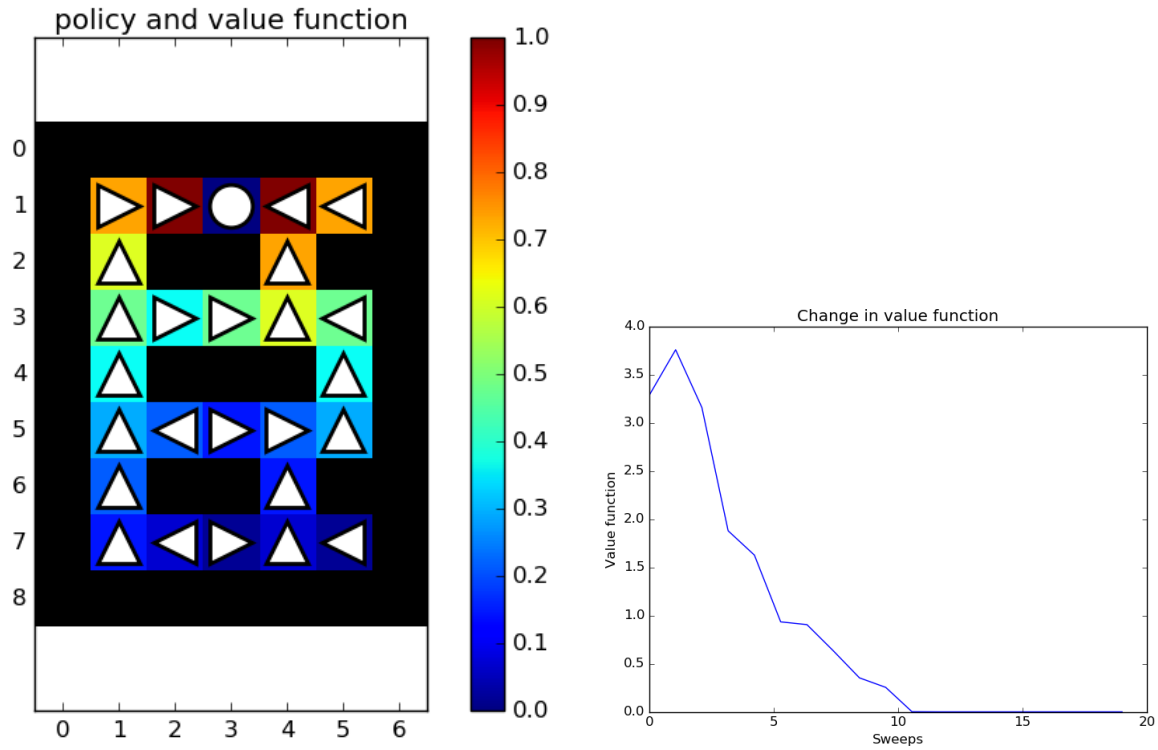
17 May 2019

TU Delft
Coordinated by Prof Jens Kober

Matthijs Bekendam 4725751 J.M.Bekendam@student.tudelft.nl

Contents

1 Question 1



(a) Image of grid world with colors indicating optimal value function and arrows indicating the optimal policy.

