Programming Assignment 4

1 Running the Program

The `q_learn.py` is the training script

This script saves training results in a numpy file, and this script takes a additional parameter `-name` which is the file name to which the results are saved.

for example: `$ q_learn.py -alg q -size 3 -exps 100 -gamma 0.8 -epsilon 0.1 -name first_run`

would produce a new file at `./first_run.npy`

Use `tune.py` to perform parameter searches and save all the results.

To plot and visualize those `npy` files, we use `visualize.py`, see `visualize.py` for details of how to plot graphs.

2 Q learning

2.1 Learning Rate

As shown in Figure 1 that as we increase the learning rate α, the convergence becomes quicker. The extreme case being $\alpha = 0$ where the Q value never gets updated, hence nothing is learned.

However, setting learning rate too high might cause unstable Q value estimates (see Figure 1 where $\alpha = 1$) during learning or even divergence.

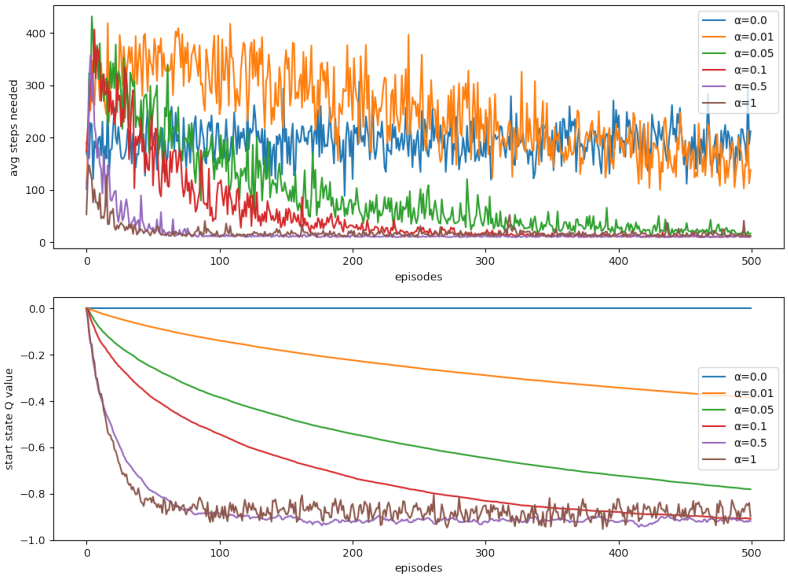

Figure 1: Q Learn with $\epsilon = 0.1$

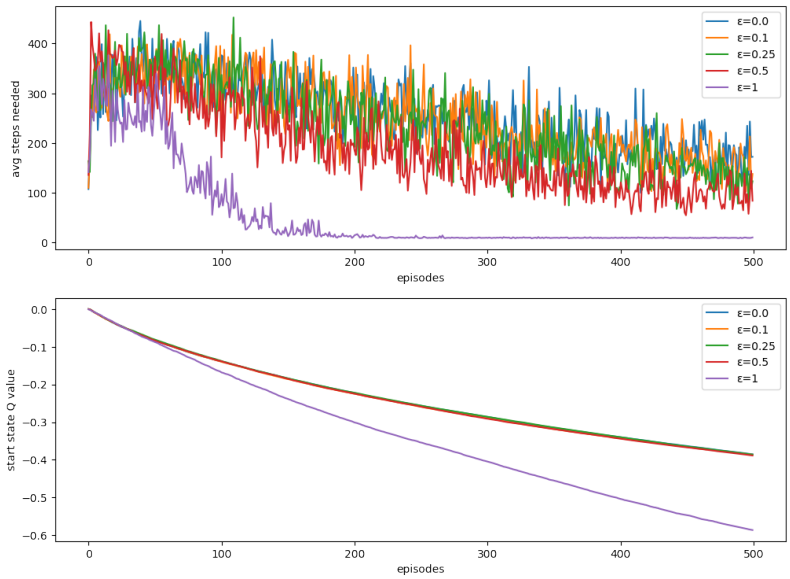

Figure 2: Q Learn with $\alpha = 0.01$

2.2 Epsilon

With a high ϵ, we are able to explore more and converge to optimal value much quicker. See Figure 2 where $\epsilon = 1$. Being such an explorer, in this case, does not incur a high exploration cost due to the state space being small ($|S| = 25$).

