IN4320 Machine Learning Exercise

Exercises Reinforcement Learning

In this exercise you will apply various basic reinforcement learning methods to a toy example. Cite all the references you employed. Hand-in a PDF of the report via BrightSpace. Page limit is 4 pages A4 font-size 12 (including figures, tables, algorithms etc. Only the title page and references don't count towards the page limit). Please do not include your code. You can use any programming language you like.

Scenario Our robot now has to navigate a more complicated maze, see Fig. 1. The robot still has the 4 actions "left", "right", "up" and "down". It is working in world with 63 states, the obstacles are always at the locations indicated in the figure. It always starts in the state marked "S" and needs to reach the state marked with "G". The state "G" is terminal, i.e., the episode ends immediately once the robots reaches it. It gets a reward of 1 when reaching the goal:

$$r_{t+1}(s_t, a_t, s_{t+1}) = \begin{cases} 1 & \text{if } s_{t+1} = \text{"G"} \\ 0 & \text{otherwise} \end{cases}$$

The transitions are deterministic, the actions move the robot one field in the respective direction. If the action would lead to a state that is marked in gray, or if the robot is already in a gray state (no matter which action it takes), the robot does not move, i.e., $s_{t+1} = s_t$.

Exercise 1 (20 points) Implement the scenario and Q-iteration. The Q-function is initialized with 0. For $\gamma = 0.9$ the optimal V-function is given in Fig. 2. Reproduce this result. How many sweeps over all state-action pairs are required? What is the optimal policy π^*? Why is the value of the state corresponding to "G" equal to 0?

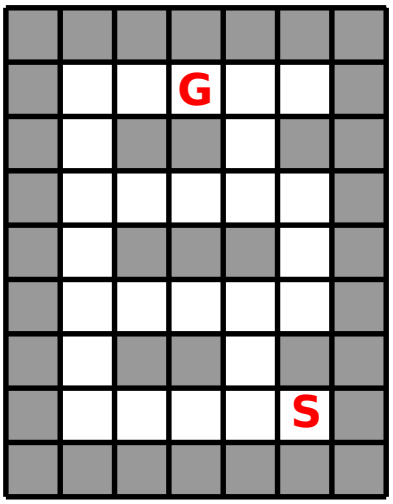

Figure 1: The robot maze.

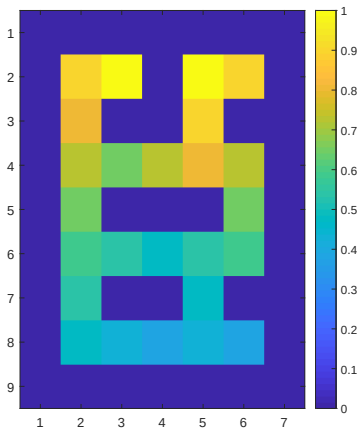

Figure 2: The optimal value function for $\gamma = 0.9$.

Exercise 2 (15 points) Show the optimal value functions V^* for $\gamma = 0$, $\gamma = 0.1$,$\gamma = 0.5$, and $\gamma = 1$. Discuss the influence of the discount factor γ.

Usually γ is defined as $0 \leq \gamma < 1$. Explain why $\gamma = 1$ will work here nevertheless.

Exercise 3 (25 points) Implement Q-learning with ε-greedy exploration. The Q-function is initialized with 0. After every episode the robot is reset to the starting state. For the rest of the exercises use $\gamma = 0.9$. You might need to set a maximum number of steps per episode and if the robot exceeds start a new episode.

Try different values for the exploration ε and the learning rate α (at least 3 each). Plot the difference (2-norm) between the value function estimated by Q-learning and the true value function (from Q-iteration) over the number of interactions with the system. Provide plots for the different values of ε and α. Describe and explain the differences in behavior. In this exercise high values of α will work well, when will this lead to problems?

Exercise 4 (40 points) Q-learning can take a huge number of interactions with the system (in my implementation 250000 steps) while Q-iteration converges significantly more quickly. Explain this behavior.

Implement two methods to speed up Q-learning from the literature (you don't need the explain the implementation in the report). You can think for example of including prior knowledge, better exploration, better initialization, better rewards, introducing hierarchy, etc. Briefly explain the methods. Evaluate each of the two methods separately and the combination of both of them (with graphs, figures, tables, ideally show the variance in the methods by doing multiple runs with different random seeds). Compare those results and the results from plain Q-learning. Discuss the advantages and disadvantages of the methods you picked.