Released: 17th Feb Deadline: 2nd Mar

Problem Set 3

INF266 - Reinforcement Learning
Spring 2025

IMPORTANT: Submission of this problem set is **compulsory** to be admitted to the final exam and graduate from the course.

Part I. Problems

In the first part (*Problems*), you are required to provide answers for all the *obligatory* questions, that is, questions marked by a star (*). The remaining questions are optional; however, you are invited to solve them because: (i) they are stepping-stones towards the obligatory questions; (ii) they are similar to the questions you will be asked at the oral exam.

1. Monte Carlo methods

Exercise1 Monte Carlo Methods

Consider the application of a Monte Carlo method to the computation of the value of π.

1. Consider a square with a side length of 1. Define analytically the ratio X between the area of the circle inscribed in the square and the square itself.
2. Equate the analytical ratio X that you have computed to an empirical ratio $\hat{X}$ you will compute via Monte Carlo, and isolate π.
3. Sample 10000 pairs of numbers (x_1, x_2) generated as follows:
   - $x_1 \sim \texttt{Unif}[0,1]$ are i.i.d. samples from a uniform distribution over $[0,1]$;
   - $x_2 \sim \texttt{Unif}[0,1]$ are i.i.d. samples from a uniform distribution over $[0,1]$.

4. Estimate $\hat{X}$ from the samples, by taking the ratio of points falling within and without the circle.
5. Estimate π.
6. Generate a new dataset of 10000 pairs of numbers (x_1, x_2) generated as follows:
    - $x_1 \sim \texttt{Unif}[0, 1]$ are i.i.d. samples from a uniform distribution over $[0, 1]$;
    - $x_2 = x_1 + \texttt{Unif}[-0.1, 0.1]$, that is, x_2 is equal to x_1 plus a i.i.d. random noise sampled from a uniform distribution over $[-0.1, 0.1]$.
7. Estimate the new $\hat{X}$ and π.
8. (*) How do the two estimates change and why?

Exercise2 MC and MABs

You previously showed that MABs can be modelled as MDPs. Now, we want to show that the ϵ-greedy algorithm for MABs can be seen as an instance of an MC algorithm. Specifically:

1. (*) What sort of problem are we solving in a MAB: prediction or control?
2. (*) In a MAB you compute expected rewards of arms. Would this correspond to a state-value function or an action-value function?
3. (*) What would correspond to a *trajectory* in a MAB? What would be a collection of *trajectories*?
4. (*) To what MC family of algorithms would you relate the MAB algorithms we studied: MC prediction or MC control?
5. (*) What would correspond to *MC prediction* in a MAB?
6. (*) What is the purpose of ϵ in MABs and in MC control?
7. (*) Could we reduce a standard RL problem solved by MC to a MAB problem? If so, what would we lose?

Exercise3 Importance Sampling

Consider the old slot machine you played with in the first problem assignment. Specifically: the first slot machine R takes values on the domain $[0\$, 5\$, 10\$, 20\$]$ with probabilities $[.35, .3, .25, .1]$; the second slot machine S takes values on the same domain $[0\$, 5\$, 10\$, 20\$]$ but with probabilities $[.3, .35, .3, .05]$.

1. As ground truth to use for reference, compute the exact expected values $E[R]$ and $E[S]$.
2. Implement the two slot machines as functions from which you can get samples.
3. Play with slot machine S. Sample 10000 rewards and compute the empirical expectation $\hat{E}[S]$.
4. (*) Compute the ratio ρ between the distribution probability of R and S.

5. (*) Suppose now that you only have the samples from S that you have already collected. Suppose, also, that someone let you know the value of ρ. Use importance sampling to compute the empirical expectation of $\hat{E}[R]$ from the samples of S and ρ.
6. Suppose now that you play with a machine S' with a reward distribution of $[.25, .25, .25, .25]$. Sample again 10000 rewards, evaluate $\hat{E}[S']$, compute the ratio ρ between the distribution probability of R and S', and estimate $\hat{E}[R]$ via importance sampling from the samples of S' and ρ.
7. Suppose now that you play with a machine S'' with a reward distribution of $[.001, .499, .499, .001]$. Sample again 10000 rewards, evaluate $\hat{E}[S'']$, compute the ratio ρ between the distribution probability of R and S'', and estimate $\hat{E}[R]$ via importance sampling from the samples of S'' and ρ.
8. (*) How do the results you computed differ? What explains the differences?

Exercise4 Q-learning

In class we introduced Q-learning as an off-policy algorithm.

1. (*) Given that we do off-policy control, why do we not need to rely on importance ratio in Q-learning?

Part II. Report

In the second part (*Report*), you are required to provide a report of maximum 2 pages describing the experiments you have run and analyzed to answer the *obligatory* questions, that is, questions marked by a star (*). The remaining questions are optional; however, you are invited to solve them because: (i) they are stepping-stones towards the obligatory questions; (ii) they are similar to the questions you will be asked at the oral exam. You will be evaluated only on your report. Review the documents on `coding tips` and `writing tips` for advices on how to tackle this part.

