AE598 HW1: Model-Based and Model-Free RL

I. Introduction

Policy iteration and value iteration are both model-based RL algorithms. The model-based algorithms require explicit transition dynamics based on the Markov decision process, whereas the model-free algorithms do not rely on a priori transition dynamics.

II. Gridworld and Pendulum

In this assignment, we apply the aforementioned algorithms to two distinct environments modeled as Markov decision processes. The grid world is of dimension 5×5 and there are 4 possible actions, whereas the pendulum, being an inherently continuous-time problem, is discretized into a finite state and action domain.

III. Results

Policy and value iterations are straightforward as we use a fixed discount factor of $\gamma = 0.95$. SARSA and Q-Learning, however, require two other hyperparameters: α, the learning rate in the TD update equations, and ϵ, which balances exploration and exploitation when choosing an ϵ-greedy action. In my implementation for SARSA and Q-Learning, I randomly chose a few different values for ϵ and α:

$$\alpha = [0.3, 0.5, 0.8], \epsilon = [0.10.30.5] \tag{1}$$

Intuitively, the larger α is, the more aggressive the TD update is as it represents a learning rate. The larger ϵ is, the more exploratory the action is, as it represents a tradeoff between exploration and exploitation.

A. Gridworld

1) Policy Iteration: Fig. 1 are the results obtained from applying policy iteration to the grid world.

2) Value Iteration: Fig. 2 are the results obtained from applying value iteration to the grid world.

3) SARSA: Fig. 3 are the results obtained from applying SARSA to the grid world.

4) Q-Learning: Fig. 4 are the results obtained from applying Q-learning to the grid world.

5) Discussion: By observing the differences between all four algorithms applied to the grid world, we notice that the model-free methods return similar trends in the state-value function, whereas the model-based methods return a decreasing trend in the state-value function. This is likely due to the nature of the trade-off between exploration and exploitation.

Moreover, it is noticed that the mean value returned by the policy iteration follows a somewhat weird trend. There might be some convergence issues. It is interesting to note that

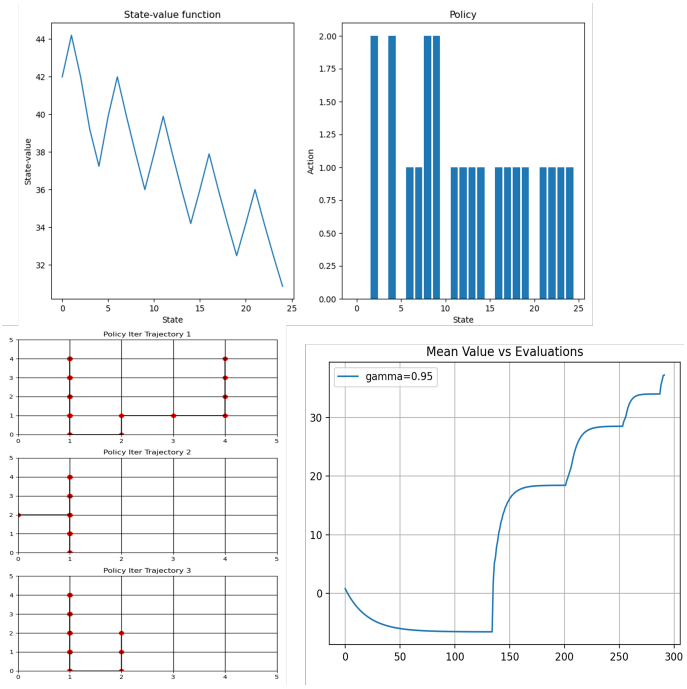

Fig. 1: Policy Iteration results (grid)

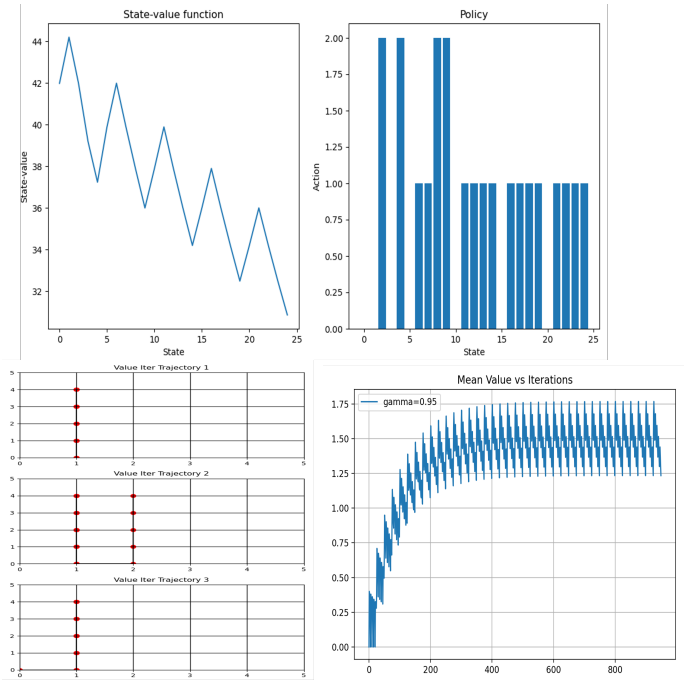

Fig. 2: Value Iteration results (grid)

