COMP 579 Reinforcement learning
Assignment 2

Tung Vu (261143697)
Vraj Patel (261022581)

Dr. Doina Precup
McGill School Of Computer Science
McGill University

1 Exercise 1: Tabular RL

In this problem, we compared the performance of SARSA and expected SARSA on the Frozen Lake domain from the Gym environment suite. We have used the default environment to train the agent.

1.1 SARSA

SARSA(State-Action-Reward-State-Action) is on-policy learning algorithm, which means that the policy used to select actions during learning is the same policy used to select actions during evaluation.
The leaning of the action-value pair for SARSA algorithm follows following equation:

$$Q^{new}(s_t, a_t) = Q(s_t, a_t) + \alpha[R_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \tag{1}$$

1.2 Expected SARSA

Expected SARSA(State-Action-Reward-State-Action) is an off-policy algorithm, which means that the policy used during learning is different from the policy used during evaluation.
The update for the action value pairs for the Expected SARSA algorithm follows this equation:

$$Q^{new}(s_t, a_t) = Q(s_t, a_t) + \alpha[R_t + \gamma \sum_a \pi(a|s_{t+1}) Q(s_{t+1}, a) - Q(s_t, a_t)] \tag{2}$$

1.3 Exploration

For the exploration of the states we used Softmax function to calculate the probability of different actions.

$$\text{softmax}(\mathbf{x})_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \tag{3}$$

$$x_i = \frac{Q(s_t, a_t)}{T} \tag{4}$$

For this experiment, the temperature parameter determines the randomness of the agent's action selection, higher values of temperature provide more exploration to the agent meanwhile, whereas lower temperature exploits the agent's behavior. Since our state space is very low(16), we decided to use lower values of the temperatures so that we can get high average rewards.

1.4 Results

Final training performance for different parameters

For this experiment, we have chosen three different values of temperatures and learning rates with a fixed discounted factor $\gamma = 0.95$ given in Table 1.

Learning Rate	Temperature
0.03	0.01
0.05	0.02
0.08	0.03

Table 1. Values of the different parameters

The performance of the final training averaged over the last 10 episodes and the 10 runs for SARSA and Expected SARSA are given in Figure 1.

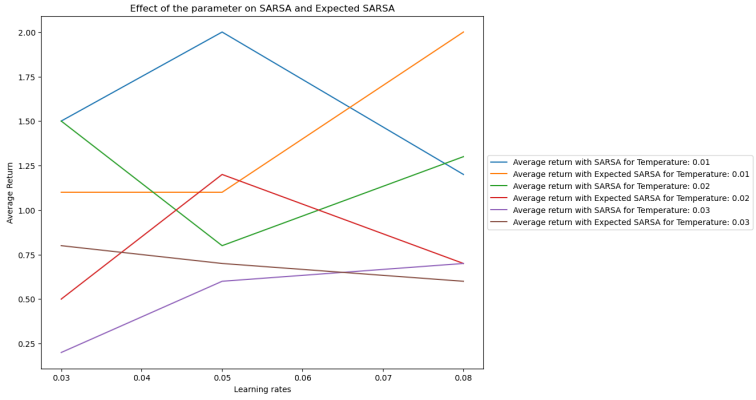

Fig. 1. Average rewards for different values of parameters for training

Final testing performance for different parameters

For the testing, we have used the same parameters given in Table 1.
The performance of the final testing performance, during the final testing episode, averaged over the 10 runs is given in Figure 2.

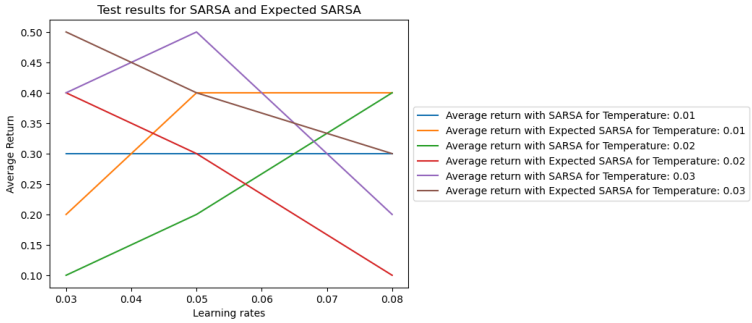

Fig. 2. Average rewards for different values of parameters for testing

As from the above Figures 1 and 2, we can say that overall, expected SARSA gives a higher average for the rewards compared to SARSA.

