Online and Reinforcement Learning

2024-2025

Home Assignment 7

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

The deadline for this assignment is **28 March 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 Short Questions `(12 points) [Sadegh]`

Determine whether each statement below is True or False and provide a very brief justification.

1. In a finite average-reward MDP with a finite diameter, the optimal gain does not depend on the starting state.

   □ True □ False

   Justification:

2. Consider a finite average-reward MDP M. If the diameter of M is finite, then it is possible to reach any state from an arbitrary state, regardless of how we choose actions.

   □ True □ False

   Justification:

3. Consider a finite average-reward MDP M. If M is ergodic, then the optimal gain and optimal bias function of M can be determined uniquely.

   □ True   □ False

   Justification:

4. Consider a finite discounted MDP M, and let `Alg` be a PAC-MDP algorithm, with input parameters ε and δ. When executing `Alg` on M, there must exist a finite time step after which all output policies by `Alg` are ε-optimal.

   □ True   □ False

   Justification:

2 Offline Evaluation of Bandit Algorithms - the Practical Part (43 points) [Yevgeny]

1. In "The theoretical part" of Part 2 of Exercise 5.14 in Yevgeny's lecture notes you were asked to modify the UCB1 and EXP3 algorithms to work with importance-weighted data for offline evaluation. Please, write down a pseudo-code of the two modified algorithms (at the same level of detail as the pseudo-codes in the lecture notes). For EXP3, please, write down an anytime version. If you failed to solve the task, you are welcome to consult the reference solution.

2. Now solve "The practical part" of Part 2 of Exercise 5.14 in Yevgeny's lecture notes.

3 Grid-World: Continual and undiscounted (20 points) [Sadegh]

In this exercise, we model a continual grid-world game as an average-reward MDP, and solve it using `VI`.

Consider the 4-room Grid-World MDP shown in Figure 1. It is made of a grid of size 7 × 7, which has accessible states $S = 20$ (after removing the walls). The agent starts in the upper-left corner (shown in red). A reward of 1 is placed in the lower-right corner (shown in yellow), and the rest of the states give no reward. Upon reaching the rewarding state (in yellow), the agent is *teleported* to the initial state regardless of the action, that is, she will be in the upper left corner at the beginning of the following slot. The agent can perform the 4 compass actions going up, left, down, or right (of course, when away from walls). However, the floor is slippery and brings stochasticity to the next-state. Specifically, under each of the aforementioned four actions, she moves in the chosen direction (with probability 0.7), stays in the same state (with probability 0.1), or goes in each of the two perpendicular directions (each with probability 0.1) —this environment is sometimes referred to as the *frozen lake* MDP. Walls act as reflectors, i.e., they cause moving back to the current state. A Python implementation of this MDP is provided in `HA7_gridworld.py`.

We can model this task as an average-reward MDP.

(i) Implement Value Iteration (`VI`) for average-reward MDPs. Your implementation should receive an MDP M and an accuracy parameter ε as input, and output a policy, its gain, and an associated bias function.

(ii) Solve the grid-world task above using Value Iteration (`VI`). Report an optimal policy, the optimal gain $g^\star$, and the span of the optimal bias function, i.e., $\mathrm{sp}(b^\star) := \max_s b^\star(s) - \min_s b^\star(s)$. Furthermore, visualize the derived optimal policy using arrows in the figure or by arranging it using a suitably defined matrix. (We remark that to ensure that `VI` returns an optimal policy, ε must be sufficiently small; here, $\varepsilon = 10^{-6}$ suffices.)

(iii) How do you interpret $1/g^\star$ in this task?

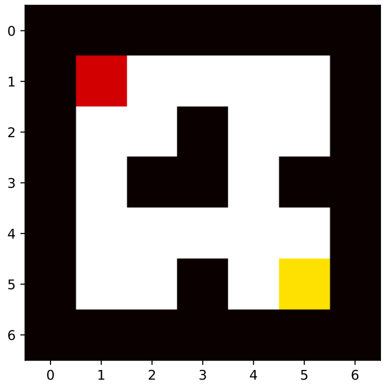

Figure 1: The 4-Room Grid-World MDP

4 An Empirical Evaluation of `UCB Q-learning` `(25 points) [Sadegh]`

In this exercise, we empirically examine the performance of `UCB-QL` (Wang et al., 2020).

Implement `UCB Q-learning` and examine it in 5-state RiverSwim with $\gamma = 0.92$, $\varepsilon = 0.13$, and $\delta = 0.05$ for a time horizon $T = 2 \times 10^6$. In your code, the starting state must be sampled from the uniform distribution. (You may choose, as suggested in the slides, $H = \frac{1}{1-\gamma}\log(1/\varepsilon)$ and $b(k) = \sqrt{\frac{H}{k}\log(SA\log(k+1)/\delta)}$.)

Let $n(t)$ denote the cumulative number of ε-bad time steps as a function of t, namely,

$$n(t) = \sum_{\tau=1}^{t} \mathbb{I}\big\{V^{\pi_\tau}(s_\tau) < V^\star(s_\tau) - \varepsilon\big\}.$$

(i) Plot a sample path of $n(t)$ as a function of time t.

(ii) Plot $n(t)$, averaged over 100 independent runs, along with 95% confidence intervals. (For a set of i.i.d. random variables $\mathbf{X} = \{X_1, \dots, X_n\}$, we define the 95% confidence interval by

$$\Big[\texttt{mean}(\mathbf{X}) - 1.96\frac{\texttt{std}(\mathbf{X})}{\sqrt{n}}, \texttt{mean}(\mathbf{X}) + 1.96\frac{\texttt{std}(\mathbf{X})}{\sqrt{n}}\Big],$$

where `mean` and `std` denote the mean and the standard deviation, respectively.)

References

Yuanhao Wang, Kefan Dong, , Xiaoyu Chen, and Liwei Wang. Q-learning with UCB exploration is sample efficient for infinite-horizon MDP. In *International Conference on Learning Representations*, 2020.