COPS Summer of Code 2025

Intelligence Guild

Club Of Programmers, IIT (BHU) Varanasi

Markov Decision Process

2 June – 8 June 2025

Official IG Website: https://cops-iitbhu.github.io/IG-website/

All deadlines are strict. No extensions will be granted.

Introduction

COPS Summer of Code (CSOC) is a flagship initiative under the Club Of Programmers, IIT (BHU) Varanasi, with all verticals contributing through focused tracks. This document outlines the prerequisites for the Intelligence Guild vertical.

Modules will be released weekly and from time to time. **Adhere strictly to deadlines**. Submissions will be evaluated on approach, technical correctness, and clarity. The most technically accurate solution may not necessarily be the one chosen; clarity of thought and a well-reasoned approach will be valued more.

Communities

All communication for the programme will be conducted strictly via Discord. Do not reach out through other channels. Resources and updates will be posted on Github, and all notifications will be made via Discord.

Final Report

A concise report may be submitted along with your final assignment. While **not mandatory**, it may strengthen your overall evaluation. Reports must be written in LaTeX and submitted in PDF format only. We are not interested in surface-level descriptions — focus strictly on your analysis, approach, and reasoning. The report itself constitutes the final assignment. No additional files are to be submitted. Refer to the Assignment section for details. Submit your report here.

Contact Details

In case of any doubts, clarifications, or guidance, you can contact one of us. We request that you stick to Discord as the preferred mode of communication for all the questions that you have as it will also benefit others. However, you can reach out to us through other means in case we fail to respond on Discord.

- Yashashwi Singhania - 7905584242

Markov Decision Process

MDPs are a mathematically idealized formulation of the reinforcement learning problem, enabling precise theoretical analysis.

- Chapter 3 from the RL textbook (rigorous and mathematically dense, may be challenging to follow independently) - Sutton and Barto
- Emphasize theoretical understanding and the underlying mathematics, as the conceptual difficulty will increase rapidly. Use online resources to clarify doubts while reading.

Dynamic Programming

Dynamic programming (DP) refers to a class of algorithms used to compute optimal policies when a perfect model of the environment is available in the form of a Markov decision process (MDP). While classical DP algorithms have **limited practical utility** in reinforcement learning due to their reliance on a known model and their computational cost, they remain foundational from a theoretical perspective.

- DeepMind x UCL by David Silver - Videos 1–4 provide coverage up to dynamic programming.
- Chapter 4 from Sutton and Barto

Implementations

Now that you've understood the theory, the question is—how do we implement these algorithms? To do so, we require an environment that provides feedback to train the agent.

- Gymnasium, a maintained fork of OpenAI's Gym library, offers a graphical interface for representing RL problems.

Note: You are encouraged to look up topics you find unclear. This module will become substantially more involved as you progress.

Assignment

Objective

There will be multiple tasks to analyze and complete. Approach each of them independently, focusing on developing your own ideas and implementations. It is entirely acceptable, and expected, to encounter difficulties during initial attempts. Prioritize substance and clarity in your work, avoid filler text, as it tends to weaken the overall quality of your submission.

You are encouraged to study the theoretical solutions of the problems after giving them a try. **Do not use AI tools to frame your answers**, as doing so undermines

the purpose of the exercise. AI may be used for conceptual understanding (though this is discouraged), but if you do refer to it, clearly indicate where and how it was used. Likewise, mention any instances where you consulted external resources or references. These mentions will only make your assignment stronger.

The solutions to these problems are already known to us. The objective here is to assess your understanding and help you develop intuition.

Project Tasks

Derive optimal bellman equations for value and action value functions

- What is optimality?
- What is expectation and how does it relate to bellman equations?
- Solve exercise 3.25 - 3.29 from Sutton and Barto

Prove convergence of policy and value iterations

- Try to build some intuition on how are they working and why should they work
- What actually is GPI?
- What is the complete and rigorous proof of their convergence?
- What is the time and memory complexity for each of the algorithms? Where are each of them particularly useful?

Implement DP algorithms on Frozen Lake from Gymnasium

- Use the Frozen Lake environment (or a similar one with discrete action spaces) from the Gymnasium library.
- Create a custom version of the Frozen Lake environment. Additionally, design a second custom environment with an expanded state or action space. Register both environments.
- Use dynamic programming to find an optimal policy.
- Evaluate and compare the performance of these methods across all three environments (original, custom, and expanded custom). Compare key metrics such as convergence time and average episode length (not limited to these, find some other metrics which can be useful for comparison) .

Bonus Tasks:

- This is absolutely not compulsory and very understandable if you didn't understand how to do it. Just be honest.
- Solve all the exercises from chapters 3 and 4 from the book. Don't worry if you aren't able to, they are meant to be hard. Give them a go and mention any progress made for each of them.

Submission Guidelines

- Create a GitHub repository named `<roll_number>-CSOC-IG` (e.g., `23014019-CSOC-IG`)
- Repository organization:
  - A folder named Reinforcement-Learning/MDP/ containing all source code implementations, the next week's topics in this track will be maintained under the subdir Reinforcement-Learning. Please ensure this.
  - The final report in PDF format, authored using LaTeX, use OverLeaf.

  Everything must be in the github repo itself.
- Submit the repository link via the provided Google Form here
- **Deadlines are strict and will not be extended**

Final Remarks

This module establishes the mathematical and algorithmic foundations essential for understanding reinforcement learning. Mastery of MDPs and dynamic programming will serve as a critical base for all subsequent modules. Ensure your understanding is precise, as future topics will build on these concepts without simplification.

***Adios**, and keep learning!*