Reinforcement Learning

Course Instructors: Prof. Nicolò Cesa-Bianchi and Prof. Alfio Ferrara

Course Assistants: Elisabetta Rocchetti (PhD student), Luigi Foscari (PhD student)

Data Science and Economics Master Degree, Università degli Studi di Milano - Academic Year 2024-25

Project Descriptions

The final project consists in the implementation of a RL solution to a simple problem. Students have to provide access to a GitHub repository containing the code and reproducible results. Finally, the project will be discussed after a 10 minutes presentation in English with slides. The project discussion will be set by appointment, according to the following procedure:

1. Subscribe one of the official exam dates
2. Submit the link to your GitHub repository as soon as the project is ready
3. Contact Prof. Ferrara and set and appointment for the discussion
4. The appointment for the exam part on theory should be taken directly with Prof. Cesa-Bianchi

The project is an individual work.

All the projects are presented in a simplified version. There are **no constraints on the technology**. You can use the gymnasium environment or any other framework for modeling environments and agents.

Focus of the project is to **evaluate the student understanding of the problems** and the exam will include **questions about theory** starting from the proposed project.

Generally speaking, the project proposals are just **ideas that can be modified or just taken as suggestions** for developing your own objectives.

Try to have fun!

Project 1: The Chaotic Chef's Quest for the Perfect Meal

Main Focus

Comparison between tabular and approximate RL methods.

Problem Description

In this game, the agent plays as a **chef** who explores the city markets in order to find the best ingredients from a limited supply in order to **cook delicious dishes**. Different recipes require different combinations of ingredients, and some combinations are more valuable than others. However, certain ingredients are not fitting well together, causing failures or penalties.

The city map consists of a **5x5 grid** where each cell represents a market selling an ingredient. The agent moves around the grid, collecting ingredients, and must decide when to stop collecting and attempt to create a dish. The goal is to maximize the total dish value while minimizing wasted ingredients.

Challenging variant

As an interesting variant, we could introduce a monetary budget of the chef, different costs for the ingredients, and a revenue for the dishes, depending on their quality. In this case, the goal could be to maximize the chef wealth after a certain number of rounds.

Tasks

- Model the environment as an MDP, defining states, actions, and rewards.
- Implement two RL agents: one using **Q-learning or Sarsa with a tabular representation** and another using **function approximation (e.g., linear aproximation or neural networks)**.
- Compare the learning performance and policy efficiency between the two approaches.
- Analyze how state representation affects convergence speed and sample efficiency.

Project 2: The Gambler

Main Focus

Linear solutions and feature design.

Problem Description

Implement a reinforcement learning agent to play a simplified version of Blackjack. The agent must decide whether to "hit" or "stand" based on the game state.

Challenging variant

The agent is an expert gambler, so cheating is always an option, but there are risks! The idea is that a possible action is always to peek at the next card in the deck. If this is successful you gain information that can be used to choose if play "hit" or "stand". But if the agent is caught there is a penalty.

Tasks

- Model the environment, defining an appropriate state representation (e.g., player hand value, dealer's visible card, ace usability).
- Implement an RL agent using **Linear Function Approximation (LFA)** to estimate state-action values.

- Explore different **feature representations**, including binary features (card presence) and polynomial features.
- Evaluate the impact of different feature sets on learning speed and policy performance.

Project 3: Primal Hunt

Main Focus

DQN and the importance of experience replay and target network.

Problem Description

The agent is a primitive hunter trying to gather enough food to survive. The environment is dangerous due to the presence of wild animals, hostile tribes and natural obstacles such as rivers, mountains, forests. Hunting requires energy and there are risks to be injuried. The hunting goals are different: vegetables and fruit are easy and require low energy, but provide low reward; small animals are moderately risky and provide a moderate reward; big animals provide a large reward but they huge risks. The goal is to come back home at night with enough food to survive, recovering all the energy used during the day.

Challenging variant

The environment occasionally changes as wild animals and hostile tribes move across the map, and shifting weather conditions may alter natural obstacles. However, the agent can learn to recognize the signs of danger.