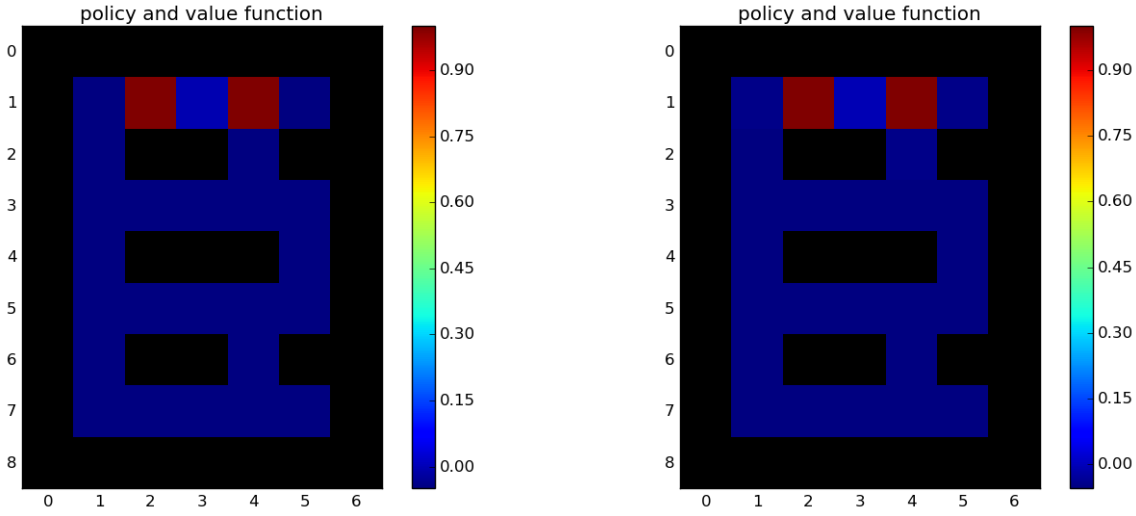
(b) Change in value function. After about 10 sweeps the value function does not change anymore and is convergenced.

Figure 1

Figure 1b shows about 10 sweeps are required. The optimal policy and optimal value function are shown in Figure 1a. The value of the state "G" is zero since in the given scenario the robot receives a reward of one, only a step before it reaches the goal (episodic). This is why blocks next to "G" have a value of 1.

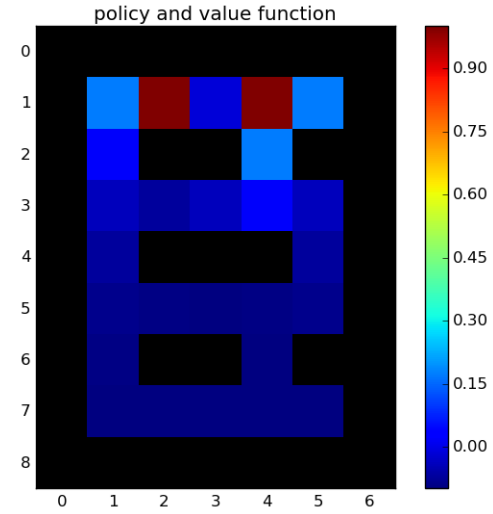
2 Question 2

Figure 2 shows four different scenarios in which the discount factor differs.

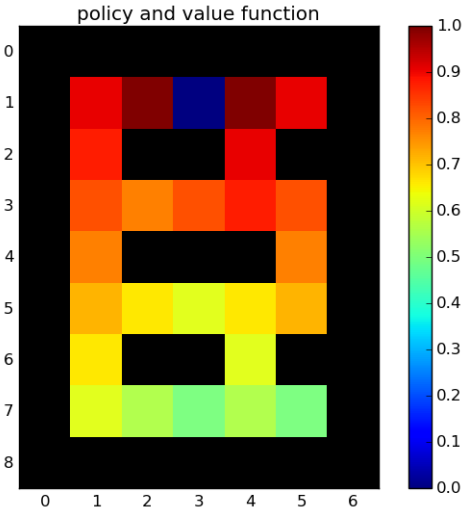


(a) Discount factor (γ) = 0

(b) Discount factor (γ) = 0.1



(c) Discount factor (γ) = 0.5



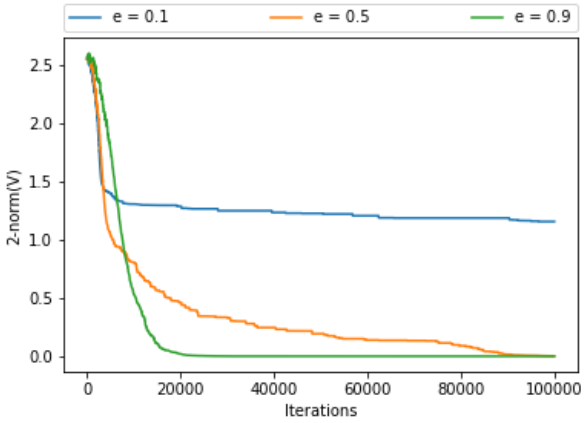
(d) Discount factor (γ) = 1

Figure 2: Optimal value function of different discount factors using 20 iterations. Default reward = -0.05.