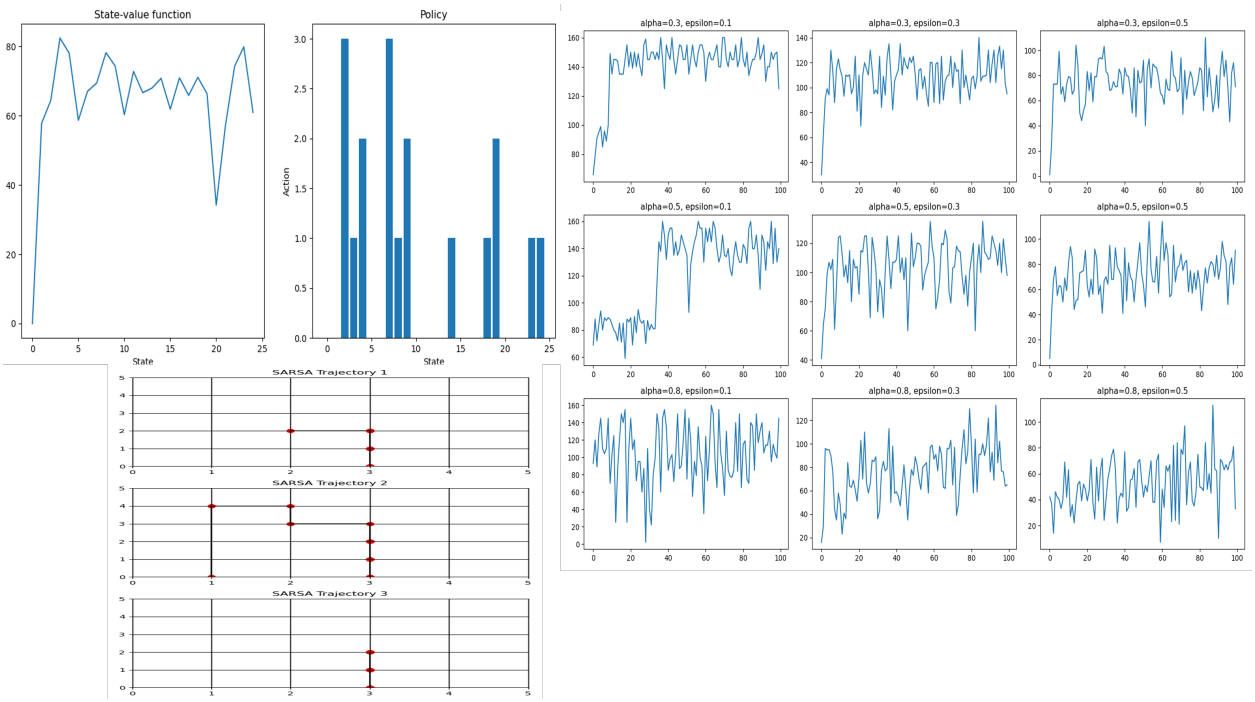

Fig. 3: SARSA results (grid)

