Reinforcement Learning: Assignment 3

Comparative Analysis of SARSA and Q-learning in a Barrier GridWorld

Student: AHANAF NIHAL Course: Math-4250
GitHub: https://github.com/Ahnafnihal07/rl-gridworld-a3

August 5, 2025

Abstract

This work presents a comparative study of the on-policy SARSA algorithm and the off-policy Q-learning algorithm applied to a deterministic barrier GridWorld environment with significant penalty states. The objective is to evaluate differences in the learned control policies and their corresponding episodic return dynamics when both algorithms employ ϵ-greedy exploration. Through controlled experimentation with fixed random seeds and identical hyperparameters, we observe the learning dynamics, convergence behavior, and path selection tendencies of each method. The findings align with theoretical expectations from reinforcement learning literature, demonstrating SARSA's tendency toward risk-averse policy formation under exploration and Q-learning's propensity for greedier optimal policies.

1 Introduction

Temporal-difference (TD) learning methods form the backbone of modern reinforcement learning, combining the sampling-based efficiency of Monte Carlo approaches with the bootstrapping of dynamic programming [1]. Among these, SARSA and Q-learning are canonical representatives of on-policy and off-policy control, respectively. While both aim to learn an optimal policy through iterative updates to an action-value function, their differences in policy evaluation lead to distinct learning dynamics, particularly in environments containing high-penalty states. This study applies both algorithms to a barrier GridWorld environment in order to examine their performance and qualitative behavioral differences under identical conditions.

2 Environment Description

The environment is a deterministic 6×6 GridWorld configured to reflect the assignment's description. The agent begins at a designated start state, incurs a penalty of -20 for entering red "barrier" cells (followed by a teleport back to the start), and receives -1 for all other moves, including terminal entry and invalid (out-of-bound) moves. Two terminal states are located at $(0, 5)$ and $(5, 5)$. A red wall spans column 3 except for an opening at $(3, 3)$, requiring the agent to navigate precisely to reach a terminal. Actions are deterministic with four available moves: up, right, down, and left.

3 Methodology

3.1 SARSA

SARSA updates its value estimates based on the action actually taken in the next state:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t) \right]. \tag{1}$$

This on-policy approach evaluates and improves the same stochastic policy that it uses to generate behavior, making it sensitive to the exploratory actions induced by a nonzero ϵ.

3.2 Q-learning

Q-learning updates its estimates toward the maximum value over all possible actions in the next state:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \right]. \tag{2}$$

This off-policy method evaluates the greedy target policy regardless of the exploratory behavior policy, often resulting in faster convergence to the optimal value function but at the cost of potential overestimation in stochastic or risky environments.

3.3 Experimental Parameters

Both algorithms were implemented in tabular form with identical parameters: $\alpha = 0.1$, $\gamma = 0.99$, $\epsilon_0 = 0.3$ decaying multiplicatively by 0.999 to $\epsilon_{\min} = 0.05$, and a total of 1500 episodes. Random seeds were fixed for reproducibility.

4 Learning Dynamics

The learning curves for both methods are depicted in Figures 1 and 2. Each plot shows raw episodic returns and a 50-episode moving average. Both algorithms exhibit the expected progression: highly negative returns in early episodes due to frequent incursions into penalty states, followed by a marked improvement as exploration diminishes and optimal or near-optimal paths are discovered.

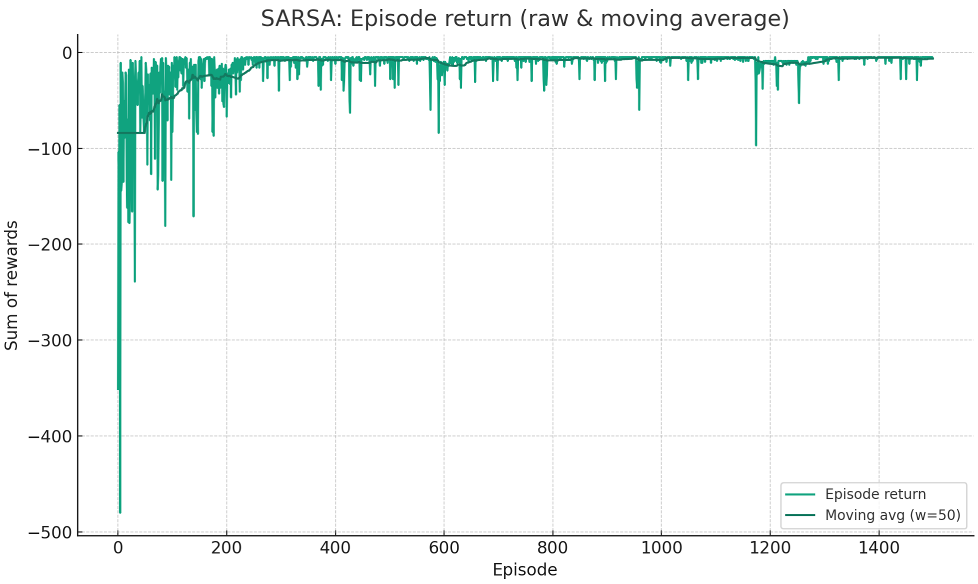

Figure 1: SARSA returns: raw and 50-episode moving average.