Tasks

- Implement the environment and define a suitable state-action space.
- Train an RL agent using **Deep Q-Networks (DQN)** to learn optimal paths.
- Conduct an **ablation study** to analyze the importance of **experience replay** and **target networks** in stabilizing learning.
- Compare training curves with and without these components to demonstrate their impact.

Project 4: Push Your Luck!

Main Focus

Policy Gradient Methods.

Problem Description

In this space exploration game, the agent rolls a pool of dices to explore a tricky dungeon looking for treasures. Each roll result is associated with a different outcome in terms of the size of the treasury discovered but also in terms of the risks associated with exploration (e.g., being injuried, game over, loose part of the cumulated treasures). At each game turn, the agent needs to take a decision between stopping exploration and come back home with the treasuries cumulated or push its luck, rolling the dices one more time.

Challenging variant

Taking risky decisions may be beneficial in terms of experience. The negative effects of certain dice roll outcomes could become slightly less severe as those outcomes occur more frequently. In other words, each time the agent experiences a certain result, that result changes and becomes a little less negative.

Tasks

- Model the problem as a Markov Decision Process (MDP).
- Implement a **policy gradient-based RL agent** (e.g., REINFORCE or Actor-Critic).
- Investigate the effect of different reward structures and baseline techniques on learning efficiency.
- Compare policy gradient-based learning with a value-based approach (e.g., Q-learning).

Project 5: Harvest and Thrive

Main Focus

Continuous action spaces.

Problem Description

In this resource management game, the agent must allocate resources such as water, fertilizer, and labor to optimize crop growth while managing limited supplies in a farm. Each turn, the environment dynamically changes with fluctuating weather conditions, soil health variations, and shifting market prices, requiring the agent to adapt its strategy accordingly. Overwatering may lead to crop diseases, while under-fertilization slows growth and reduces yield, depending on the crops and the weather and soil conditions. At the end of each year, the farm production is sold and this represents the reward that the agent wants to maximize in the long-term (i.e., after a certain number of years in an episodic setting).

Challenging variant

The agent must also decide when to sell harvested crops, balancing immediate profits with the potential for better market prices in the future. The economic aspect introduces a layer of complexity, as reinvesting in farm improvements or expanding crop diversity can lead to long-term success but requires careful financial planning. Additionally, over-farming can degrade soil quality, making sustainability a key factor in optimizing long-term yield.

Tasks

- Model the environment with a **continuous action space**, requiring **policy-based RL approaches**.
- Compare different exploration strategies.

- Evaluate how different action space constraints affect learning performance.

Project 6: King Robert's birthday tournament

Main Focus

Multi-agent reinforcement learning.

Problem Description

In the great hall of King's Landing, Robert Baratheon, Lord of the Seven Kingdoms and Protector of the Realm, decrees a yearly tournament in celebration of his birth. The rival lords of the realm are summoned to send forth their bravest knights, who shall face one another in deadly combat. The winner shall bring glory and riches to their house, but the defeated will meet a gruesome end, their names quickly fading into obscurity. By the command of young Prince Joffrey, each duel must end in death.

As a newly appointed lord of the realm, you must choose a knight to represent your house in the tournament. Each year, you will select a champion to face the warriors of House Frey. Choose wisely, for if your champion falls in battle, they are lost to you forever. However, if they return victorious, their strength will grow, and they will be a mightier force for the years to come.

The first year of the tournament brings a cruel decision. Both your house and House Frey have several knights to choose from. As long as either house can muster a champion fit for battle, they may continue to send warriors into the fray. But beware, for a lost knight cannot be sent again, and each defeat brings your house closer to ruin.

Challenging variant

In a twist of fate, Prince Joffrey soon grows bored of the bloodshed and loses interest in the tournament. Now, the duels no longer must end in death. A defeated champion may return to the roster. In this new era, champions who are wounded in battle — whether victorious or defeated — may incur minor injuries that keep them from fighting for a couple of years. Should your knight be so unfortunate, you must wait for their recovery or choose another champion for the next tournament. The uncertainty of fate looms larger than ever, and the balance of power may shift with every wound, victory, or defeat.

Tasks

- Model the tournament as a 2v2 zero-sum game in which the two players (houses) have a set of actions (champtions), each pair of chosen actions has a possibility of granting victory to either house.
- Consider different learning procedures.