Homework: Reinforcement Learning

This homework sheet will test your knowledge on reinforcement learning.

8

a) Decide to which learning type the following tasks belong. Choose from the options: supervised, unsupervised or reinforcement learning.

1. Training a program to play tic tac toe
2. Finding patterns in market data
3. Analyzing the sentiment of a sentence
4. Determining credit-worthiness of bank customers
5. Have a robot balance a pole
6. Analyze how a sequence of treatments affect a patient's health
7. Predict tomorrow's weather
8. Determining number of species in various animal data

4

b) Which approach of reinforcement learning (model-based or model-free) would you choose for the following games? Explain the reasons for your choice.

1. Tic tac toe
2. Chess

2

c) Name a sample task for each model-based and model-free reinforcement learning.

7

d) Model an MDP (S, A, R, T) in order to help a student making a decision to what extent he should study for an exam. Unfortunately, there are only three days left to prepare all materials. Furthermore, he gold a cold, which reduce his ability to study and might increase the risk that he is not absolutely fit.

In more detail, one can encode the above scenario with the following reward structure. For each day where he is sick and studies, he gets a reward of 1. For each day where he is healthy and studies, he earns a reward of 2. If he is healthy on the day of the exam, he receives an additional reward of 5.

His studies affect whether he is healthy or sick the subsequent day. More precisely, when he studies being sick, he face a change of 50 % that he is also sick the next day. If he instead rests on that day, he increases his chances to recover to 80 % but does not earn a reward. Luckily, he won't get sick again after he has once recovered.

In addition, also denote the initial and terminal states.

1

e) Now consider a simplified MDP of the previous task. Here, the student is sick and has only one day to study for the exam. Reward and transition probability remain the same. Find an optimal policy for the student, whether he should rest or study on his last day of preparation in order to maximize his reward. To do this, calculate the optimal value function over all possible policies.

9

f) Using `MDPtoolbox`, create a MDP for a 1×3 grid. In this grid, the central position gives a reward of 10. The left position results into a reward of -1 and the right position a reward of -10.

The agent can choose between the actions of moving `left` or `right` but cannot cross the left or right boundaries of the grid. An action does not succeed all the time but is subject to perturbations, in fact, it only succeeds with 80 % chance. With 20 % probability, the agent stays in its place.

Apply a policy iteration to your MDP with a discount factor of 0.9. Which are the recommended actions for each state?

2

g) Modify your code from the previous exercise such that the right state provides a reward of -1000 instead of -10. Can you explain the change in your policy? How can we ensure to obtain the optimal long-term policy with a reward of -10?

1

h) The performance of reinforcement learning depends on the choice of several parameters, such as the exploration rate ε, the discount factor γ and the learning rate α. How can one find suitable values for them?

2

i) Consider the 2×2 grid example from the lecture. The example there was deterministic and, hence, the environment always executed the desired action correctly. However, many practical applications are subject to disturbances and, especially in robotics, the intended action does not result into the anticipated state.

In order to adjust for that, change the codes such that a random action is taken in 30 % of all cases (it does not necessary be a viable option though). Then use R to compute the optimal policy from the new state-action table.

4

j) An alternative to Q-learning with ε-greedy action selection is a method named *simulated annealing*. We briefly summarize the method below and then ask to change the method `learnEpisode(...)` accordingly. Run the new method with the `simulateEnvironment(...)` from the lecture slides.

Q-learning with ε-greedy selection relies upon a fixed ε. This, however, might be not beneficial as the agent gains confidence in its state-action function and should do less exploration over time.

In the variant with simulated annealing, the probability to take a certain action is proportional to the Boltzmann distribution

$$e^{Q(s,a)/T}$$

with parameters as follows: T is as a parameter that decreases over time. As a result, a larger T makes all actions more or less equally likely, while smaller values of T prefer the current-best action according to the state-action table. When implementing the variant with simulated annealing, choose an asymptotic function for T.

Hint: refer to the help pages in order to understand all variants of `sample(...)`.

20

k) Extend the Q-learning method from the lecture by a discount factor γ. In addition, plot the reward for the first 20 episodes when testing different parameters.

Hint: the plot might not look as nice as expected as this is still a toy example.