

3. Exercise Sheet

This exercise sheet is rather long and includes two time consuming programming exercises. We are aware that not everybody knows how to program python, therefore, it is okay to work in pairs for Tasks 3 and 5. However, we hope that the examples and the provided documentation helps to understand what's going on. For the discussed methods it is extremely important to see how they work in a real problem scenario. Therefore, some coding is unavoidable. Thanks for your understanding!

Exercise 1 Monte Carlo Method - Understanding

Think about the following questions and give possible solutions to the problems stated:

- In Monte Carlo method, if we start with a deterministic π , some/many (s, a) -pairs will never be visited! How can we make sure that (almost) all pairs are visited?
- The influence of updates shrinks with an increasing number of episodes when using the Monte Carlo Method. Why does this happen? What can we do to resolve this problem?
- The Monte Carlo method needs an end of an episode to calculate the update using the return of this episode. What do we do in never-ending games? How can we solve such problems?
- During the lecture we discussed that the Monte Carlo Method updates its value estimation for each state. We usually discuss state spaces with a finite number of states. What can we do if the state-space is infinite?

Exercise 2 Monte Carlo Method - Application

You want to estimate the time needed to get home. Therefore, you start to record the times needed for each part of your travel. After 3 days you created the following list.

Travel checkpoint	Wednesday	Thursday	Friday
Start at office	6:00	5:30	1:00
Reached the car	6:10	5:35	1:03
Leaving the university complex	6:15	5:45	1:10
Getting on the highway	6:20	5:50	1:15
Leaving the highway	6:45	6:20	2:30
Arrive at home	6:50	6:30	2:35

- Use the Monte Carlo Method to update the value estimate for each episode. Use the first method as initial values for $V(s)$.
- Use the Constant- α Monte Carlo Method to update the value estimate for each episode. Compare the differences of setting α to $\alpha = 0.1$ and $\alpha = 0.5$.

(Hint: use the traveled minutes as reward and the remaining time till you arrive at home as your return function.)

Exercise 3 Monte Carlo Method - Programming

This task can be solved in pairs!

The website of OpenAI offers a python package called "gym". This package includes many interesting problems of the reinforcement learning area. We will use the Cart Pole example from last exercise.

- a) Install Python ≥ 3.5 and the gym package.
 - See <https://gym.openai.com/docs/> for further instructions. In case you don't want to install it locally you can use an online interpreter like: <https://repl.it/> and install the gym package using the console on the right. (However, this will not allow to show plots, which are quite useful to see whats happening.)
- b) Follow the documentation to get the Cart Pole example up and running.
 - There is a known bug that can occur using Windows, where the program crashes after closing the environment. Since this happens after the whole experiment is run, it will not influence this task!
- c) In this example you have two actions available. Either set action to 0 or to 1 depending on the state observation.
- d) The observation is an array containing 4 values. The bounds are:
 - i) cart position: $[-2.4, 2.4]$
 - ii) cart position differential: $[-\infty, \infty]$
 - iii) pole angle: $[-19, 19]$
 - iv) pole angle differential: $[-\infty, \infty]$
- e) Discretize the values of the observations space and store a four-dimensional matrix $V(s)$.
- f) Update the matrix using either the Monte Carlo Method or the Constant- α MC Method.

Exercise 4 Temporal Difference Learning - Application

- a) Apply Temporal Difference Learning ($TD(0)$) with $\alpha = 0.5$ for the example in Task 2 of this exercise sheet.
- b) Explain the differences of the error calculation in Temporal Difference Learning and the Monte Carlo Method.

Exercise 5 Additional Exercise - Temporal Difference Learning - Programming

This task can be solved in pairs!

- See the instructions of Exercise 3 to install the Gym platform.
- Implement Temporal Difference Learning to update the value estimates in $V(s)$ over time.