In the case where Learning Rate is high (Figure 3), however, we do not observe high ϵ being much helpful in convergence speed; On the other hand, high ϵ helps the Q to stay stable even under the high Learning Rate. Again, due to the state and action space being simple, we do not pay high exploration cost.

3 SARSA(λ)

With a high λ, we are able to trace and update the value of the initial state very quickly, as shown in Figure 4 where $\lambda = 1$. Also notice that, in the above case, the Q values not unstable, an intuitive explanation would be that our eligibility trace is not decaying properly so that unnecessary blames are set onto the initial state. Observe that this problem seize to exist when we decrease λ.

On the other hand, increasing λ (not by too much) makes convergence faster, Notice that SARSA in general is much faster in terms convergence speed than Q Learning, one can tell by comparing Figure 4 with the red line ($\alpha = 1$) in Figure 1.

An abnormality we can see in Figure 4 is that we have an explosion in "avg steps needed" at the start of the training. This is an result of SARSA always takes an action before updating Q values, resulting in the agent always heading downward before the reward of terminal state can propagate back.

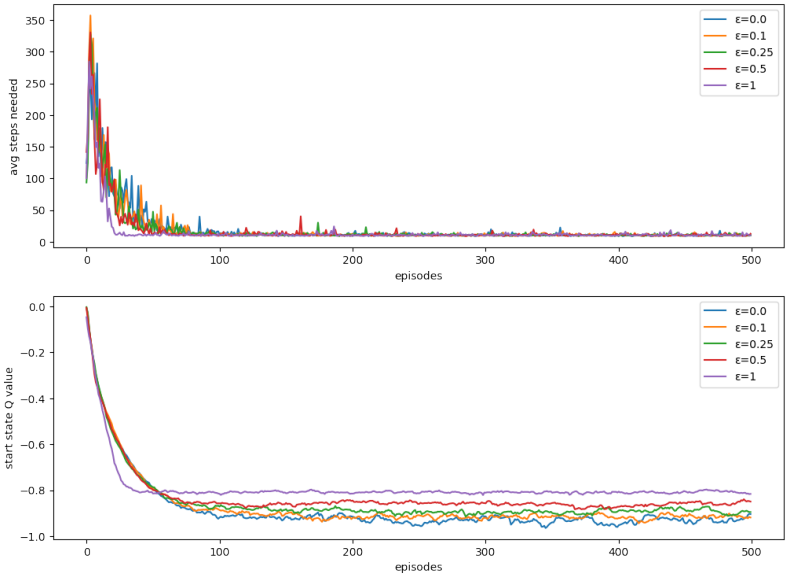

