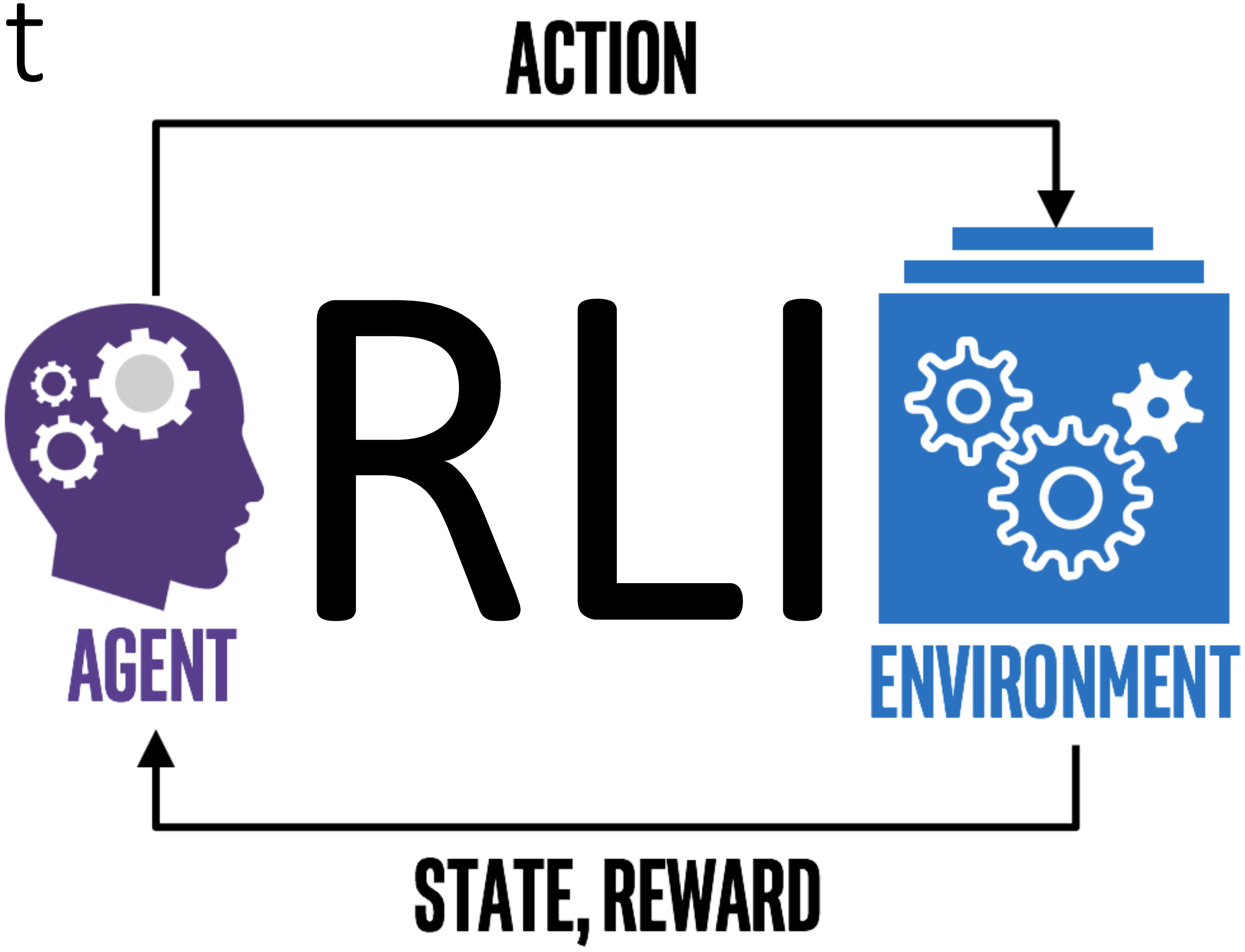
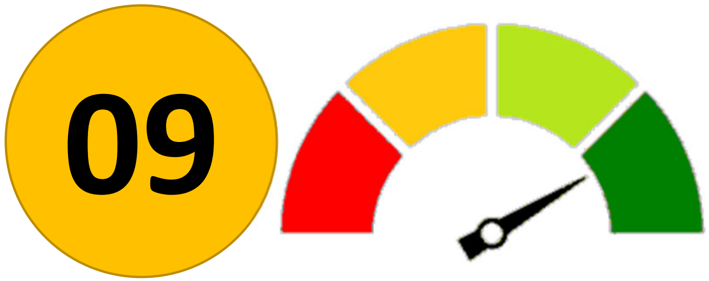
Reinforcement Learning Introduction





José Manuel Rey, IE

Reinforcement Learning Introduction Assignment 09.00

Date: Feb/5 2025

Due Date: Mar/7 2025 [SESSION 17]

Ongoing grading (maximum)

