CS6700 : Reinforcement Learning
Written Assignment #3

Deadline: 2 April 2019, 11:55 pm

- 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.**

1. Let us consider the effect of approximation on policy search and value function based methods. Suppose that a policy gradient method uses a class of policies that do not contain the optimal policy; and a value function based method uses a function approximator that can represent the values of the policies of this class, but not that of the optimal policy.
   (a) (2 points) Why would you consider the policy gradient approach to be better than the value function based approach?
   (b) (2 points) Under what circumstances would the value function based approach be better than the policy gradient approach?
   (c) (2 points) Is there some circumstance under which either of the method can find the optimal policy?

2. You are given an MDP, with states s_1 , and s_2 and actions a_1 and a_2 . Suppose the states s are represented by three features, $\phi_1(s)$, $\phi_2(s)$ and $\phi_3(s)$, where $\phi_1(s_1) = 1$, $\phi_1(s_2) = -1$, $\phi_2(s_1) = -1$, $\phi_2(s_2) = -1$, $\phi_3(s_1) = -1$ and $\phi_3(s_2) = 1$.
   (a) (5 points) What class of state value functions can be represented using only these features in a linear function approximator? Explain how you arrived at your answer.
   (b) (3 points) Give the explicit backup for each parameter for state s_2 for linear, gradient descent TD(0) assuming the experience: $s_2, a_2, -5, s_1, a_1$.

3. When we implement TD(λ) using a linear function approximator(FA), we need to maintain an eligibility trace for each parameter in the FA
   (a) (4 points) Give a complete specification (pseudo code or algorithm) for such an implementation.
   (b) (2 points) What form of eligibility traces are more appropriate in this case: replacing or accumulating?

4. Answer the following questions with respect to the DQN algorithm:

- (2 points) When using one-step TD backup, the TD target is $R_{t+1} + \gamma V(S_{t+1}, \theta)$ and the update to the neural network parameter is as follows:

$$\Delta\theta = \alpha(R_{t+1} + \gamma V(S_{t+1}, \theta) - V(S_t, \theta))\nabla_\theta V(S_t, \theta) \tag{1}$$

  Is the update correct ? Is any term missing ? Justify your answer

- (2 points) Describe the two ways to update the target network. Which one is better and why ?

5. (4 points) What is the role of the experience replay in DQN? Consequent works in literature sample transitions from the experience replay, in proportion to the TD-error. Hence, instead of sampling transitions using a uniform-random strategy, higher TD-error transitions are sampled at a higher frequency. Why would such a modification help?

6. (3 points) We discussed two different motivations for actor-critic algorithms: the original motivation was as an extension of reinforcement comparison, and the modern motivation is as a variance reduction mechanism for policy gradient algorithms. Why is the original version of actor-critic not a policy gradient method?