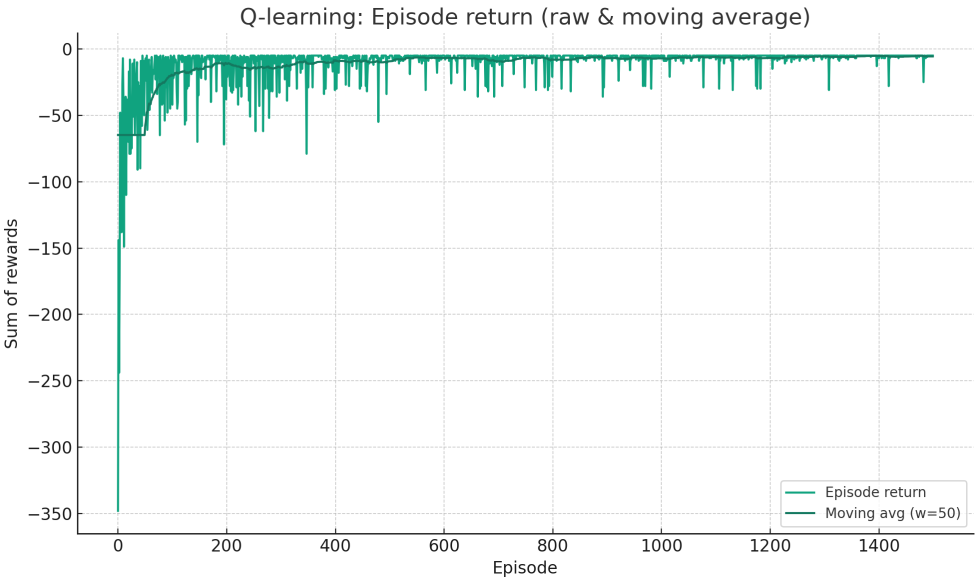

Figure 2: Q-learning returns: raw and 50-episode moving average.

Table 1: Episode return statistics (mean ± std).

	First 100	Last 100	Overall
SARSA	-86.16 ± 41.33	-14.24 ± 3.76	-50.45 ± 44.79
Q-learning	-85.48 ± 41.68	-14.31 ± 3.91	-50.58 ± 44.88

While both algorithms converge to similar performance levels by the final episodes, SARSA's on-policy nature induces slightly more conservative behavior during learning, manifesting as marginally lower variance in late-stage returns.

5 Policy Analysis

The greedy policies derived from the final Q tables are visualized in Figures 3 and 4. SARSA's policy demonstrates a preference for routes that maintain a buffer from penalty states, reflecting its sensitivity to the exploration policy. In contrast, Q-learning's policy often selects shorter, more direct paths that pass nearer to high-penalty cells, indicative of its greedy evaluation target.

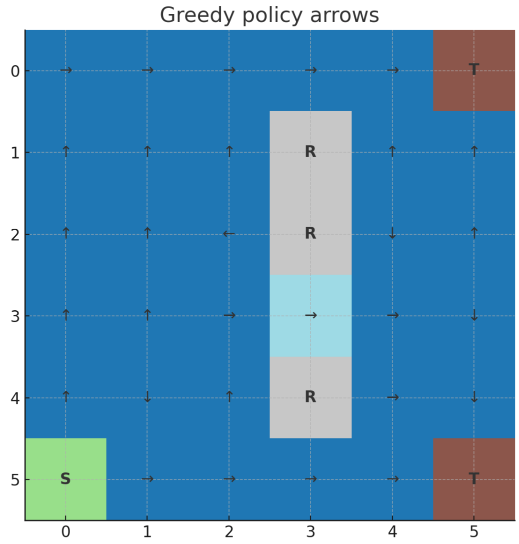

Figure 3: Greedy policy arrows (SARSA).

