CS 747 Assignment 3 Report

Rishabh Dahale
17D070008

November 13, 2020

1 Windy Gridworld

For the task of windy gridworld, I have used $\epsilon = 0.1$ and $\alpha = 0.5$. Annealing of the learning rate is not done. When on boundary, if any move results in next place of outside of grid, the final place is clipped to the boundary after calculating the net effect. For e.g. if we are on the top row in a column with wind 1, and we move one step to the right, then the final position will be on the boundary on the column to the right of current column. This can be seen in the table below. CS is the current state, NS is the next state.

	0	CS	NS	0
Wind	0	1	0	0

All the 3 algorithms i.e. Sarsa(0), Q-Learning and Expected Sarsa were run on this grid. The plots of this run can be seen in the figure 1 The highlighted region around the lines are the $\mu \pm \sigma$ region where σ is the standard deviation from the experiments. The experiments was run over 20 random seeds.
It can be seen that Q-Learning performs the best and have the least variance. This is because it is an off-policy algorithm, Due to this, at every time step it updates the Q values according to the best policy present at that time. This can be seen in Sarsa also. If we look at the update equation of the sarsa carefully

$$\hat{Q}^{t+1}(s^t, a^t) \leftarrow \hat{Q}^t(s^t, a^t) + \alpha\{r^t + \gamma\hat{Q}^t(s^{t+1}, a^{t+1}) - \hat{Q}^t(s^t, a^t)\} \tag{1}$$

Here a^{t+1} is choosen according to policy at time t. This means that it will chose the optimal action with probability $1-\epsilon$ and a random action with a probability of ϵ. Becuase of this, it will perform same update as that of Q-Learning for $1 - \epsilon + \frac{\epsilon}{N}$ times, where N is the number of actions. As this is very similar to that of Q-Learning, we can see that the graphs are close.
Expected Sarsa is also an on-policy update. It shows the worst performance and highest variance among the 3 algorithms.

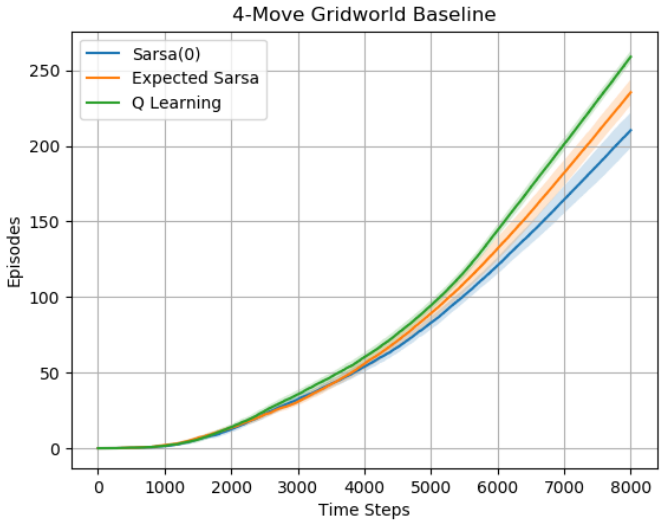

Figure 1: Baseline Performances of Different Algorithm

2 Sarsa(0)

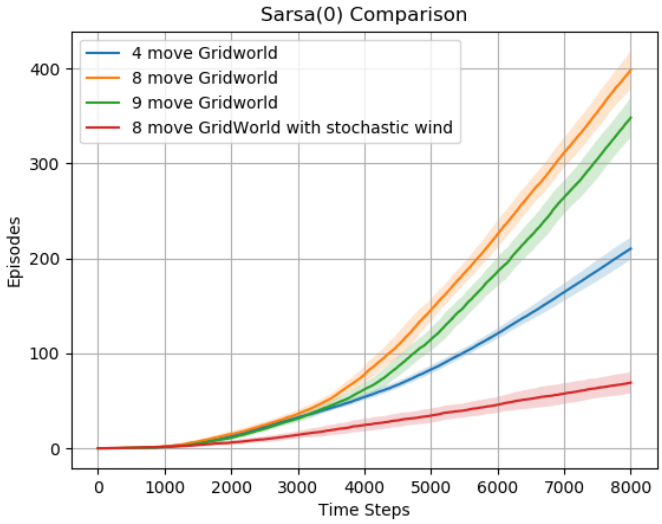

Figure 2: Performance of Sarsa(0) for various setting of the Windy Gridworld Problem

From the figure 2 it can be seen that Sarsa algorithm performs the best when King's move are allowed. This is because it can move diagonally which will result in shorter path than the one with 4 move gridworld. In the case when the wind is made stochastic, it can be seen that the we get the least episodes for a given time steps. This is expected because as the wind is stochastic, we cannot predict exactly what will be the result due to wind. For some episodes it may favour us for some it may not. Because of this, on an average it will take more steps to complete an episode.

It can be also seen that the standard deviation of the 4 move task is lower than that when King's moves are allowed. This is because in 4 move case, the number of sub-optimal actions are fewer. This result in higher net probability of selecting optimal arm $(1 - \epsilon + \frac{\epsilon}{N})$.