Figure 3: Q Learn with $\alpha = 0.5$

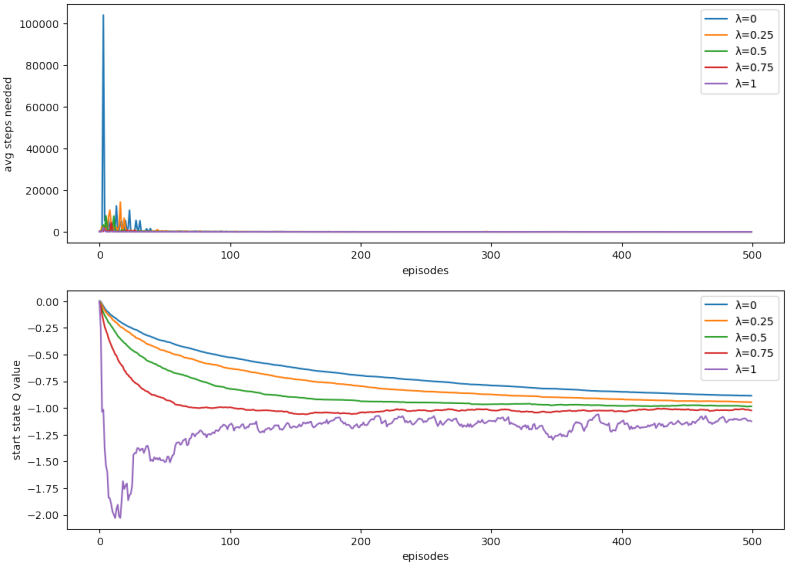

Figure 4: SARSA with $\alpha = 0.1, \epsilon = 0.1$

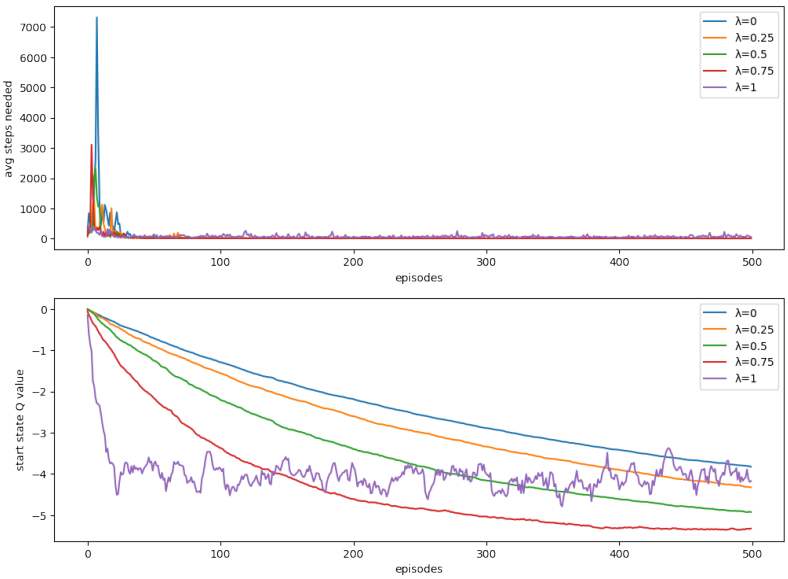

Figure 5: SARSA with $\alpha = 0.1, \epsilon = 1$

3.1 The Effect of ϵ

Different from Q Learning, we observe obvious exploration cost when we increase epsilon by comparing Figure 4 and Figure 5. Also observe that if combined with high λ, the instability would cause the values to wobble or even diverge. This is due to when the agent is absolutely exploring, the eligibility trace, generated from a random path, does not give much insight as it should.

4 10×10 Environment

4.1 Learning Rate

As we can see in Figure 6 that when learning rate is high ($\alpha = 0.5$) we get the best performance (but unstable value estimations). Although, the optimal value might be somewhere between 0.25 and 0.5, a more detailed search is not necessary for our purpose.

4.2 Epsilon

From Figure 7, we see that, initial state value-wise, different ϵ values performs about the same. However, we see $\epsilon = 0.1$ actually converged fastest, at around 70 episodes. Again, same for all parameter searching, the true optimal value for this environment is improbable to be exactly 0.1, my guess is somewhere between 0.1 and 0.5.

