# CS6700 : Reinforcement Learning
## Written Assignment #3

Deadline: 31st July,2020

- This is an individual assignment. Collaborations and discussions are strictly prohibited.
- Be precise with your explanations. Unnecessary verbosity will be penalized.
- Check the Moodle discussion forums regularly for updates regarding the assignment.
- **Please start early.**

AUTHOR : Prajjwal Kumar
ROLL NUMBER : NA16B115

1. (3 marks) Consider the problem of solving POMDPs using Deep Reinforcement Learning. Can you think of ways to modify the standard DQN architecture to ensure it can remember histories of states. Does the experience replay also need to be modified? Explain.

**Solution:**
In Q-Learning update, Q-values are updated as:

$$Q(s,a) = Q(s,a) + \alpha(r + \gamma \text{max} Q(s',a') - Q(s,a)) \tag{1}$$

But for POMDPs, dealing with Q-values using above equation is tough. So, ways to stabilize the learning :

Using experience replay: Memorize a fixed no of experiences in replay memory and sampling them uniformly.
Separate Target Network: This is done to provide state update targets to main network.
RMSProp: It is a learning method for regulating parameter adjustment rate of the network.

2. (4 marks) Exploration is often ameliorated by the use of counts over the various states. For example, one could maintain a visitation count $N(s)$, for every state and use the same to generate an intrinsic reward $(r_i(s))$ for visiting that state.

$$r_i(s) = \tau \times \frac{1}{N(s)}$$

However, it is intractable to maintain these counts in high-dimensional spaces, since the count values will be zero for a large fraction of the states. Can you suggest a solution(s) to handle this scenario? How can you maintain an approximation of the counts for a large number of states and possibly generalize to unseen states?

**Solution:** We can't generalize the above said counts across the state space in high dimensional spaces. Even if a un-visited state has some features matching with some already visited state, instead of using this information, the agent treats the un-visited state as a completely new state. For making this algorithm more efficient, it needs to be made sure that the agent really visits completely unrelated states in order to widen its scope of learning.
Here, we can use Approximate RL;
1. Avoiding the Bellman's curse of dimensionality, learning happens in a reasonable time and space.
2. We can easily generalize new states here. We can either approximate the value function or search in the policy space or both for solving that.

3. (5 marks) Suppose that the MDP defined on the observation space is k-th order Markov, i.e. remembering the last k observations is enough to predict the future. Consider using a belief state based approach for solving this problem. For any starting state and initial belief, the belief distribution will localize to the right state after k updates, i.e., the true state the agent is in will have a probability of 1 and the other states will have a probability of 0. Is this statement true or false? Explain your answer.

**Solution:** It is a FALSE statement.

We cannot say that after k updates, the belief distribution will localize to the right state as method predicting it can contain errors. Some other states may have same probability of the agent localizing to them.

4. (3 marks) Q-MDPs are a technique for solving the problem of behaving in POMDPs. The behavior produced by this approximation would not be optimal. In what sense is it not optimal? Are there circumstances under which it can be optimal?

**Solution:** In a QMDP technique used for solving problems in POMDPs, the corresponding MDP is solved. Then, as there are partial observations, a Q-function is converted to a value function over belief states and actions.

$$Q_a(b) = \sum_a b(s) Q_{MDP}(s, a) \tag{2}$$

So, for a MDP, we can be sure of a policy being optimal only when we know the state of agent at all points but for a POMDP, it is not possible as we know the partial observation of the states in it as the name says. Therefore, the policy is optimal only if the POMDP is completely observable.

5. (3 marks) What are some advantages and disadvantages of A3C over DQN? What are some potential issues that can be caused by asynchronous updates in A3C?

**Solution:**
ADVANTAGES OF A3C:
In DQN, experience replay(sampling from replay memory) is used and is computationally as well as memory-wise intensive. While in A3C, multiple agents are working in parallel in multiple instances of the environment asynchronously. So, these set of parallel agents may at be different states at some given time step leading to no relations in training samples and more stationary distribution, subsequently stabilizing the learning of the algorithm. In addition to this, A3C also allows the usage of on-policy methods like SARSA with off-policy ones but DQN only allows the later.

DISADVANTAGES OF A3C:
As we know, in A3C, multiple agents work in asynchronous manner in different instance of the environments and they update the parameters at different time leading to continuous change of common objective function. It goes on till the last update meaning these updates do not contribute to the learning and reduce performance of the algorithm.

