Online and Reinforcement Learning

2024-2025

Home Assignment 1

Yevgney Seldin **Sadegh Talebi**
Department of Computer Science
University of Copenhagen

The deadline for this assignment is **12 February 2025, 20:59**. You must submit your *individual* solution electronically via the Absalon home page.

A solution consists of:

- A PDF file with detailed answers to the questions, which may include graphs and tables if needed. Do *not* include your full source code in the PDF file, only selected lines if you are asked to do so.
- A .zip file with all your solution source code with comments about the major steps involved in each question (see below). Source code must be submitted in the original file format, not as PDF. The programming language of the course is Python.

Important Remarks:

- IMPORTANT: Do NOT zip the PDF file, since zipped files cannot be opened in *SpeedGrader*. Zipped PDF submissions will not be graded.
- Your PDF report should be self-sufficient. I.e., it should be possible to grade it without opening the .zip file. We do not guarantee opening the .zip file when grading.
- Your code should be structured such that there is one main file (or one main file per question) that we can run to reproduce all the results presented in your report. This main file can, if you like, call other files with functions, classes, etc.
- Handwritten solutions will not be accepted.

1 Find an online learning problem from real life (5 points) [Yevgeny]

Solve Exercise 5.1 in Yevgeny's lecture notes.

2 Follow The Leader (FTL) algorithm for i.i.d. full information games (25 points) [Yevgeny]

Solve Exercise 5.2 in Yevgeny's lecture notes.

3 Improved Parametrization of UCB1 (20 points) [Yevgeny]

Exercise 5.5, Part 2 ["Write a simulation ..."] in Yevgeny's lecture notes. (Part 1 of the exercise will be given later.)

4 The worst case gap for a fixed T (0 points) [Yevgeny] (Optional)

Solve Exercise 5.4 in Yevgeny's lecture notes.

5 Decoupling exploration and exploitation in i.i.d. multiarmed bandits (0 points) [Yevgeny] (Optional)

Solve Exercise 5.3 in Yevgeny's lecture notes.

6 Example of Policies in RiverSwim (14 points) [Sadegh]

Part 1. Consider the following policies defined in the RiverSwim MDP (Figure 1). For each case, determine to which class the policy belongs (i.e., $\Pi^{\mathsf{SD}}, \Pi^{\mathsf{SR}}, \Pi^{\mathsf{HD}}, \Pi^{\mathsf{HR}}$). Provide a short explanation and state any assumptions that you may make.

(i) π_a defined as: Swim to the right if the current state is not 1; otherwise swim to the left.

(ii) π_b defined as: If time slot t is an even integer, swim to the right; otherwise, flip a fair coin, then swim to right (resp. left) if the outcome is 'Head' (resp. 'Tail').

(iii) π_c defined as: At $t = 1$, swim to the left. For $t > 1$, swim to the right if the index of the previous state is odd; otherwise swim to the left. (For $t = 1$, swim to the left.)

(iv) π_d defined as: Flip a fair coin. If the outcome is 'Head' and the current state is either $L-1$ or L, then swim to the right; otherwise swim to the left.

(v) π_e defined as: If it rains, swim to the right; otherwise, swim to the left. (It rains independently of the agent's swimming direction and position.)

Part 2. Make an arbitrary example of a history-dependent deterministic policy in RiverSwim. The policy must not be stationary.

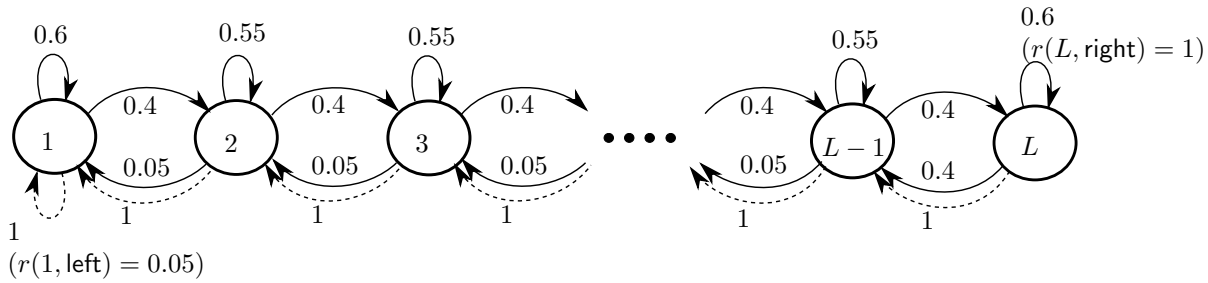

Figure 1: The L-state RiverSwim MDP (Strehl and Littman, 2008)

