Online and Reinforcement Learning 2023-2024

Christian Igel Yevgeny Seldin Sadegh Talebi

Home Assignment 2

Deadline: 21:00, Wednesday, 21 February 2024

The assignments must be answered individually - each student must write and submit his/her own solution. We encourage you to work on the assignments on your own, but we do not prevent you from discussing the questions in small groups. If you do so, you are requested to list the group partners in your individual submission.

Submission format: *Please, upload your answers in a single* `.pdf` *file and additional* `.zip` *file with all the code that you used to solve the assignment. (The* `.pdf` *should* ***not*** *be part of the* `.zip` *file.)*

IMPORTANT: *We are interested in how you solve the problems, not in the final answers. Please, write down the calculations and comment your solutions.*

1 Short Questions (10 points) [Sadegh]

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

1. In a finite discounted MDP, every possible policy induces a Markov Reward Process.

   ☐ True ☐ False

   Justification:

2. In a finite discounted MDP, if a policy π is optimal, it is necessarily stationary deterministic.

   ☐ True ☐ False

   Justification:

3. Per-step computational complexity of Value Iteration (`VI`) is less than that of Policy Iteration (`PI`).

   ☐ True ☐ False

   Justification:

4. In a finite discounted MDPs, a greedy policy with respect to optimal action-value function, $Q^\star$, corresponds to an optimal policy.

   ☐ True ☐ False

   Justification:

5. Policy Iteration (`PI`) stops after a finite number of iterations but may return a near-optimal policy.

   ☐ True ☐ False

   Justification:

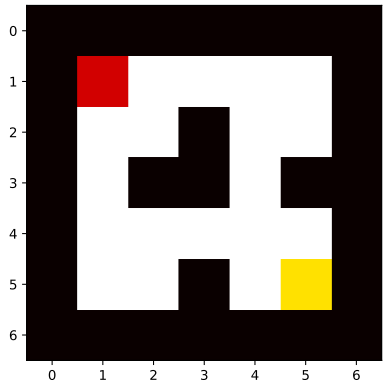

Figure 1: The 4-Room Grid-World MDP

2 Solving a Discounted Grid-World (25 points) [Sadegh]

In this exercise, we model a grid-world game as a discounted MDP, and solve it using `PI` and `VI`.

Consider the 4-room Grid-World MDP shown in Figure 1. It is made of a grid of size 7×7, which has $S = 20$ accessible states (after removing 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. Once in the rewarding state (in yellow), the agent stays there forever (and continues receiving the reward). 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 `HA2_gridworld.py`. Rewards are discounted with rate $\gamma = 0.98$.

We can model this task as an discounted MDP.

(i) Solve the grid-world task above using `PI`. (You may use the Python implementation of `PI` provided in the same file.) Report an optimal policy along with the optimal value function $V^\star$. Furthermore, visualize the derived optimal policy using arrows in the figure or by arranging it using a suitably defined matrix.

(ii) Implement `VI` and use it to solve the grid-world task above. Your implementation should receive an MDP M and an accuracy parameter ε as input, and output a policy and the corresponding value. (Note that to ensure that `VI` returns an optimal policy, ε must be sufficiently small; here, $\varepsilon = 10^{-6}$ suffices.)

(iii) Now consider a variant of the task where upon reaching the state in yellow, the agent is *teleported* to any other $S - 1$ states with equal probability. Repeat Part (ii) for this variant of Grid-World with $\gamma = 0.98$.

(iv) Repeat Part (iii) with $\gamma = 0.995$ and discuss how this new discount impact the value of the state in red.

3 Robbing Banks (15 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 observe 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 is 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 chases the agent, but moves 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 similarly. (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.

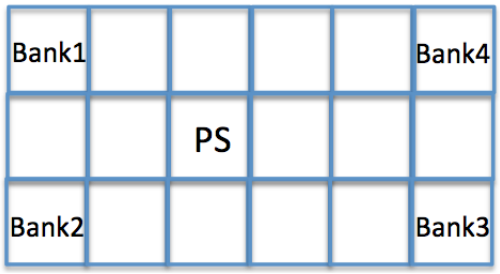

Figure 2: The city

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, specify the transition probabilities when the agent is at Bank 2 and the police is at Bank 3.

