Monte Carlo and TD Learning

Anshul Choudhary

16 June, 2025

1 Introduction

In this task we use 4 algorithms: **Monte Carlo, SARSA, Q-learning**, and **Policy Iteration** on two environments; **Frozen Lake** and **Cliff Walk**. We are using a dynamic programming (DP) algorithm so that we can compare it with other algorithms which do not assume complete knowledge of the environment. In all three of these algorithms, I have implemented ϵ**-soft** policies.

2 Analysis of Results

2.1 Frozen Lake

Algorithm	Time Taken (s)	Avg. Episode Length	Avg. Reward
Monte Carlo	63.19	65.22	1.0
SARSA	9.31	68.37	0.9912
Q-learning	11.60	65.45	1.0
Policy Iteration	0.08	59.38	1.0

Table 1: Comparison of algorithms on Frozen Lake (10x10)

The environment used was a **10×10** randomly generated Frozen Lake environment with **slipperiness enabled**.

We can observe that the average rewards are almost similar—actually, they are all learning the optimal policy. The small difference in SARSA is just due to the **stochastic nature** of the environment.

Policy Iteration is the fastest among all. This is because, unlike the other three, it has complete knowledge of the environment's dynamics from the start, while the others have to learn by running episodes. SARSA and Q-learning take less time than Monte Carlo; the reason for this is that both SARSA and Q-learning **bootstrap**, i.e., they use information learnt in other states to update the value of the current state. This is especially helpful in environments which follow the **Markov property**.

2.2 Cliff Walking

Algorithm	**Time Taken (s)**	**Avg. Episode Length**	**Avg. Reward**
SARSA	0.92	15.0	-15.0
Q-learning	0.60	13.0	-13.0
Policy Iteration	0.017	13.0	-13.0

Table 2: Comparison of algorithms on Cliff Walk

Just like in the book, **SARSA has learnt the suboptimal but safe policy** which takes a longer route. On the other hand, **Q-learning and Policy Iteration** have learnt the optimal policy. A difference here is that in the book, Q-learning was performing worse than SARSA; that was because they were using the ϵ-greedy policy. But when testing, there is no need for the ϵ factor anymore since learning is already done and now we don't need to explore. Therefore, both Q-learning and Policy Iteration follow the shortest path to the goal.