COMP 138 RL: Programming Assignment 3

Hannah Doherty

November 9, 2022

1 Example 6.6: Cliff Walking

This gridworld example compares Sarsa and Q-learning, highlighting the difference between on-policy (Sarsa) and off-policy (Q-learning) methods. Consider the gridworld shown to the right. This is a standard undiscounted, episodic task, with start and goal states, and the usual actions causing movement up, down, right, and left. Reward is -1 on all transitions except those into the region marked "The Cliff." Stepping into this region incurs a reward of -100 and sends the agent instantly back to the start. The graph to the right shows the performance of the Sarsa and Q-learning methods with "ϵ-greedy action selection, ϵ = 0.1. After an initial transient, Q-learning learns values for the optimal policy, that which travels right along the edge of the cliff. Unfortunately, this results in its occasionally falling off the cliff because of the "ϵ-greedy action selection. Sarsa, on the other hand, takes the action selection into account and learns the longer but safer path through the upper part of the grid. Although Q-learning actually learns the values of the optimal policy, its online performance is worse than that of Sarsa, which learns the roundabout policy. Of course, if ϵ were gradually reduced, then both methods would asymptotically converge to the optimal policy. [1]

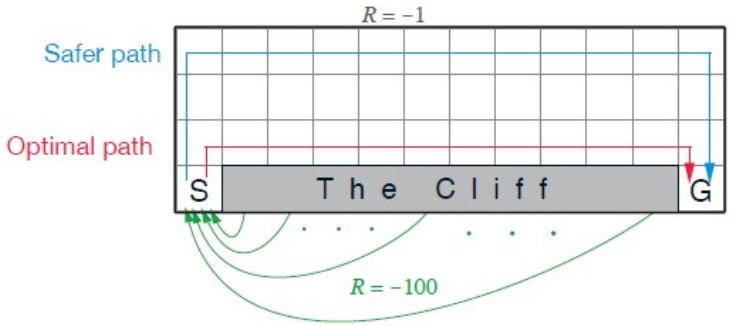

Figure 1: Cliff-walking Problem

2 Problem Statement

For this assignment, I will do experiments using temporal difference methods on the Cliff-walking example from Chapter 6 of Sutton and Barto.
The main requirements are: 1) I implement and test at least 2 algorithms (they could both be TD or you could compare a TD vs an MC methods on a domain); and 2) I answer two separate questions with experiments: the primary question is to replicate the results in the example while the secondary one is a question I formulate myself.

3 Problem Solution

3.1 Analysis

To solve the Cliff-walking problem, I implemented two Temporal Difference reinforcement learning techniques - Q-Learning and SARSA. The learning rate is 0.5 to avoid a fast convergence. The discount factor is 0.9 to show the importance of future rewards. And finally epsilon is 01 to should that the agent should take random actions 10% of the time. The plot refers to "batches of 10" which means that 10 consecutive were grouped together and then normalization is applied to the batch. It was found that normalization gave for better visualization.
SARSA is an on-policy method where the q-value depends on the action performed by the current policy. Q-Learning decides what action to take based on looking at the state value alone. Q-Learning learns the optimal policy by moving along the cliff but the random exploration causes it to fall off creating higher penalties.

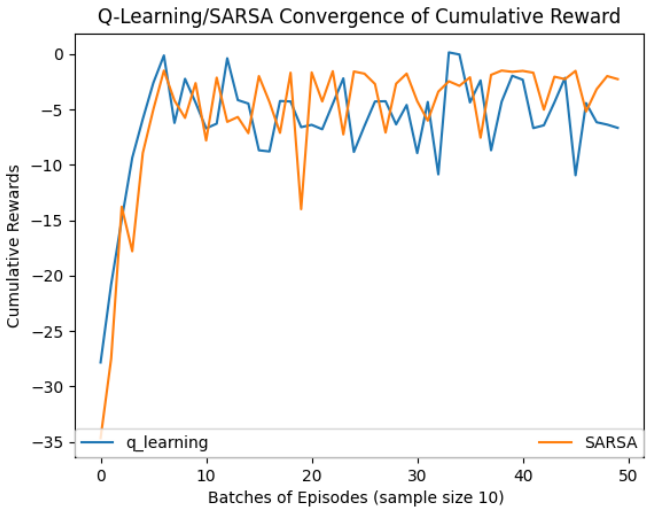

Figure 2: Cliff-walking Problem: Plot of Q-Learning and SARSA showing cumulative reward for each batch of episode. ϵ=0.1