Leaning curve on best parameters

After doing experiments, we have picked the best parameters given in Table 2

Algorithm	SARSA	Expected SARSA
T	0.03	0.03
α	0.05	0.03

Table 2. Best parameters for each algorithm

Learning curve for SARSA

In Figure 3, we can see the learning of the agent and increasing the average rewards for SARSA with the shaded area as standard deviation from the mean. Here, the variance is very high even though, the average reward increases as the number of segments increases.

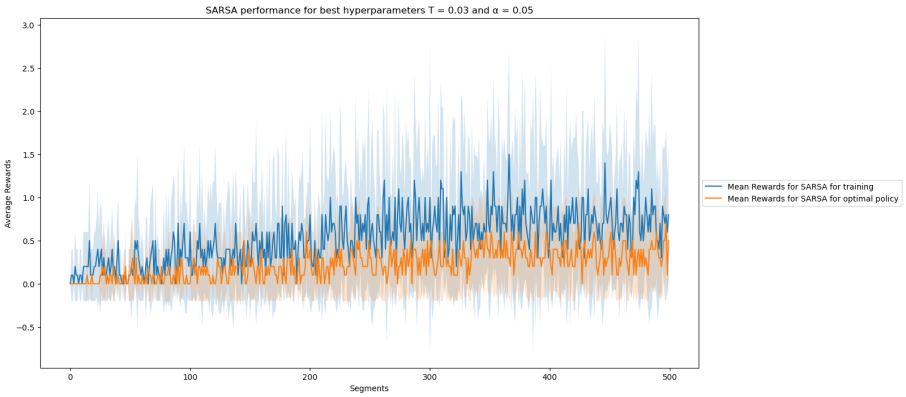

Fig. 3. Learning curve for training and testing for SARSA

Learning curve for Expected SARSA

Figure 4 shows the learning curve of the Expected SARSA algorithm. Mean rewards for expected SARSA converge very faster compared to SARSA, at the same time, the standard deviation is low compared to SARSA.

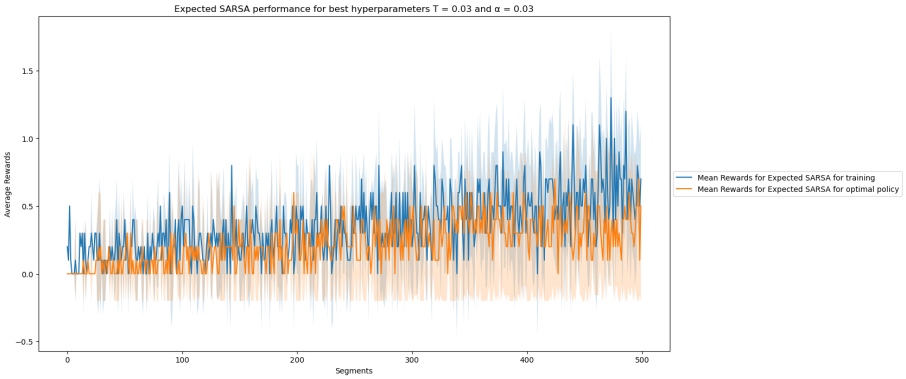

Fig. 4. Learning curve for training and testing for Expected SARSA

Conclusion

Based on the experimental results which can be seen in Figure 5, we have found that Expected SARSA outperformed SARSA in terms of both speed of convergence and consistency of results. Expected SARSA updates the Q-table more efficiently and effectively, allowing it to learn the optimal action faster than SARSA. Additionally, the lower variance in rewards observed with Expected SARSA indicates that it provides more consistent results, which is an important factor in many real-world applications.