6. (6 marks) There are a variety of very efficient heuristics available for solving deterministic travelling salesman problems. We would like to take advantage of such heuristics in solving certain classes of large scale navigational problems in stochastic domains. These problems involve navigating from one well demarcated region to another. For e.g., consider the problem of delivering mail to the office rooms in a multi storey building.

(a) (4 marks) Outline a method to achieve this, using concepts from hierarchical RL.

**Solution:** Ant Colony Optimisation (ACO) is one of the most used solution of the travelling salesman problem(TSP). Commonalities between Ant system and Q-learning were used to make an improved method of ACO called Ant-Q.
Using HRL method, we kinda divide the problems into different clusters making sub-tasks of the problem. For the given example, the sub-tasks can be choosing the floor of the building first and then choosing the room where the mail is to be delivered.

ACO is based on ants movement. While moving on some path, ants secret pheromones to keep a memory of the path and this ultimately leads to finding a trail with shortest path. The above said AntQ algorithm along with the MAXQ algorithm making a hybrid of HRL and ACO. Also, a clustering algorithm is also added to find clusters to solve TSP. Like in this case, each cluster for can contain information for each floor of the building.

The complete algorithm can proceed as following:

1. Let there be a tour $t$. First clustering algorithm is used to find clusters if present or generate them.
2. Then, average for each cluster is calculated.
3. A TSP is created using those averages and then solved using HRL and ACO hybrid or AntQ and MAXQ.
4. Combine the tour or route for all clusters.

(b) (2 marks) What problems would such an approach encounter?

**Solution:** This approach has order like $(O(N))$. So, as we go for higher state space, it will take more time to compile. Also, as the dimensions keep increasing, the solution keeps losing its optimality.
Thus, this approach is good with smaller state spaces.

7. (6 marks) This question may require you to refer to https://link.springer.com/content/pdf/10.1007/BF paper on average reward RL. Consider the 3 state MDP shown in Figure 1. Mention the recurrent class for each such policies. In the average reward setting, what are the corresponding $\rho^{\pi}$ for each such policy ? Furthermore, which of these policies are gain optimal ?

**Solution:** States A and B are recurrent states for action a1 forming recurrent class with a period of 2.
Similarly, A and C are recurrent states for action a2 forming recurrent class with a period of 2.
While, state C is recurrent for action a3.

(a) (3 marks) What are the different deterministic uni-chain policies present ?

**Solution:** An MDP is termed uni-chain if the transition matrix corresponding to every policy contains a single recurrent class, and a (possibly empty) set of transient states.

(b) (3 marks) In the average reward setting, what are the corresponding $\rho^{\pi}$ for each such policy ? Furthermore, which of these policies are gain optimal ?

**Solution:** The average reward $p^{\pi}$ associated with a particular policy at $\pi$ at a state $x$ is defined as

$$p^{\pi}(x) = \lim_{N \longrightarrow \infty} \frac{E[\Sigma_{t=0}^{N-1} R_t^{\pi}(x)]}{N} \tag{3}$$

where $R_t^{\pi}(x)$ is the reward received at time $t$ starting from state $x$ and actions are chosen under policy $\pi$. $E[\cdot]$ denotes the expected value.
The average reward MDP computes the policies that have highest expected pay-off per step.

And a gain-optimal policy is defined as the one which maximizes average reward over all states, i.e. $[p^{\pi *}(x) \geq p^{\pi}(x)$ over all policies $\pi$ and states $x]$.

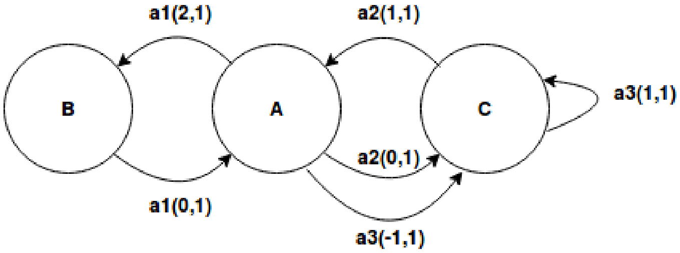

Figure 1: Notation : action(reward, transition probability). Example : a1(3, 1) refers to action a1 which results in a transition with reward +3 and probability 1

**References:**

1.https://towardsdatascience.com/the-inspiration-of-an-ant-colony-optimization-f377568ea03f

2.http://mat.uab.cat/alseda/MasterOpt/ACO_Intro.pdf

3.https://link.springer.com/content/pdf/10.1007/BF00114727.pdf