

CISC 474 - Assignment 3

TASK III

Implementation

***The implementation for both SARSA and Q-Learning was the same in this task, except when the agent was called*

run_experiment_average()

This function is similar to the *run_experiment* function defined by the professor in the environment, where the function takes in the environment, agent, number of steps and epsilon and determines the mean rewards by looping over the number of steps. However, instead of returning just the *mean_reward*, this function appends the *mean_reward* at every step to a list called rewards. The function then returns this list.

Plotting

The number of steps is initialized to 100000, a list of *epsilons* is initialized to [0.1, 0.5, 1], and an empty dictionary called *epsilon_rewards* is initialized. The function loops over the *epsilons* list and initializes the environment and the agent. For SARSA, the *SarsaAgent* class is initialized, and for Q-Learning, the *QLearningAgent* class is initialized. Next, the list of average mean rewards returned by the *run_experiment_average()* function is assigned to its respective epsilon value in the dictionary named *epsilon_rewards*.

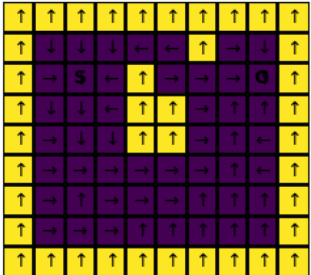
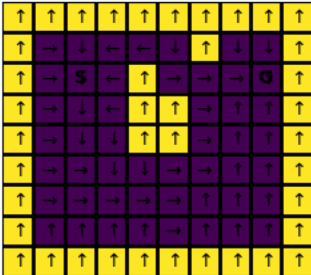
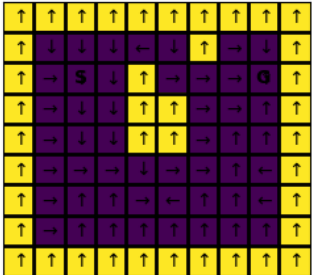
To visualize the policy, the *visualise_policy* function is called, which takes in the environment (*grid_S* or *grid_Q*) and agent (*agent_S* or *agent_Q*) of the algorithm.

Both algorithms are then plotted by looping over the elements in the rewards list in the *epsilon_rewards* dictionary and plotting the rewards received at each step. The x-axis is the *Number of Steps* and the y-axis is the *Average Reward*.

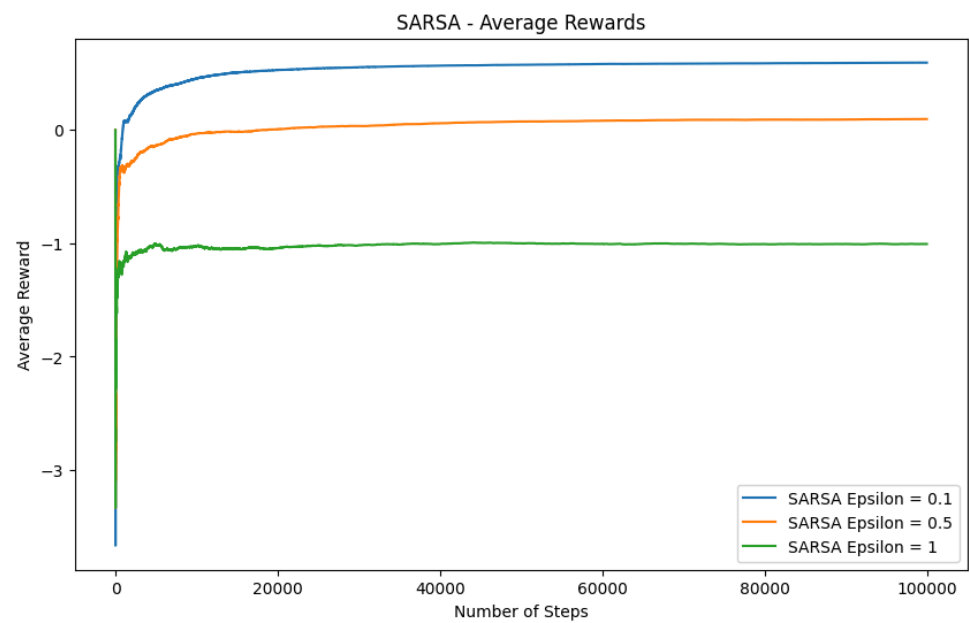
Algorithm Results

SARSA

Policy Visualization

$\epsilon = 0.1$	$\epsilon = 0.5$	$\epsilon = 1$
<p>Policy Visualization</p> 	<p>Policy Visualization</p> 	<p>Policy Visualization</p> 