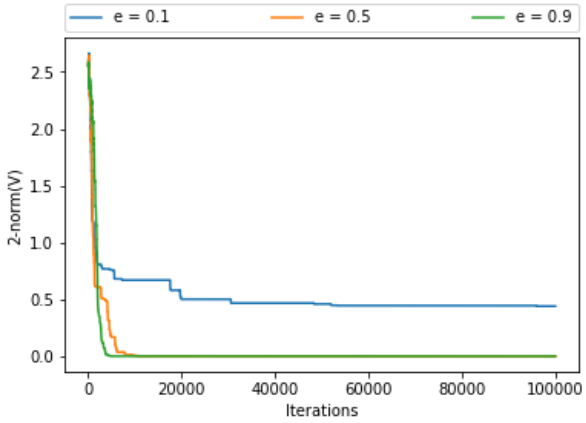
The effect gamma is that the reward after each time step decreases, essentially weighting wins and losses in the future less [3]. A factor of 0 will make the agent care only about current rewards, hence the agent will not move around since it does not take into account the positive reward at the end. A factor of 1 (or close to one) will make the agent aim for long-term high rewards, which is possible in our scenario since our learning environment is episodic.

3 Question 3

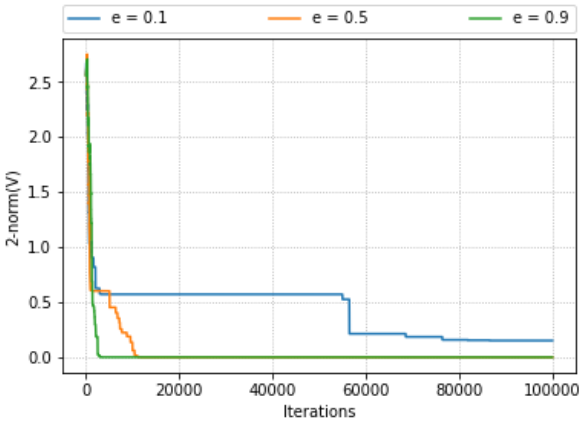
The value of epsilon determines how often a random action will be selected instead of actions bases on max future reward. It balances exploration/exploitation of the agent. Alpha (learning rate of step size) determines to what extend prior information is used, a factor 0 will make the agent only use prior information. A factor of 1 will make the agent ignore prior knowledge to enhance exploration. In deterministic settings, a learning rate of 1 is optimal.



(a) $\alpha = 0.1$



(b) $\alpha = 0.5$



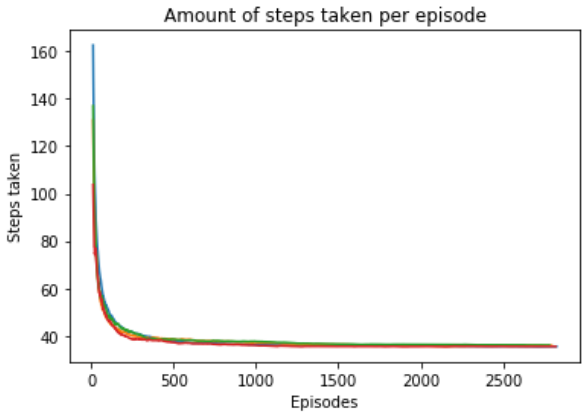
(c) $\alpha = 0.9$

Figure 3: The 2-norm of the difference between the value function estimated by Q-learning and the true value function (from Q-iteration) over the number of interactions with the system. Each figure shows a constant alpha with different epsilon values.