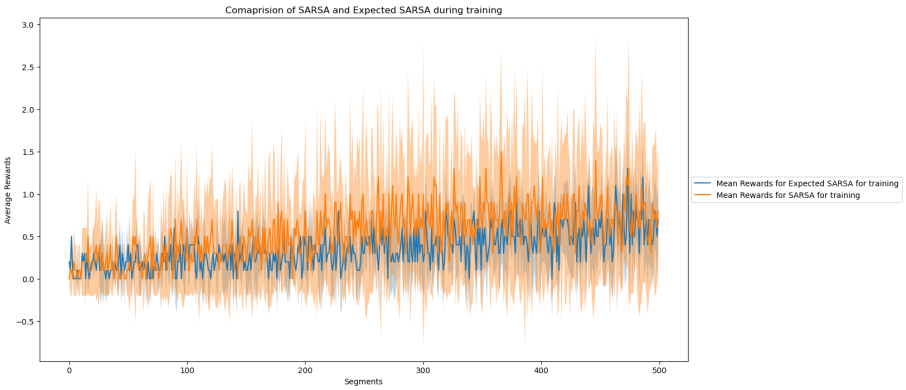

Fig. 5. Learning curve on training performance

However, it is important to note that the superiority of Expected SARSA over SARSA may depend on the specific environment and task being performed. Therefore, it is recommended to test both algorithms on a variety of environments and tasks to determine which one is best suited for a particular application. In conclusion, the results of this study suggest that Expected SARSA is a promising reinforcement learning algorithm that may be more effective and efficient than SARSA in certain scenarios.

2 Exercise 2: Function approximation in RL

In this problem, we have implemented and compared the Q-learning and Actor-critic with linear function approximation on the cart-pole environment. Regarding state discretization, our approach was to discretize the state into 10 bins based on the range of distances mentioned in the cart-pole gym documentation. Since the range of the cart velocity and the pole angular velocity are too broad to discretize, we had to sample with a pre-defined range to see the max and min values and set appropriate ranges for these two observations. We evaluated these algorithms based on the average and standard error of the learning curves over the 10 runs for each learning rate. The standard error evaluation can be viewed in the shaded area of the graphs. It is important to note that, we implemented the backpropagation and linear function approximator from scratch without using any library like Tensorflow or Pytorch.

2.1 Q-learning

In this experiment, we have used the learning rates $\alpha \in [0.0625, 0.125, 0.25]$, exploration rates $\epsilon \in [0.008, 0.05, 0.1]$ and discount factor $\gamma = 0.9$. After conducting 10 independent runs, each of 1000 episodes, the results are shown in Figures 6, 7, and 8.