7 Policy Evaluation in RiverSwim (16 points) [Sadegh]

Consider the 5-state RiverSwim MDP with $\gamma = 0.96$ (Figure 1, with $L = 5$). Consider policy π defined as:

$$\pi(s) = \begin{cases} \text{right} & \text{w.p. } 0.65 \\ \text{left} & \text{w.p. } 0.35 \end{cases} \qquad s = 1, 2, 3,$$
$$\pi(s) = \text{right} \qquad\qquad s = 4, 5.$$

(i) Compute V^π using a Monte Carlo simulation as follows. For each state s, generate n trajectories of length T via interacting with the MDP with starting state $s_1 = s$. Let $h^i = (s_1^i, a_1^i, r_1^i, \ldots, s_T^i, a_T^i, r_T^i)$ denote the i-th trajectory (and note that $s_1^i = s$). Then, $\widehat{V}^\pi(s)$ defined as

$$\widehat{V}^\pi(s) = \frac{1}{n} \sum_{i=1}^{n} \sum_{t=1}^{T} \gamma^{t-1} r_t^i$$

is a Monte Carlo approximation to $V^\pi(s)$. Compute a Monte Carlo approximation to V^π using $T = 300$ and $n = 50$.

(Note: You can make use of the Python implementation of RiverSwim provided in Absalon.)

(ii) (Optional) Compute the exact value of V^π (using direct computation) and compare it with the result of Part (i).

8 Robbing Banks (20 points) [Sadegh]

In this exercise, we model a robber chasing game using the MDP framework.

At time 1, an agent is robbing Bank 1 (see Figure 2). Then, the police gets alerted and starts chasing her from the point PS (Police Station). The agent observes where the police is, and decides in each step either to move up, left, right, down or to stay where she is. Each time the agent is at a bank and the police is not there, she collects a reward of DKK 100,000. If the police catches her, she will lose DKK 10,000, and the game will be restarted: She are brought back to Bank 1, and the police goes back to the PS. The rewards are discounted at a rate $\gamma \in (0, 1)$.

The police always chase the agent but move randomly as follows. More precisely, without loss of generality, assume that the police is on the right of the agent. If she and the police are on the same line, then the police moves up, down and left with probability 1/3. Similarly, when the police and the agent are on the same column, and when she is above the police, then the police moves up, right and left with probability 1/3. Other cases are defined in a similar way. (We assume that walls act as reflectors, similarly to the grid-world.) Finally, when the police and the agent are neither on the same line, nor on the same column, then the police moves up, down, right, or left with probability 1/4. The agent's objective is to maximize her expected cumulative discounted reward.

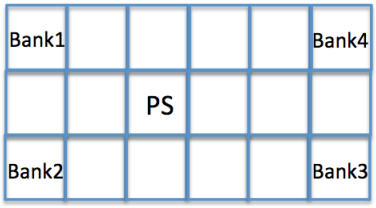

Figure 2: The city

We would like to formulate this problem as an MDP.

(i) Indicate a suitable notion of state (i.e., one that could be used to define an MDP), and indicate the corresponding state-space. Indicate the corresponding action-space.

(ii) Specify the reward function for the chosen notions of state and action.

(iii) For the chosen state-action notion, specify the transition probabilities when the agent is at Bank 1 and the police is at Bank 4.

References

Alexander L Strehl and Michael L Littman. An analysis of model-based interval estimation for Markov decision processes. *Journal of Computer and System Sciences*, 74(8): 1309–1331, 2008.