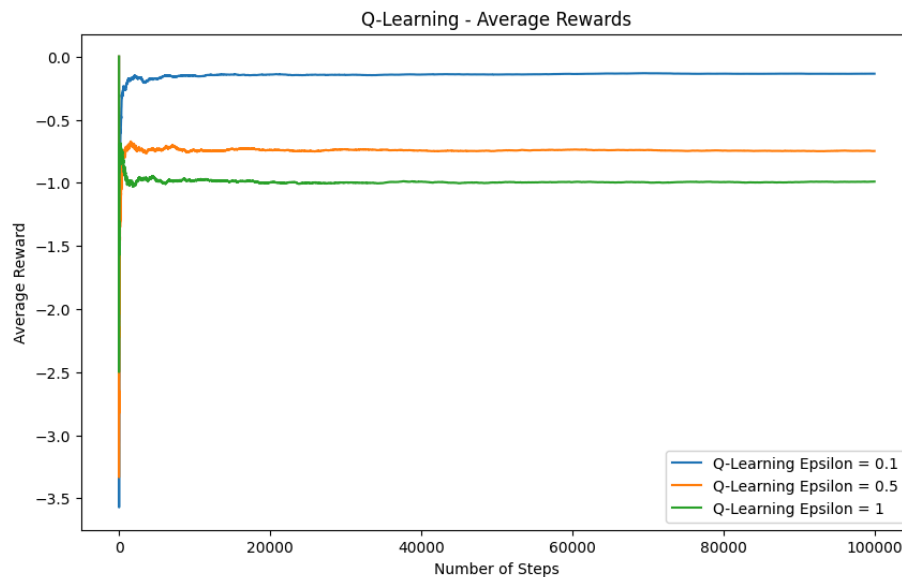
Average Rewards Graph



Q-Learning
Policy Visualization

$\epsilon = 0.1$	$\epsilon = 0.5$	$\epsilon = 1$
<p>Policy Visualization</p>	<p>Policy Visualization</p>	<p>Policy Visualization</p>

Average Rewards Graph



How do different values of epsilon affect the training in both algorithms?

****Note:** SARSA will always choose the safer path and Q-Learning will always choose the optimal path.

SARSA

Epsilon Value = 0.1

When the epsilon value equals 0.1, there is a lower chance of exploration. As the agent is exploiting more, it is able to improve its performance relatively quicker as it chooses the optimal actions which give a higher reward. In the graph where there is a slight drop in the value for the average rewards, this is an indication that the agent missed out on determining the most optimal action, which could have led to a higher reward. For example, in the initial steps, the average reward is around -0.2 and it drops to around -0.25. This indicates how it did not determine the most optimal action at a specific step. In the long term, the graph is relatively stable as there is less variability in the average rewards. This indicates that the agent chooses the most optimal actions to receive the highest rewards.

Epsilon Value = 0.5

When the epsilon value equals 0.5, the agent is exploring and exploiting. The agent continues to explore the environment while continuing to discover higher rewards which gives a better learning outcome. Initially, there may be a greater chance of variability as the agent is discovering various actions. In the long run, the average rewards show an upward trend as the agent continues to learn from exploration. In the graph, there is greater variability in steps before 12000. However, the graph does show more stability and an upward trend after this.

Epsilon Value = 1

When the epsilon value is equal to 1 (or higher), there is a greater chance of exploration. The agent will likely explore various actions for each state, leading to a lower reward average in the initial steps of training as the optimal actions are not exploited. Initially, the graph may have

multiple ups and downs, showing that the agent is exploring multiple actions that may have fewer rewards. This is visible in the graph where the value for the average rewards spikes up to 0 and then drops to -0.75, then -1.15 and then increases to around -0.9. The values of the reward are relatively low compared to other epsilons and there is greater variability in the initial states.

Q-Learning

Epsilon Value = 0.1

Like SARSA, when the epsilon value is set to 0.1, this indicates that the agent is exploiting more than exploring. In this case, the agent is able to improve its performance faster as it is choosing the best actions. Compared to SARSA, the curve for Q-Learning converges faster and shows less variability as the values are based on the maximum future rewards. Additionally, compared to other curves in the same graph, the curve for the epsilon value of 0.1 displays less variability than other curves.

Epsilon Value = 0.5

When the value of the epsilon is 0.5, the agent is both exploring and exploiting. Initially, the graph has more variability as the agent focuses on exploration, but the average number of rewards received increases over time. For example, in the initial steps, the average reward received is around -0.75 and after a few steps, it plateaus to around -0.85 and then gradually increases.

Epsilon Value = 1

When the value of the epsilon is 1, the agent is purely exploring by choosing actions randomly. In this case, it takes more time for the agent to learn the optimal policy. Initially, the graph shows a significant amount of variability. For example, the curve first peaks at 0.0, immediately drops right below -1.5, then increases to about -0.9, then drops right below -1.0, and then gradually increases. While this occurs, have either determined the optimal policy or have less variability.

Additionally, as the values for the epsilon are reduced to below 0.1, the curves converge faster and show less variability.

What is the optimal value for epsilon? (both algorithms)

The optimal value for both algorithms is 0.1, as it returns the highest average reward. The average reward for Q-Learning is lower than the average reward for SARSA because during the e-greedy policy selection, Q-Learning learns the optimal path and SARSA learns the safer path. The curve for Q-Learning does converge faster than the one for SARSA because the algorithm determines the actions that provide the maximum return. In riskier environments, Q-Learning will give better values for returns than SARSA as SARSA is more cautious and avoids paths that involve penalties.

Additionally, epsilon values under 0.1 provide an even better return for average rewards as the algorithm reduces exploration. This is visible in the graphs below:

