CS747 Programming Assignment 3

Anubhav Agarwal,17D070026

November 2020

1 Task 1 and Task 2

For this task, I have used the value of alpha to be equal to 0.5 and ϵ=0.1. Hence, I am exploring with the probability 0.1 and greedy exploitation with probability 0.9.

I have kept the states as a tuple of x and y coordinates. The final destination lies at the origin. Thus, when the agent arrives at the origin, the while loop exits.

In order to evaluate this task, please run the file core.py in the submission folder(no arguments/flags required). The function train() runs sarsa algorithm for 10000 steps and returns a tuple of lists with x and y values(timesteps vs no. of episodes). Subsequently, a figure is plotted using matplotlib as shown below.

The function is called 10 more times and average is taken before plotting the final result.

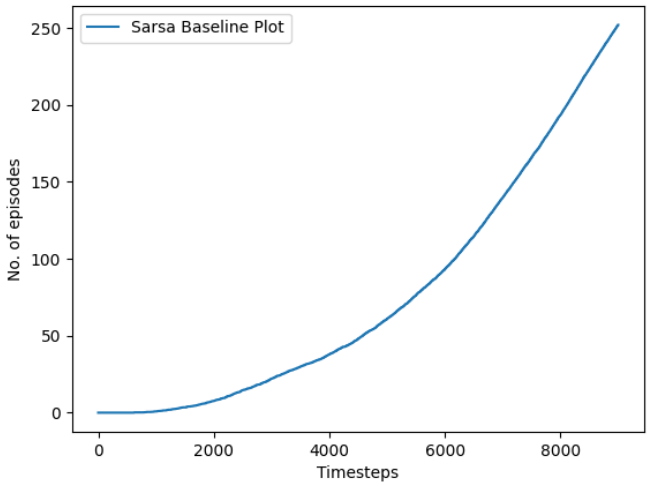

Figure 1: Sarsa Baseline Plot in task 2 with alpha=0.5 and epsilon=0.1

2 Task 3

This is a modification to the implementation in above task. All the parameters have been kept same except for the fact that now we can choose from 8 moves. Since the agent has greater freedom now, the number of steps to reach the destination will also decrease. The plot obtained is attached below:

To run this task, please run the file task3.py.(no arguments/flags required)

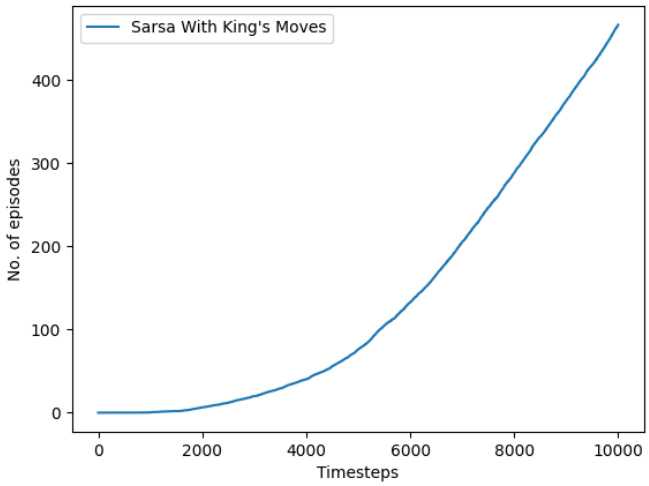

Figure 2: Sarsa Plot in task 3 with king's moves, alpha=0.5 and epsilon=0.1

3 Task 4

In this task, stochasticity has been added in the displacement due to wind. As a result, the agent may require more number of steps to reach destination as learning also gets affected. Hence, number of episodes in the same number of timesteps gets reduced significantly.

To run this task, run the file task4.py in submission folder(no arguments/flags required).

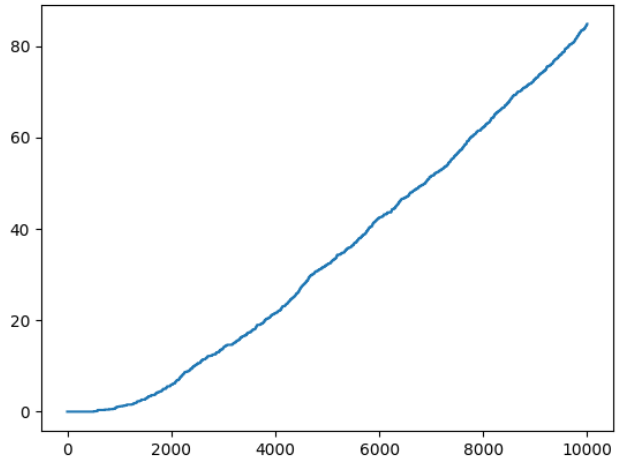

Figure 3: Sarsa Plot in task 4 with king's moves and stochasticity in wind, alpha=0.5 and epsilon=0.1

4 Task 5

Here, the three control methods have been implemented in separate functions in the file task5.py. By running the file(no arguments/flags needed), a figure is generated with the three plots comparing Q-learning, Sarsa and Expected Sarsa. Q-learning achieves the best performance followed by Expected Sarsa and Sarsa. The plot obtained is as follows:

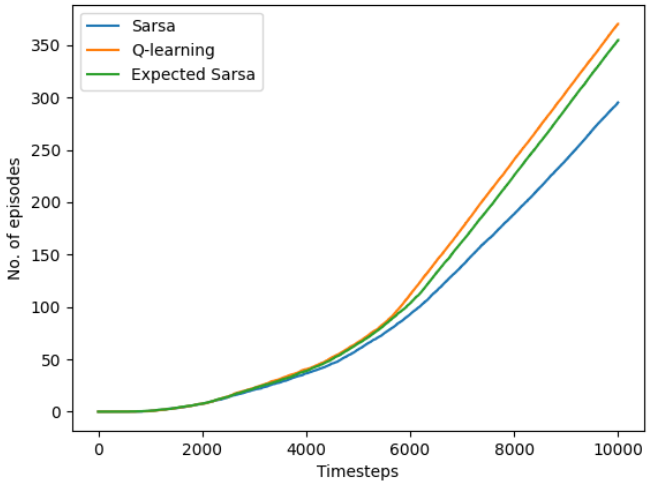

Figure 4: Sarsa Plot in task 5 comparing Q-learning, Expected Sarsa and Sarsa alpha=0.5 and epsilon=0.1