Reinforcement Learning - Assignment Report

Abderrahim Namouh

CentraleSupélec
Assignment Notebook

Abstract. This project explores the application of reinforcement learning (RL) techniques within a simulated "Text Flappy Bird" environment, a text-based version of the popular game where the objective is to navigate a bird through a series of obstacles by making timely flaps. Our study focuses on the implementation and comparison of two distinct RL agents: a Monte Carlo Agent and a SARSA (State-Action-Reward-State-Action) Agent. Each agent employs a different approach to learn and optimize its policy for maximizing the score and survival time within the game.

Keywords: Reinforcement Learning · Flappy Bird · Monte Carlo · Sarsa.

1 Introduction

This study explores two reinforcement learning agents, Monte Carlo and SARSA, within a "Text Flappy Bird" environment. By comparing these agents, we aim to understand their learning dynamics, adaptability, and the impact of parameter tuning on their performance. This investigation provides insights into the strengths and weaknesses of each approach, guiding the development of more efficient RL strategies.

2 Agents

The Monte Carlo agent operates on the principle of learning from complete episodes before making policy updates. This method aggregates the rewards received throughout an episode and updates the value functions at the end, making it particularly suited for environments where the outcome is only clear after a sequence of actions. It relies heavily on the law of large numbers, expecting that, over time, the average return for each state-action pair will converge to its expected value. The agent's performance is less sensitive to immediate variations in the game environment, making it robust but slower to adapt to new strategies or unexpected changes.

The SARSA agent, on the other hand, employs a more dynamic approach by updating its policy after each action based on the state-action-reward-state-action sequence. This method allows for incremental learning, making the agent more responsive to the environment's immediate feedback. It learns a policy that

closely approximates the optimal policy by continuously adjusting based on the outcomes of its actions. However, this sensitivity makes it more prone to fluctuations in performance, especially in the early stages of learning or when exploring new strategies. The SARSA agent's approach is beneficial for environments requiring quick adaptation to achieve optimal performance.

3 Results

The graph shows a moving average of rewards from training over 10,000 episodes:

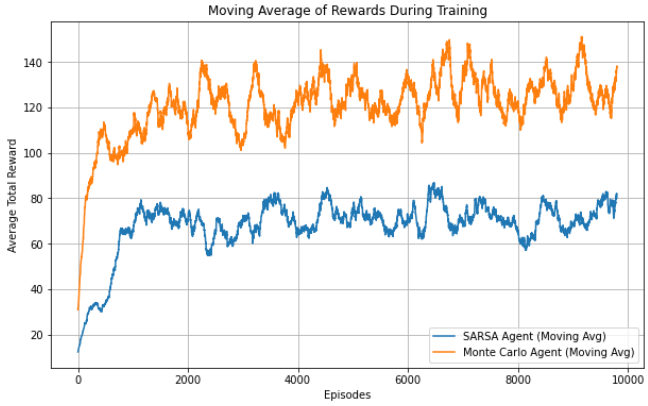

Fig. 1. Comparative Performance of SARSA and Monte Carlo Agents Over 10,000 Training Episodes

The graph reflects the intrinsic characteristics of the Monte Carlo and SARSA reinforcement learning agents. Monte Carlo's higher rewards suggest it benefits from its episode-based learning, possibly finding better policies through complete sequences of play. SARSA's lower, steadier curve aligns with its step-by-step update approach, possibly indicating a safer but less reward-optimized policy due to its on-policy, incremental learning nature. This suggests that Monte Carlo might be better at exploiting the game's structure, while SARSA maintains a more greedy approach.

3.1 State Value functions

Now we're going to explore the differences in the learned State Value functions of the two agents. The 3D plots illustrate the state-value functions learned by the SARSA and Monte Carlo agents. The SARSA agent's values show distinct peaks, suggesting a more differentiated strategy, whereas the Monte Carlo agent's plot indicates a smoother gradient, reflecting a more generalized approach to state valuation.

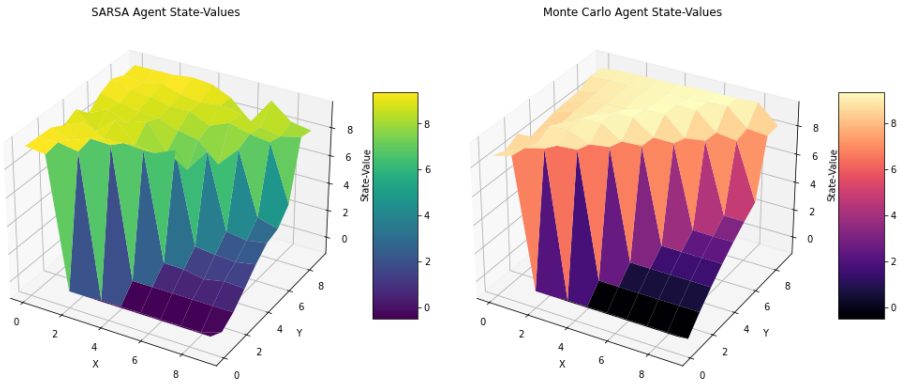

Fig. 2. Learned State-Value (Best Action) Functions by SARSA and Monte Carlo Agents

3.2 Parameters Finetuning

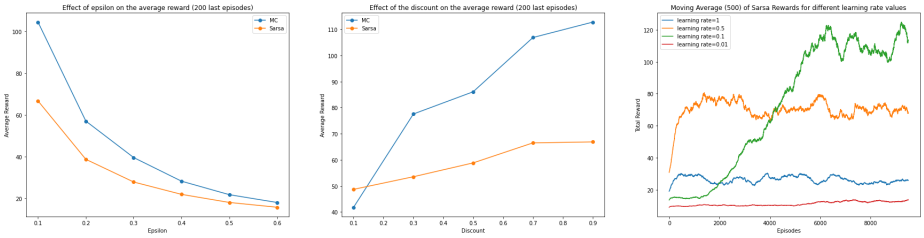

Fig. 3. Finetuning results for Epsilon, Discount and Learning Rate parameters respectively

The initial graphs detail the average rewards for the last 200 episodes, modulated by various epsilon and discount rates for both agents. A lower epsilon favors both agents similarly, reducing exploration. The Monte Carlo agent, however, displays a greater response to higher discount rates, indicating an increased benefit from future rewards. The final graph highlights the learning rate's impact on the SARSA agent, showing suboptimal performance at extreme values, whereas a learning rate of 0.1 sustains better long-term results.

4 Conclusion

Despite its simplicity, the Monte Carlo agent demonstrated robustness and consistently better performance compared to SARSA, which was sensitive to parameter tuning and showed volatility in rewards. Both agents' performances are

inherently limited by the state definition in the "Text Flappy Bird" environment, which affects their generalization capabilities. Future work should aim at refining these state definitions to enhance learning efficiency and agent adaptability in complex settings.

References

1. Richard S. Sutton, Andrew G. Barto: Reinforcement Learning: An Introduction. MIT Press (2018)
2. Christodoulidis Stergios: Source Environment: Text Flappy Bird Gym Environment