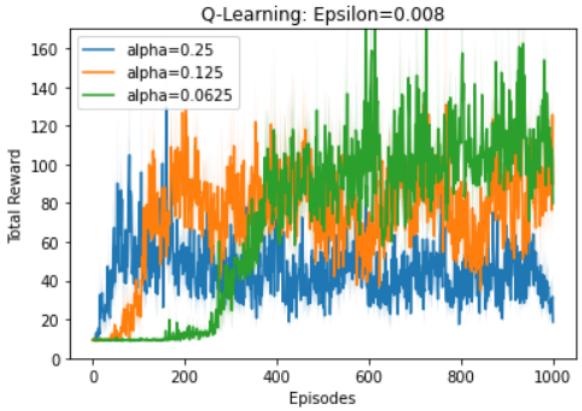

Fig. 6. Q-learning with $\epsilon = 0.008$

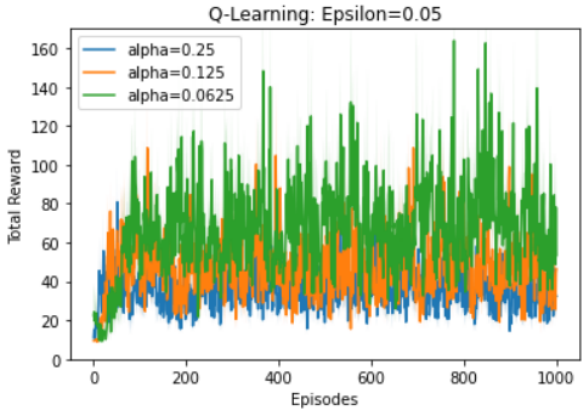

Fig. 7. Q-learning with $\epsilon = 0.05$

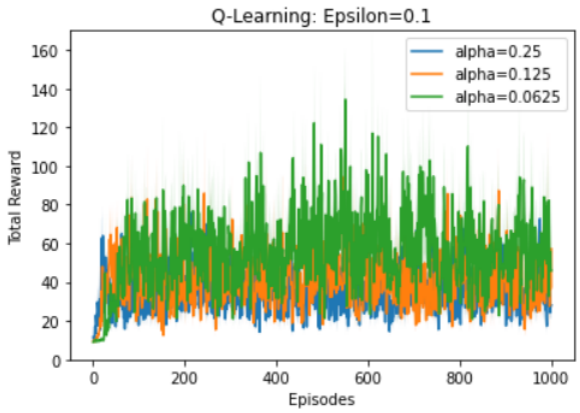

Fig. 8. Q-learning with $\epsilon = 0.1$

As can be seen in the above graphs, the performance of the Q-learning algorithm is better with a small value of $\epsilon = 0.008$, and a small value of learning rate $\alpha = 0.0625$ (green curve) compared to greater values $\epsilon \in [0.05, 0.1]$, and $\alpha \in [0.25, 0.125]$. It can be explained that, with $\epsilon = 0.008$ and learning rate $\alpha = 0.0625$, the learning curve can reach the convergent point and the total rewards (about 100) is greater compared to larger ϵ and learning rate α. Therefore, we can conclude that, in the cart-pole problem, when applying the Q-learning algorithm, we are able to achieve better performance with less exploration ($\epsilon = 0.008$) and a small learning rate ($\alpha = 0.0625$).

2.2 Actor-critic

In this experiment, we also have used the learning rate parameter $\alpha \in [0.25, 0.125, 0.0625]$, and the discount factor parameter $\gamma = 0.9$. It is important to note that in Actor-critic, we selected the action based on the probability obtained from the Soft-max so there is no need to use ϵ parameter for this experiment. After conducting 10 different runs, each of 1000 episodes, the results are shown in Figures: 9.

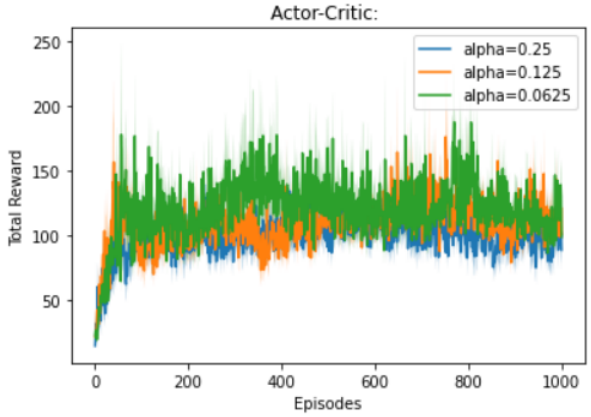

Fig. 9. Actor-critic

As can be seen in the above graphs, since the learning curve for $\alpha = 0.0625$ can reach the convergent point and the total rewards (about 120) is greater, the performance of the actor-critic algorithm is better with a small value of learning rate $\alpha = 0.0625$ (green curve) compared to greater values of α. Therefore, we

can conclude that, with our settings in the cart-pole problem, when applying the Actor-critic algorithm, we are able to achieve better performance with small learning rate ($\alpha = 0.0625$).

2.3 Best parameters choices and conclusion

Based on the result, 3 shows the best parameters for Q-learning and Actor-critic algorithms. The performance of Q-learning and Actor-critic with best parameters can be viewed in 10.

Algorithm	Q-Learning	Actor-critic
α	0.0625	0.0625
ϵ	0.008	N/A

Table 3. The best parameters selected for Q-learning and Actor-critic.

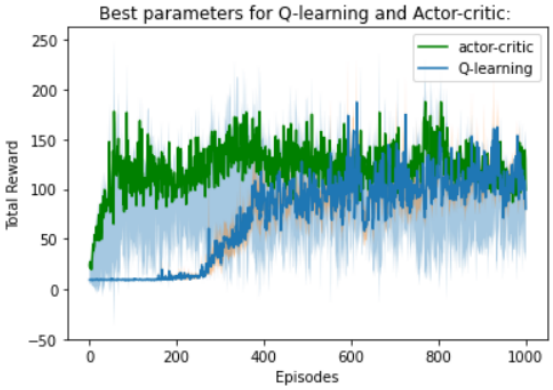

Fig. 10. Best performance for Actor-critic and Q-learning

In conclusion, the Actor-critic algorithm performs better than the Q-learning with the same learning rate. It can be explained that for the current settings in the cart-pole problem, the actor-critic's learning curve reaches the convergence point faster than that of Q-Learning. Moreover, the average return of Actor-critic (120.28) is greater than that of Q-learning (71.2).