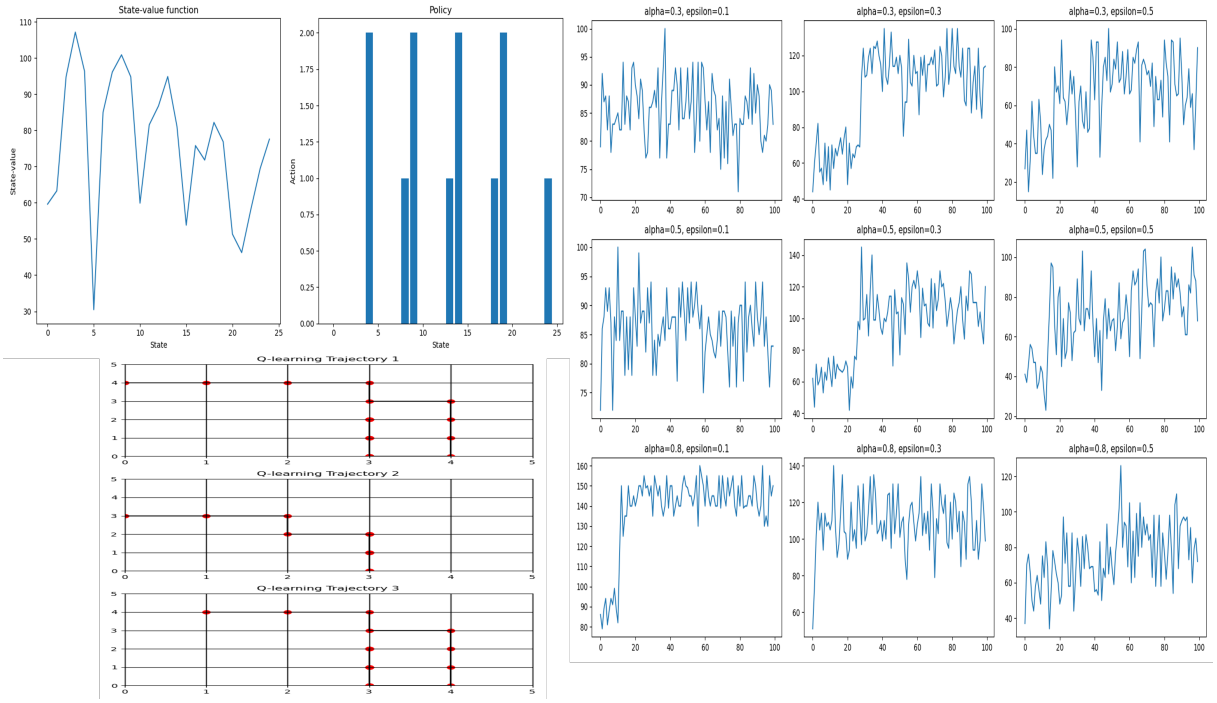

Fig. 4: Q-learning results (grid)

value iteration and policy iteration returned the same policy, as indicated in Fig. 1 and Fig. 2. The trajectory of each trained agent is plotted by converting the simulated trajectory into the $x-y$ coordinates of the grid world. For example, if a particular state in the trajectory is 7, then it corresponds to (2,1) in the $x-y$ coordinates (refer to more details in the code).

Another common pattern in the learning curves of both model-free methods is that there are a lot of fluctuations. I think it is related to the specific hyperparameters chosen.

B. Pendulum

1) SARSA: Fig. 5 are the results obtained from applying SARSA to the pendulum.

2) Q-Learning: Fig. 6 are the results obtained from applying Q-learning to the pendulum.

3) Discussion: By inspecting Fig. 6 and Fig. 5, we notice that the dimensionality of the problem is much higher. The state space is discretized into a grid of dimension 31×31 (for θ and $\dot{\theta}$), and the action space (τ) is discretized into 31 discrete points. From the learning curves, we see that the total returns are negative at the beginning and approach a value of 1 as the pendulum approaches the upright position. The trajectory of the pendulum is plotted by converting the state into the $x-y$ plane using sinusoidal mapping. However, from the example trajectories are shown we see that the pendulum does not always reach the upright position.