4 MDPs with Similar Parameters Have Similar Values (20 points) [Sadegh]

In this exercise, we study a classical result that concerns the difference in value functions between two MDPs that are defined on the same state-action space, and whose transition and reward functions are close in some sense.

Consider two finite discounted MDPs $M_1 = (\mathcal{S}, \mathcal{A}, P_1, R_1, \gamma)$ and $M_2 = (\mathcal{S}, \mathcal{A}, P_2, R_2, \gamma)$. Assume the two reward functions take values in the range $[0, R_{\max}]$. Suppose that for all state-action pairs (s, a),

$$|R_1(s, a) - R_2(s, a)| \leq \alpha, \qquad \|P_1(\cdot|s, a) - P_2(\cdot|s, a)\|_1 \leq \beta$$

for some numbers $\alpha > 0$ and $\beta > 0$. Show that for all stationary deterministic policy π and state-action pair (s, a),

$$|Q_1^\pi(s, a) - Q_2^\pi(s, a)| \leq \frac{\alpha + \gamma R_{\max}\beta}{(1 - \gamma)^2}.$$

5 Policy Evaluation in RiverSwim (30 points) [Sadegh]

In this exercise, we examine policy evaluation in RiverSwim via Temporal Difference (`TD`) and the model-based method (`MB-PE`).

Consider the 5-state RiverSwim MDP with $\gamma = 0.95$ —RiverSwim was introduced in Home Assignment 1 and the lecture note. We are interested in estimating the value function of the following two policies:

1. `data_policy1.csv`, which collects $(s_t, a_t, r_t, s_{t+1}), t = 1, \ldots, 25000$, where a_t is chosen according to a policy π_1 that prescribes always going to the right (i.e., $\pi_1(s) = \mathsf{right}$, for all $s \in \mathcal{S}$).
2. `data_policy2.csv`, which collects $(s_t, a_t, r_t, s_{t+1}), t = 1, \ldots, 25000$, where a_t is chosen according to a randomized policy π_2 defined as follows: For all $s \in \mathcal{S}$, $\pi_2(s) = \begin{cases} \mathsf{right} & \text{w.p. } 0.5 \\ \mathsf{left} & \text{w.p. } 0.5. \end{cases}$

We wish to estimate the value functions of π_1 and π_2 (i.e., $V^{\pi_1}(s)$ and $V^{\pi_2}(s)$ for all $s \in \mathcal{S}$) *only* using the two datasets.

(i) Estimate V^{π_1} and V^{π_2} using both `TD` and `MB-PE`. Run `TD` with the following learning rates: (a) $\alpha_t = 10/(t^{2/3} + 1)$ for $t \geq 1$, and (b) $\alpha'_t = \frac{10}{N_t(s_t)^{7/9}+1}$, where $N_t(s)$ denotes the number of times state s has been visited up to time t, that is $N_t(s) = \sum_{t'=1}^{t-1} \mathbb{I}\{s_{t'} = s\}$.
(Note that you are supposed to use `data_policy1.csv` for V^{π_1}, and `data_policy2.csv` for V^{π_2}.) As output, you should report 6 value functions (i.e., 6 vectors of length 5 each).

(ii) For each policy and each method, plot the error $\|V^\pi - \widehat{V}_t^\pi\|_\infty$ as a function of t, for $\pi \in \{\pi_1, \pi_2\}$, where $V^\pi = (I - \gamma P^\pi)^{-1} r^\pi$ denotes the true value function of π. (You must report 6 curves, and preferably in the same figure, with a logarithmic y-axis.)

(iii) In view of the obtained results, how do you compare the used methods? Briefly discuss the implication of the theoretical convergence guarantees for `TD` and `MB-PE` for these cases. (Note: The last part involves, among other things, verifying the Robbins-Monro conditions for $(\alpha_t)_{t\geq 1}$ and $(\alpha'_t)_{t\geq 1}$.)

6 Output of `VI` (optional) [Sadegh]

Consider the Value Iteration (`VI`) algorithm. Argue that if the input accuracy ε is smaller than a cutoff MDP-dependent threshold $\varepsilon_{\mathsf{cutoff}}$, the resultant policy is optimal. Derive $\varepsilon_{\mathsf{cutoff}}$ for a given MDP M.

Good luck!
Christian, Yevgeny, and Sadegh