50 Individual Assignment [mini-groups]

Assignment Description Overview

This assignment consists of 2 parts/questions:

Assignment 09.01 → "Jack's Car Rental Problem – Version 1" [25 points]

- jack_car_problem_v0.ipynb

Assignment 09.02 → "Blackjack Off-Policy" [25 points][7 questions/3.5 pts per question]

- Off-Policy MC Control with Weighted Importance Sampling.ipynb

For each part/question you should elaborate a self-contained ".ipynb" file (derived from the documents delivered with the assignment), providing (when required) a clear concise explanation of the conceptual approach, the diagrams or models representing the problem, comments for the code used and the explanation of the solution

(*) Note: the files blackjack_gym.py, plotting.py, are included in the assignment files as they are used as external libraries by the jupyter notebooks [no changes should be made in any of them]

SUBMIT YOUR WORK BY ZIPPING TOGETHER THE 2 NOTEBOOKS (COMPLETED WITH YOUR ANSWERS) IN A SINGLE SUBMISSION FILE NAMED:

"RLI_09_00 – mini-group number.zip" (see the attached .txt for the mini-groups)

Administrative and additional notes

- Find attached (in the zip file) the configuration of the mini-groups for the assignment:
  - →RLI_2024-25 - ASSIGNMENT RLI_09_00 TEAMS.txt
  - →Note: you will find two versions of this file, either (A) or (B). You will be listed in the one corresponding to your IE class

  Only submit ONE zip file per group. Choose ONE member of each group for that submission. In case you send more than one, ensure that all the submissions by the different members are identical, as the system will choose just ONLY one of them (randomly) for the review and evaluation of the whole group.

- As usual, the code should run without warnings or errors. All additional required files should be included in the ZIP file, and if you were using any "novel" additional external library for your work, clearly indicate a brief description of its purpose, the version used and the **"`!pip install <package>`"** required for its installation

Assignment 09.01 Description Overview Jack’s Car Rental Problem – version 1

You must adapt the solutions obtained for the original “Jack’s Car Rental Problem” [Example 4.2: Jack’s Car Rental in Reinforcement Learning - 2nd ed Richard Sutton and Andrew Barto] (we will call them version 0) to the new conditions proposed in Sutton & Barto “Exercise 4.7 (programming)” (we will call them version 1)

Use the jupyter notebook created for solving the first problem (*jack_car_problem_v0.ipynb*) and adapt it conveniently for the second problem (name you new *document as jack_car_problem_v1.ipynb*)

The description of each problem is repeated below in the following slides

Jack's Car Rental (Version 0)

Here are key specific points about the problem:

- 2 locations A, B
- Each location can only hold 20 cars.
- Every time a car is rented, we earn $10 (Reward)
- Every time we move a car overnight to another location, it costs us $2 (Negative Reward).
- The maximum number of cars we can move overnight is 5 (Action).
- The number of cars requested and returned at each location (n) on any given day are Poisson random variables.
- The expected number (lambda) of **rental requests at the first and second location is 3 and 4 respectively.**
- The expected number of **rental returns at the first and second location is 3 and 2 respectively.**
- Our discount rate for future returns, (γ), is 0.9.
- The time step are days (thus, one step in an iteration can be considered a full day), the state is the number of cars at each location at the end of the day, and the actions are the net number of cars moved between the two locations overnight.

Jack's Car Rental (Version 1)

Let’s see what happens if we add some non-linearities [*as they are costs or conditions not directly proportional to the number of cars moved, as they were the original ones stated for the problem in version 0*] like the following to the problem above:

- One of Jack’s employees at the first location rides a bus home each night and lives near the second location. He is happy to shuttle one car to the second location for free.
- Each additional car still costs $2, as do all cars moved in the other direction.
- In addition, Jack has limited parking space at each location. If more than 10 cars are kept overnight at a location (after any moving of cars), then an additional cost of $4 must be incurred to use a second parking lot (independent of how many cars are kept there) → We will have one more reward of -$4 for the second parking lot if needed

Assignment 09.02 Description Overview Blackjack Off-Policy

Complete the code appropriately in the attached notebook (*Off-Policy MC Control with Weighted Importance Sampling.ipynb*) in order to create a working version of an implementation on an Off-Policy Monte Carlo Control method with Weighted Importance Sampling

ASSIGNMENT (Technical Note) blackjack_gym.py

- This exercise also introduces the use of OpenAI compliant environments (OpenAI Gymnasium) [blackjack_gym.py]
- For compatibility issues we will be using **OpenAI Gymnasium** version >0.26.0. The current version (as of Jan/2025) is 1.0.0.
- The suggested (minimalistic) installation is: `pip install gymnasium[box2d]` or just `pip install gymnasium`
- In the case that you were already using some other older versions of OpenAI Gym (<=0.21.00), and want to keep it active (for compatibility with some older code you want to run) create and use a virtual python environment to avoid conflicts