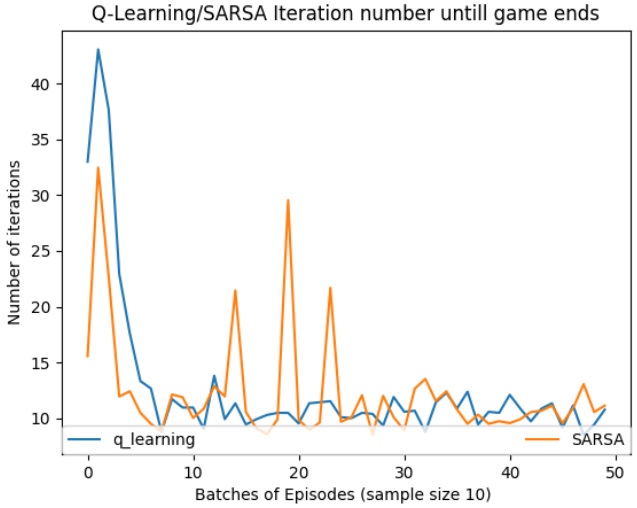

Figure 3: Cliff-walking Problem: Plots of Q-Learning and SARSA showing number of iterations for each batch of episode. ϵ=0.1

The plot below shows the data without the normalization applied. One of the fundamental differences shown is the cumulative rewards are much lower

initially.Something that should be mentioned that these are on two different runs so they aren't on the same 500 episodes.

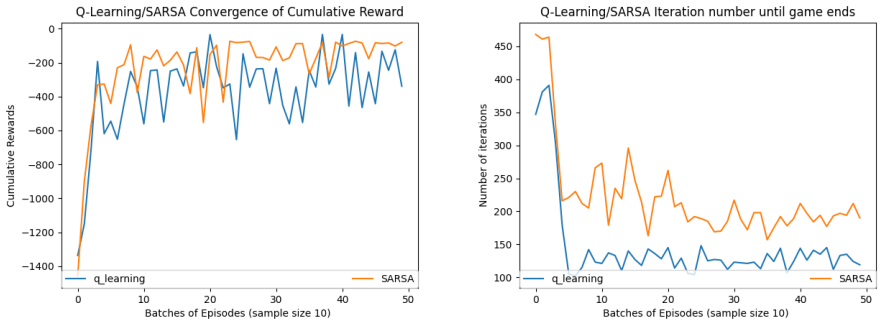

Figure 4: Plots of Q-Learning and SARSA showing cumulative reward and number of iterations for each batch of episode. ϵ=0.1

4 Conclusion

In conclusion, you can see in the plots from the Analysis section that in the beginning, the agent keeps falling off the cliff so the reward is extremely low. But after about 50-100 episodes the agent starts to learn about the cliff's low reward and we see an increase in average reward. For Q-learning, the agent is closer to the cliff but has a shorter path versus the SARSA agent which takes a longer path and safer route. But at the end of the episodes and the training, we see that SARSA benefits from a longer safer path with higher rewards on average compared to Q-learning.

5 Experiment

5.1 Hypothesis

I hypothesize that if I decrease the epsilon value then I'll show a faster convergence. I believe this will work because once our policy has learned how to be efficient then the amount of exploration will decrease.

5.2 Analysis

To solve the Cliff-walking problem, I implemented two Temporal Difference reinforcement learning techniques - Q-Learning and SARSA. The learning rate is 0.5 to avoid a fast convergence. The discount factor is 0.9 to show the importance of future rewards. And finally epsilon is 01 to should that the agent should take random actions 10% of the time. The plot refers to "batches of 10" which

means that 10 consecutive were grouped together and then normalization is applied to the batch. It was found that normalization gave for better visualization.

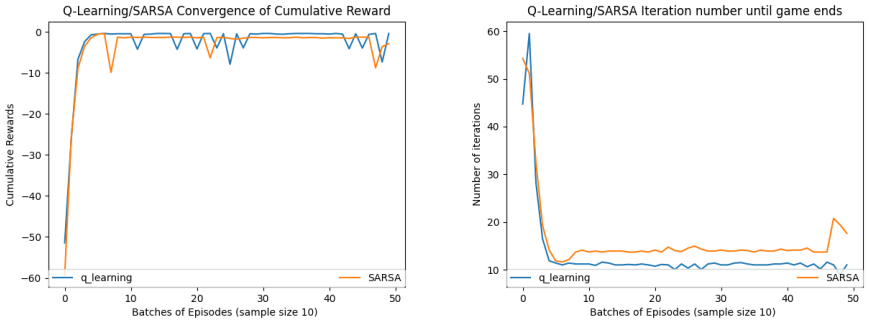

Figure 5: Plots of Q-Learning and SARSA showing cumulative reward and number of iterations for each batch of episode. ϵ=0.01