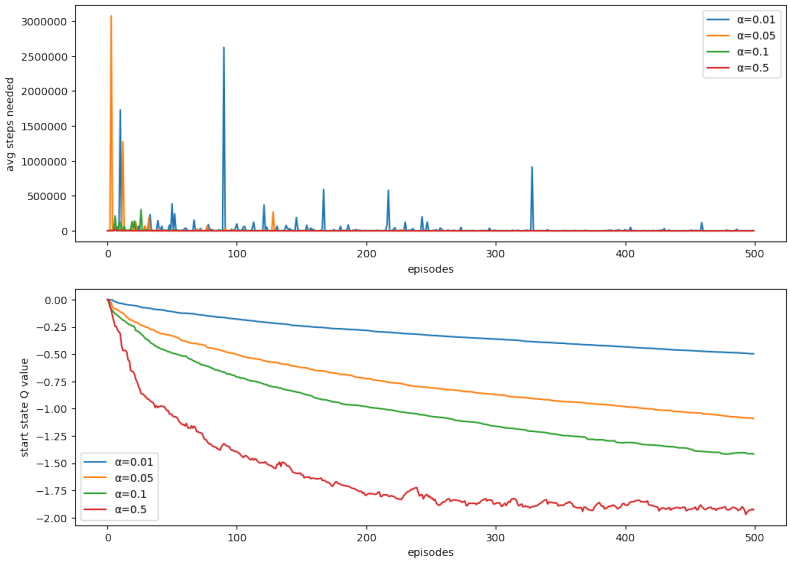

Figure 6: 10×10 Environment with $\epsilon = 0.25$

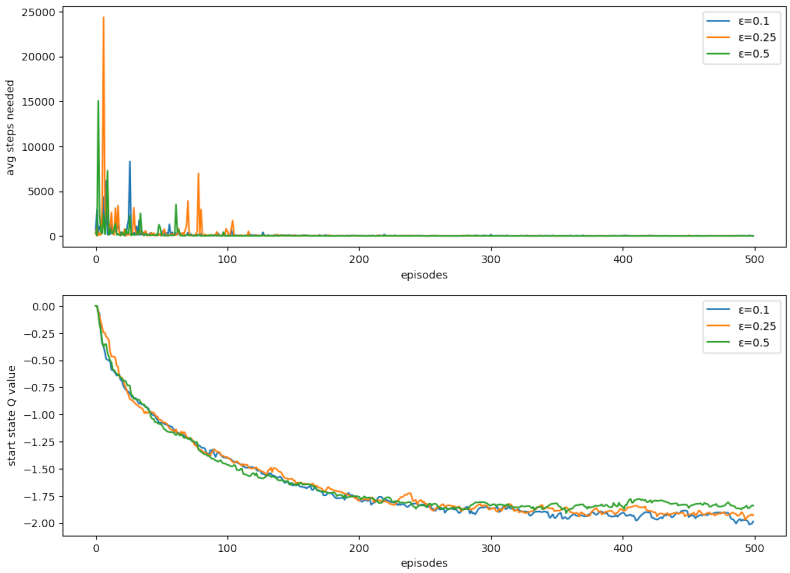

Figure 7: 10×10 Environment with $\alpha = 0.5$

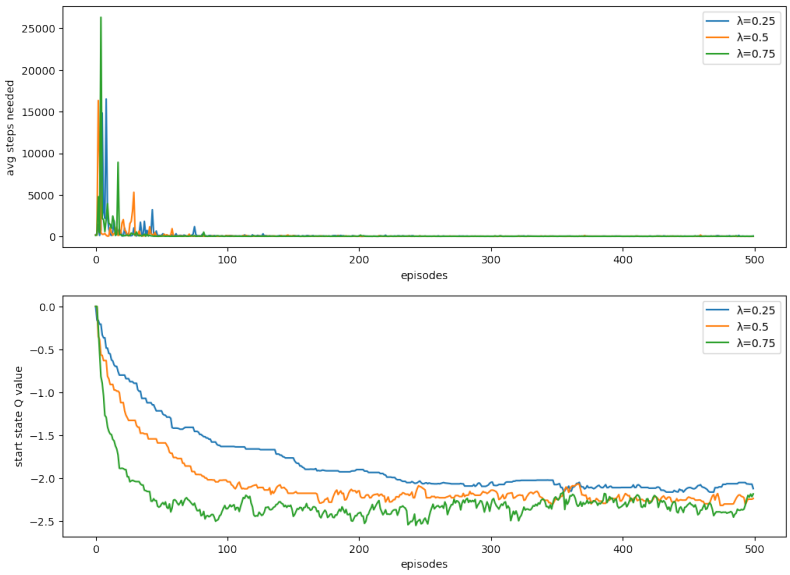

Figure 8: 10×10 Environment with $\alpha = 0.5, \epsilon = 0.1$

4.3 SARSA(λ) parameters

High λ is particularly helpful now that we come to a larger state space. We we tell from Figure 8 that with a $\lambda = 0.5$, we achieve convergence after around 65 episodes. Therefore, a good set of parameters would be $\lambda = 0.5, \alpha = 0.5, \epsilon = 0.1$. We keep epsilon low because since we are not annealing ϵ, low ϵ help us to because closer to optimal policy.

5 Thoughts and Observations for $\lambda, \alpha, \epsilon$

5.1 α

We see from several example from above that α directly corresponds to training efficiently as long as it is not high enough to cause value estimating to wobble or diverge.

5.2 ϵ

High ϵ typically means longer convergence time. However, setting ϵ too low or to zero also increases convergence time (but will eventually converge because our environment is simple).

High ϵ also has a negative impact when using SARSA(λ), since a high exploratory agent's eligibility trace does not reflect its "real" decisions.