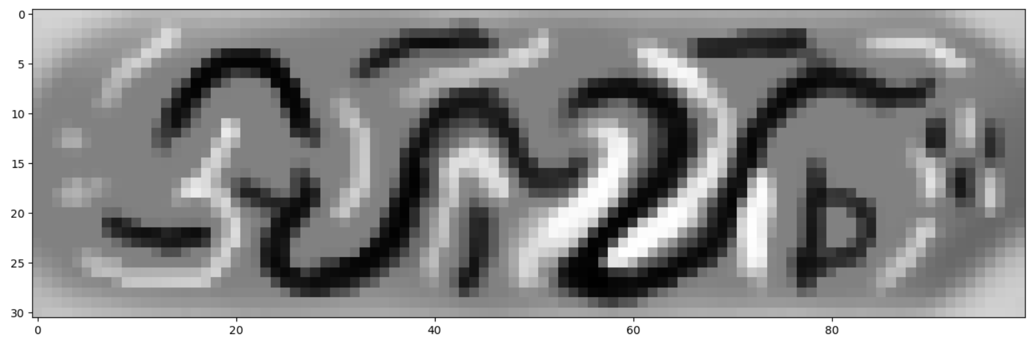
Exercise5 MC and TD

Consider again the problem of the robot searching an optimal path down the side of a mountain which you worked on in the previous problem set.

After your previous study, the robot has now been deployed on the side of a new mountain. Moreover its actuators have been improved allowing it to move in more directions (*larger action space*), although they are now a little bit less reliable (*stochastic transition function*).

We summarize below facts about the dynamics of the environment from the previous assignment, as well as the new features:

- *Objective:* the aim of the robot is to find the fastest route to the root of the mountain, avoiding to traverse terrain that is too rough and uneven.
- *Representation:* the side of the mountain is represented as a *gridworld* made up by 31 rows and 100 columns (Notice that the encoding is the same as in the last problem assignment).
- *Start state:* the robot will start somewhere in the leftmost column (top of the mountain).
- *End state:* the robot aims at reaching the rightmost column (bottom of the mountain).
- *Actions:* the robot can move in five directions on the grid (forward, forward up, forward down, up and down).
- *Transitions:* actions are not always successful, and the robot may stochastically end up in a different state from the one it tried to achieve.
- *Reward:* each square has a roughness index providing a reward signal of how fast the robot can move through that square.
- *Violation of borders:* if the robot attempt to move beyond the map in the top or bottom direction, it will fail to escape and just remain where it was.



A company, SpaceY, has already studied this problem and run its own robot according to their own proprietary policy π. However, they have made available a set of trajectories that they have collected while using π in the file `trajectories.pickle`. Notice that, because the finite amount of battery time, all the episodes ended after a finite number of steps.

1. Load the trajectories and perform MC evaluation for $v_\pi(s)$.
2. Load the trajectories and perform MC evaluation for $q_\pi(a, s)$.
3. Plot the state-value function $v_\pi(s)$.
4. (*) Can you perform MC improvement? If so, perform it and explain whether the new policy is optimal or not.
5. (*) Explain and comment the result you obtained. Are all the trajectories equally useful for performing MC evaluation and improvement?
6. (*) Can you perform MC control? If so, perform it and explain whether the new policy is optimal or not.

Your company has now received the robot of SpaceY, together with a simulator for the environment. Load the first version of the environment `mountain` (`v1`) (or `mountain_lite`).

7. (*) Choose a starting policy for the robot, and justify your choice.
8. Perform MC control and learn an optimal policy for the robot.
9. (*) Did you manage to learn an optimal policy? What challenges did you find (e.g.: computational time, starting policy)?

An even more advanced company, SpaceZ, developed a robot with more degrees of freedom, able to move in all directions (all four cardinal directions plus the four diagonal directions). You have been given a prototype of this robot and you want to train in the same environment as before. Load the second version of the environment `mountain` (`v2`) (or `mountain_lite`).

10. Attempt to learn an optimal policy for the SpaceZ agent using MC control. Does it converge? How does the change in the environment affect the computational cost of your training?

Your company decides to try a more efficient approach based on a control algorithm that may work in real-time, without waiting for episodes to terminate.

11. Implement SARSA(0) and use it to learn an optimal policy for the robot.

12. (*) Choose a meaningful metric, and compare the learning process using MC and SARSA(0)
13. Implement SARSA(n) and use it to learn an optimal policy for the robot, considering at least three different values for n.
14. (*) Choose a meaningful metric, and compare the learning process using SARSA(0) and SARSA(n) with the values of n that you chose.
15. Implement Q-learning and use it to learn an optimal policy for the robot.
16. (*) Choose a meaningful metric, and compare the learning process using Q-learning against the previous algorithms.