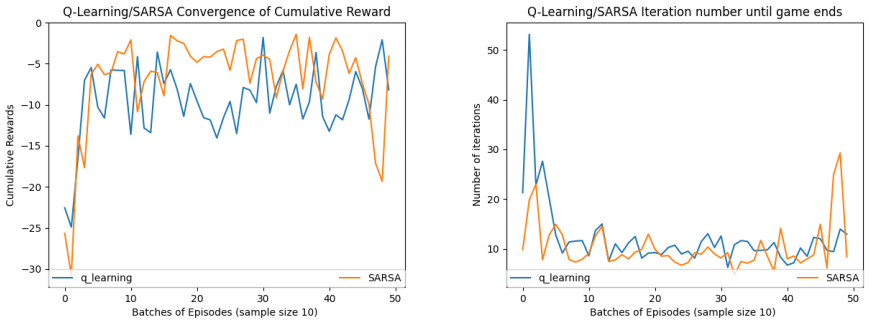

Figure 6: Plots of Q-Learning and SARSA showing cumulative reward and number of iterations for each batch of episode. ϵ=0.2

5.3 Conclusion

In conclusion, I have shown through analysis that my hypothesis is correct. As epsilon decreases, we see a faster convergence in Figure 4's left plot. One thing to mention is that the Cumulative Reward is initially lower for the lower epsilon versus the higher epsilon in Figure 5. This is due to happenstance and random chance that the initial starting runs were much lower for the lower epsilon than the higher epsilon.

6 Extra Credit

6.1 Exercise 8.3

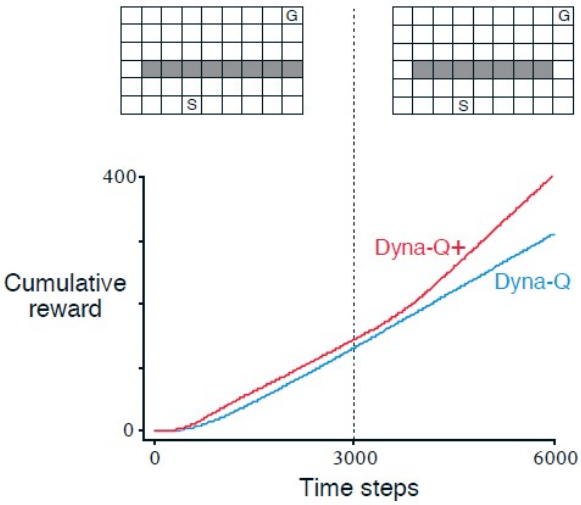

Figure 7: In textbook, this is Figure 8.5

Problem Statement: Careful inspection of Figure 8.5 reveals that the difference between Dyna-Q+ and Dyna-Q narrowed slightly over the first part of the experiment. What is the reason for this?

Solution: I think that the Dyna-Q+ and Dyna-Q narrowed slightly over the first part of the experiment because Dyna-Q+ was converging a little fast which consequently caused costs to incur. The cost makes Dyna-Q+ explore more even if it's already optimal.

6.2 Exercise 8.6

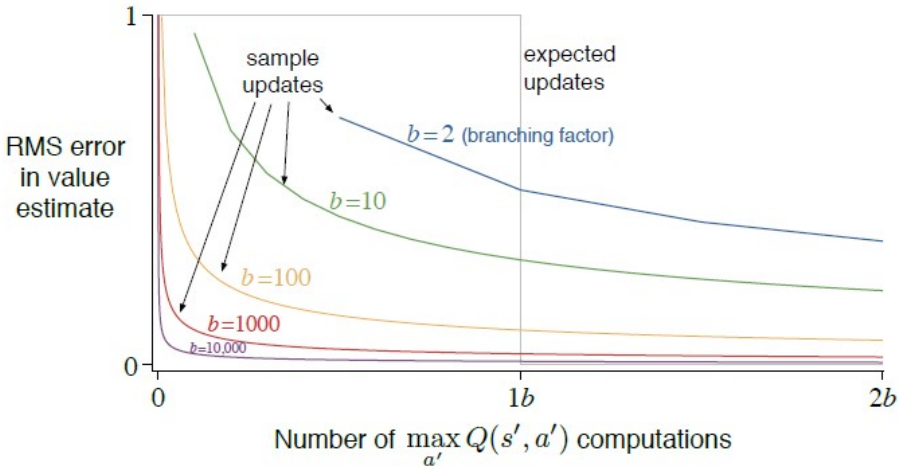

Figure 8.7: Comparison of efficiency of expected and sample updates.

Figure 8: In textbook, this is Figure 8.7

Problem Statement: The analysis above assumed that all of the b possible next states were equally likely to occur. Suppose instead that the distribution was highly skewed, that some of the b states were much more likely to occur than most. Would this strengthen or weaken the case for sample updates over expected updates? Support your answer.

Solution: The sample updates are strengthened because they can ignore low frequency states and stress the high frequency ones.

References

[1] A. Sutton, R. Barto. *Reinforcement Learning*. The MIT Press, 2020.