COPS Summer of Code 2025

Intelligence Guild

Club Of Programmers, IIT (BHU) Varanasi

Monte Carlo and TD Learning

10 June – 16 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

Monte Carlo

Unlike previous examples, here we do not assume full information of the environment, and only account for the experiences from actual interactions with the environments.

- Chapter 5 from the RL textbook - 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.

TD Learning - Q Learning and SARSA

Temporal Difference (TD) Learning is an approach that updates value estimates based on observed transitions without requiring a full model of the environment. Two key algorithms under this framework are Q-Learning and SARSA. Q-Learning is an off-policy method that learns the optimal action-value function by evaluating the best possible action at each step, regardless of the agent's actual behavior. In contrast, SARSA is an on-policy algorithm that updates its action-value estimates based on the action actually taken by the agent, resulting in behavior that aligns more closely with the current policy.

- DeepMind x UCL by David Silver - Videos 5-6.
- Chapter 6 from Sutton and Barto

Assignment

Objective

Given the struggles from last week, I have reduced the number of tasks a lot. This shouldn't take much time.

Project Tasks

Task 1 : Implementing the algorithms

- Last week you guys built environments and used built gym envs. For this task just reuse those and apply the three algorithms. I saw people were using very tiny environments, expand them to large base.
- Compare all the methods learnt in the previous weeks and this week.

Bonus Task 1:

NOTE: This is completely optional but you may get brownie points depending upon the quality of your submission.

- Build the cliff walking experiment from scratch on your own and visualise the rewards obtained during training and post training of both Q-Learning and SARSA models.

Bonus Task 2:

NOTE: This is completely optional but you may get brownie points depending upon the quality of your submission.

- Last year in CSOC I built the blackjack example from RL book (Example 5.1). I wanted to implement a few more novel actions which were not in the original example. Implement a splitting action often used in these games and try to visualise the policy (state of dealer vs state of our hands matrix and each cell would show the respective action.)
- Given you all should have the Monte Carlo algorithm by now, try to implement all the different types of the algorithms on this one.
- You are also free to do anymore changes as preferred, just document every thought process.

If you have understood and done all the previous required tasks till now, you should be able to finish the bonus tasks by Monday as well. The final task after this should be interesting and quite a learning experience for you all ;)

Submission Guidelines

- Create a GitHub repository named `<roll_number>-CSOC-IG` (e.g., `23014019-CSOC-IG`)
- Repository organization:
  - A folder named Reinforcement-Learning/Monte Carlo and TD Learning/ containing all source code implementations. 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**

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