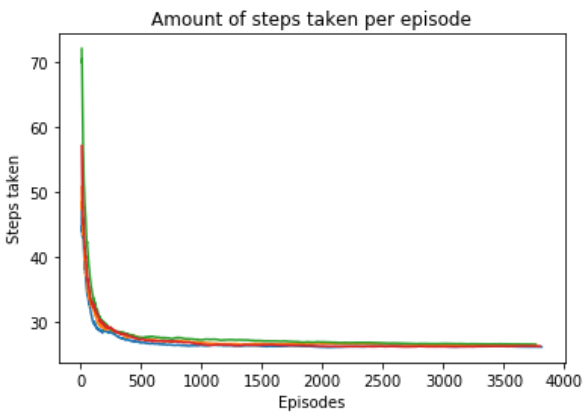
For the L2 distance to decrease we want similar value function matrices. Q-iteration and Q-learning differ in that Q-iteration analyses each grid cell each iteration, while the Q-learning agent only experiences surrounding grid cells. With a low epsilon, our agent will explore less and find its optimal path rather quickly. Since it does not explore much, it will not go into sub optimal routes which Q-iteration does evaluate. Hence, the L2 difference will be higher for lower epsilon values. As stated above, a higher alpha will also enhance exploration, hereby making Q-learning and Q-iteration more similar. This effect is shown in Figure 3.

4 Question 4

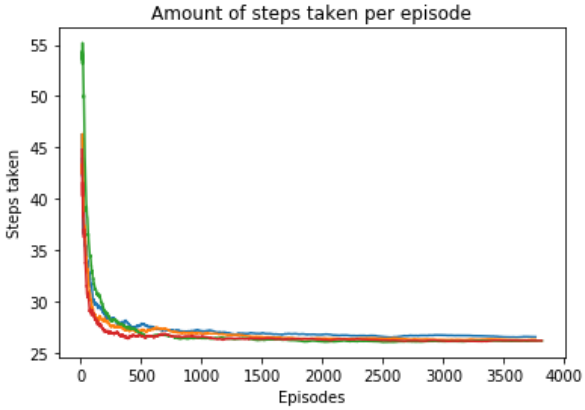
The first method used to speed up Q-learning is to remove the possibility for the agent to move downwards (see Figure 4b). The second method used was removing the possibility of the agent to stand still by letting it take an appropriate different action when the proposed action would lead into collision with a wall (see Figure 4c).



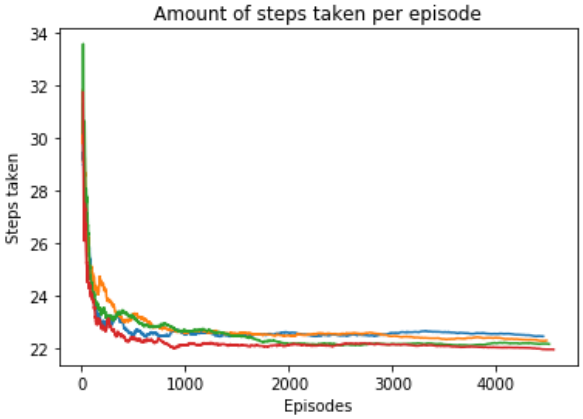
(a) Normal Q-Learning



(b) No-backwards movement Q-Learning



(c) Moving only Q-Learning



(d) Methods (b) and (c) combined

Figure 4: Amount of steps taken per episode with a moving average filter. Settings: $\alpha = 0.3$ and $\epsilon = 0.7$. Four different initializations were done with different random seeds (each Q-cell initialized with a random value between 0-0.3). About 10^6 iterations were performed.

Figure 4 shows the amount of steps per episode drastically decreases when using either methods or combined. Where the normal Q-Learning settles for an average of 40 steps per episode, combining both methods yields on average 23 steps per episode. Disadvantages of the methods would be that they are not applicable in every scenario. This is because both methods require prior knowledge of the environment and are rather specific to this particular environment.

5 References

[1]: V. Heidrich-Meisner, M. Lauer, C. Igel and M. Riedmiller (2007). Reinforcement Learning in a Nutshell. In 15th European Symposium on Artificial Neural Networks (ESANN).

2: Richard S. Sutton Andrew G. Barto (2018). Reinforcement Learning: An Introduction. 2nd Edition, MIT Press.

3: Stanford Computer Science department, Lecture 16- Machine Learning (Stanford),
$https://www.youtube.com/watch?v = RtxI449ZjSc$, Jul 22, 2008.

4: Vihar Kurama, Reinforcement Learning with Python, Nov 25, 2018

5: Vaibhav Kumar, Reinforcement learning: Temporal-Difference, SARSA, Q-Learning Expected SARSA in python, Mar 20 2018.