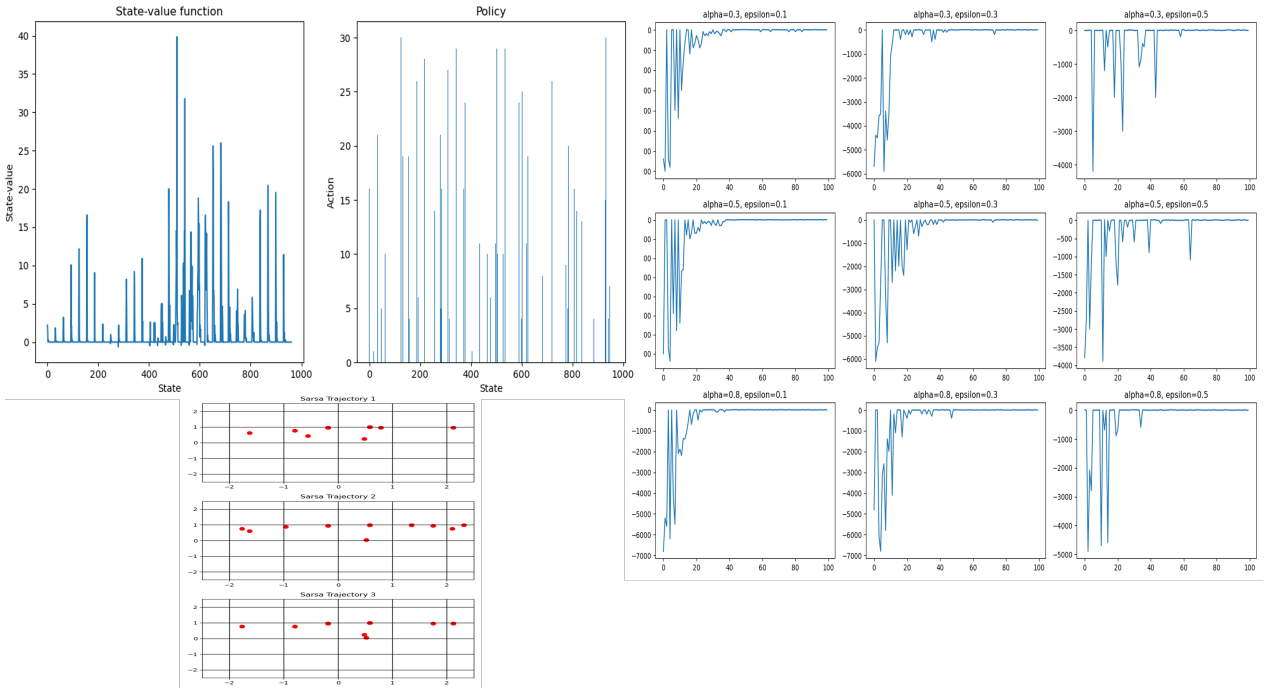

Fig. 5: SARSA results (pendulum)

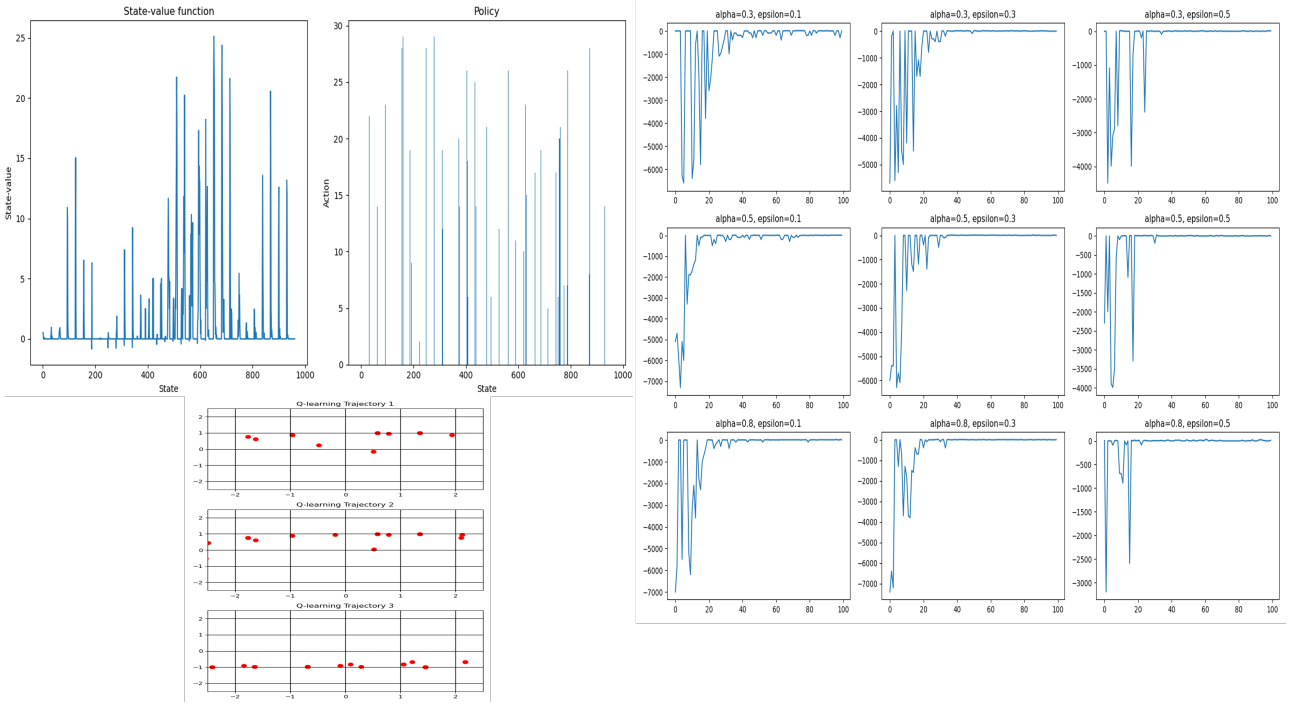

Fig. 6: Q-learning results (pendulum)