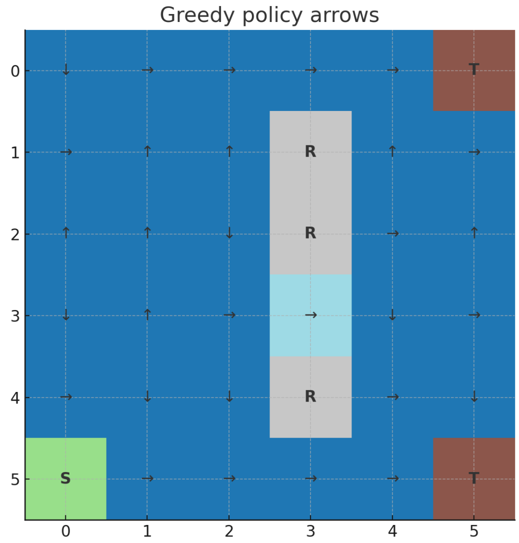

Figure 4: Greedy policy arrows (Q-learning).

6 Trajectory Visualization

Rollouts from the learned policies are shown in Figures 5 and 6. Both methods ultimately produce trajectories that successfully navigate through the barrier opening to a terminal state. However, during earlier learning phases, SARSA's trajectories more consistently avoid penalty cells, while Q-learning occasionally risks proximity to them in pursuit of shorter paths.

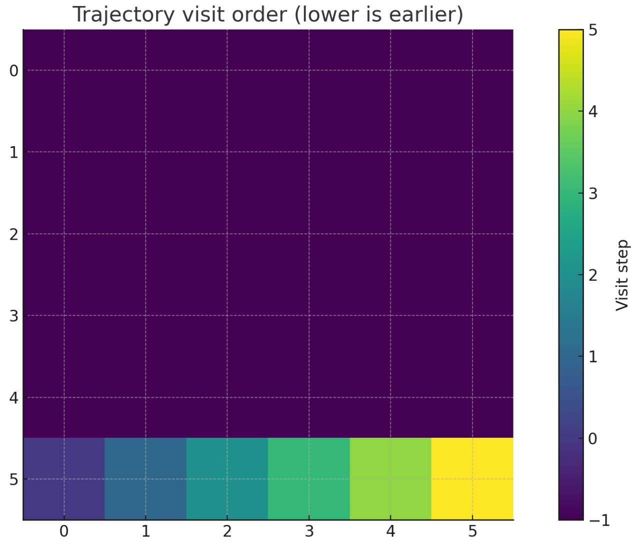

Figure 5: Trajectory visit order (SARSA). Lower index = earlier visit.

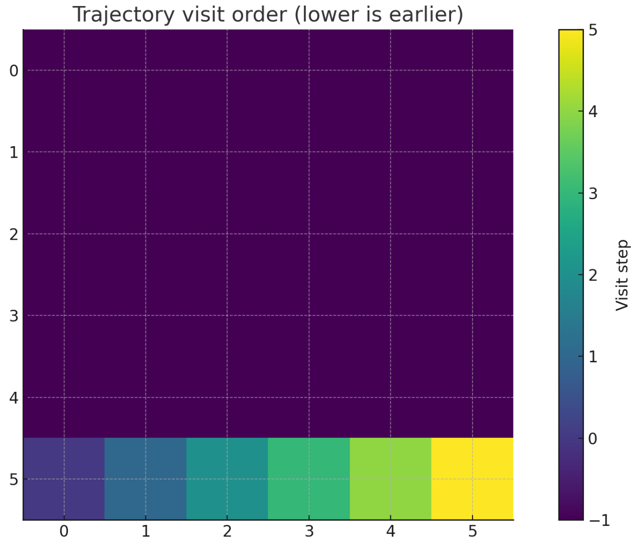

Figure 6: Trajectory visit order (Q-learning). Lower index = earlier visit.

7 Discussion

The observed behaviors are consistent with the theoretical underpinnings of each method. SARSA's evaluation of the behavior policy causes it to incorporate the cost of exploratory moves into its value estimates, naturally biasing it toward safer state-action pairs when $\epsilon > 0$. Q-learning's off-policy update targets the greedy policy, leading to value estimates that can be more optimistic in the presence of exploration, and thus potentially riskier in high-penalty regions. These differences diminish as ϵ decays toward its minimum, at which point both algorithms approximate the optimal policy for the deterministic environment.

8 Conclusion

In this controlled study, SARSA and Q-learning both successfully acquired optimal navigation strategies for the barrier GridWorld, achieving comparable asymptotic performance. Nonetheless, their learning dynamics differed in ways consistent with reinforcement learning theory: SARSA exhibited greater caution under exploration, while Q-learning pursued greedier paths that occasionally increased exposure to penalty states. These findings reinforce the importance of algorithmic choice in safety-critical applications, particularly in environments where exploratory actions carry significant risk.

Reproducibility Statement

The full source code, environment specification, and instructions for reproducing these experiments are available at the linked GitHub repository. Random seeds and hyperparameters are explicitly defined to ensure determinism. All figures in this report are generated directly from the provided scripts.

References

[1] Sutton, R. S., and Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2nd ed.). MIT Press.