5.3 λ

Setting λ too high or too low slows down convergence as shown in Figure 9. Intuitively, low λ is prone to short sighted credit assignment, while high λ is affected by exploration noise.

6 Thoughts on this Assignment

Through various experiments, one observe training progress (no. of steps needed to reach goal, and initial state's value) at every episode. Although these two metrics are insightful and provide us enough information on how the model is able to perform, we loss one important axis: time.

The essential observation is that not every episode takes the same amount of time. Therefore, comparing convergence efficiency between models not only require one to know how many episodes they each needed, but also how long did these episodes take (no. of steps). For example, in cases where epsilon is extremely high, each episode takes exponential amount of time since our agent can only get to the terminal state by luck no matter its knowledge of the environment.

This is just something on which we may be able to improve if this assignment was to be done a second time. (perhaps by future students)

7 Appendix

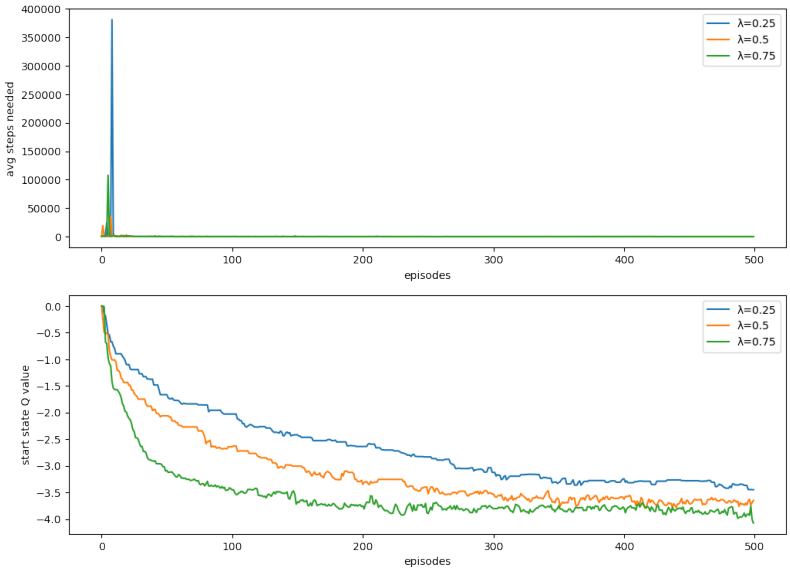

Figure 9: 10×10 Environment with $\alpha = 0.5, \epsilon = 0.1$