Comparison of Q-Learning and SARSA for Taxi-v3 Game

Introduction

In this project, we implemented reinforcement learning agents to solve the OpenAI Gym's Taxi-v3 environment, a grid-based game where the agent aims to pick up and drop off a passenger at specific locations while minimizing actions taken. The implemented algorithms include Q-Learning, SARSA, and an extension of Q-Learning with epsilon scheduling to analyze their learning efficiencies and performances.

Methods

We utilized three algorithms for the agent: Q-Learning, SARSA, and Q-Learning with epsilon scheduling. The environment was run for 1000 episodes for each algorithm. All agents were developed with the following hyperparameters: learning rate (α) of 0.1 or 0.5, discount factor (γ) of 0.99, and an epsilon-greedy policy with varying epsilon values for exploration.

The Q-Learning agent utilized a greedy approach based on learned Q-values, while SARSA (State-Action-Reward-State-Action) updated the Q-values by considering the next action chosen by the agent in each update. Additionally, a Q-Learning agent with epsilon scheduling was implemented to gradually reduce exploration over time, improving exploitation after sufficient learning.

Results

The learning curves for all three algorithms are shown in Figure 1. Q-Learning showed a slightly faster convergence compared to SARSA, which aligns with its off-policy nature that encourages more aggressive learning. SARSA, being an on-policy method, was more conservative but resulted in smoother learning progression. The Q-Learning with epsilon scheduling demonstrated improved stability towards the end, as exploration decreased gradually, leading to more optimal policy exploitation.

The final rewards converged for all agents, with Q-Learning and epsilon scheduling achieving slightly better average rewards compared to SARSA. However, the difference was not substantial, which suggests that both algorithms were able to successfully learn the optimal policy for the Taxi-v3 environment.

Conclusion

The Q-Learning algorithm converged faster but exhibited higher variance initially due to aggressive updates. SARSA provided smoother learning but at the cost of slower convergence. The epsilon scheduling for Q-Learning helped balance exploration and exploitation, leading to stable learning. Overall, each algorithm demonstrated strengths, with Q-Learning being preferable for faster learning and SARSA for safer, more reliable learning progression.

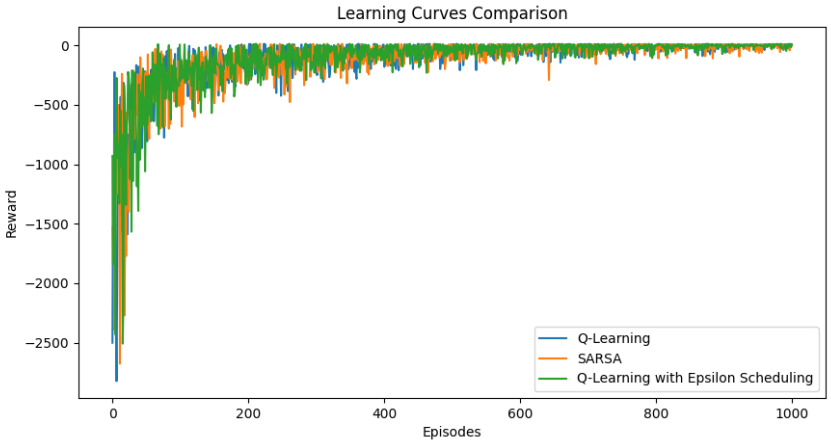

Figure 1: Learning Curves Comparison for Q-Learning, SARSA, and Q-Learning with Epsilon Scheduling