CS6700 : Reinforcement Learning
Written Assignment #2

Topics: Adv. Value-based methods, POMDP, HRL **Deadline**: 30 April 2023, 11:59 pm
Name: Purvam Jain **Roll Number: EE20B101**

- 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.
- Type your solutions in the provided LaTeXtemplate file.
- **Please start early.**

1. (3 marks) Recall the four advanced value-based methods we studied in class: Double DQN, Dueling DQN, Expected SARSA. While solving some RL tasks, you encounter the problems given below. Which advanced value-based method would you use to overcome it and why? Give one or two lines of explanation for 'why'.

   (a) (1 mark) Problem 1: In most states of the environment, choice of action doesn't matter.

   Solution: Dueling DQN: It estimates Q-values using Advantage A(s,a) and Value V(s). Due to this agent learns which actions are optimal and which states are more valuable, so the architecture can learn which states are more important irrespective of the action taken.

   (b) (1 mark) Problem 2: Agent seems to be consistently picking sub-optimal actions during exploitation.

   Solution: Double DQN: This problem is caused due to overestimation in Q-Learning, which can be overcome using two Q-value functions to reduce overestimation.

   (c) (1 mark) Problem 3: Environment is stochastic with high negative reward and low positive reward, like in cliff-walking.

   Solution: Expected SARSA: Stochastic environments have high variance, which can be reduced as expected sarsa computes the expected Q-value over all possible actions and improves upon the sarsa.

2. (4 marks) Ego-centric representations are based on an agent's current position in the world. In a sense the agent says, I don't care where I am, but I am only worried about the position of the objects in the world relative to me. You could think of the agent as being at the origin always. Comment on the suitability (advantages and disadvantages) of using an ego-centric representation in RL.

Solution:

Advantages:

1. Reduces complexity of state space and hence computation.
2. Greedy with respect to immediate state/actions as it will quickly learn to avoid states which have a high negative reward as well as reach the goal if it is within its current space representation.
3. Easily transferable policies are learnt specifically for tasks in which relative position is more important.

Disadvantages:

1. Agent doesn't learn about important global states and fails to capture environment dynamics
2. Might get stuck in local minima
3. Might not converge for tasks with long-term dependency as it only cares about immediate rewards

3. (12 marks) Santa decides that he no longer has the memory to store every good and bad deed for every child in the world. Instead, he implements a feature-based linear function approximator to determine if a child gets toys or coal. Assume for simplicity that he uses only the following few features:

   - Is the child a girl? (0 for no, 1 for yes)
   - Age? (real number from $0 - 12$)
   - Was the child good last year? (0 for no, 1 for yes)
   - Number of good deeds this year
   - Number of bad deeds this year

   Santa uses his function approximator to output a real number. If that number is greater than his good threshold, the child gets toys. Otherwise, the child gets coal.

   (a) (4 marks) Write the full equation to calculate the value for a given child (i.e., $f(s, \vec{\theta}) = \ldots$), where s is a child's name and $\vec{\theta}$ is a weight vector $\vec{\theta} = (\theta(1), \theta(2), \ldots, \theta(5))^{\mathrm{T}}$. Assume child s is described by the features given above, and that the feature values are respectively written as ϕ_s^{girl}, ϕ_s^{age}, ϕ_s^{last}, ϕ_s^{good}, and ϕ_s^{bad}.

   Solution: $f(s, \vec{\theta}) = \vec{\theta}^T \cdot \phi_s$
   $f(s, \vec{\theta}) = \theta(1)\phi_s^{\text{girl}} + \theta(2)\phi_s^{\text{age}} + \theta(3)\phi_s^{\text{last}} + \theta(4)\phi_s^{\text{good}} + \theta(5)\phi_s^{\text{bad}}$

   (b) (4 marks) What is the gradient $\left(\nabla_{\vec{\theta}} f(s, \vec{\theta})\right)$? I.e. give the vector of partial derivatives

   $$\left(\frac{\partial f(s,\vec{\theta})}{\partial \theta(1)}, \frac{\partial f(s,\vec{\theta})}{\partial \theta(2)}, \cdots, \frac{\partial f(s,\vec{\theta})}{\partial \theta(n)}\right)^{\mathrm{T}}$$

   based on your answer to the previous question.

Solution: The gradient $\left(\nabla_{\vec{\theta}} f(s,\vec{\theta})\right)$ is:
$\left(\frac{\partial f(s,\vec{\theta})}{\partial\theta(1)}, \frac{\partial f(s,\vec{\theta})}{\partial\theta(2)}, \cdots, \frac{\partial f(s,\vec{\theta})}{\partial\theta(n)}\right)^{\mathrm{T}} = (\phi_s^{\text{girl}}, \phi_s^{\text{age}}, \phi_s^{\text{last}}, \phi_s^{\text{good}}, \phi_s^{\text{bad}})^{\mathrm{T}}$

(c) (4 marks) Using the feature names given above, describe in words something about a function that would make it impossible to represent it adequately using the above linear function approximator. Can you define a new feature in terms of the original ones that would make it linearly representable?

Solution: It would be impossible to represent if we have a complex function which includes say square of the good deeds or any other kind of non-linearity such as some kind of operation between two features like product of age and bad deeds. Such complex functions would be impossible to represent using linear approximator.

It can be made linear by adding new features as square of good deeds and age times bad deeds in the above example. As a general rule we may use SVMs to cast interactions to higher dimension where they are linearly separable.

4. (5 marks) We typically assume tabula rasa learning in RL and that beyond the states and actions, you have no knowledge about the dynamics of the system. What if you had a partially specified approximate model of the world - one that tells you about the effects of the actions from certain states, i.e., the possible next states, but not the exact probabilities. Nor is the model specified for all states. How will you modify Q learning or SARSA to make effective use of the model? Specifically describe how you can reduce the number of *real* samples drawn from the *world*.

Solution: Model Based Learning: We can use the model to generate possible transitions which can be used to guide exploration and as artificial training data to update Q-value function in Q-learning and SARSA settings.For Q-Learning:
$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left(r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\right)$
Here, next state s', action a' and reward r can be simulated from the model.
Similarly, for SARSA, we can use the model to simulate transitions and update the action-value function using the simulated rewards and the estimated action-value of the next state and action. The update equation would be modified to:
$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left(r_{t+1} + \gamma Q(s_{t+1}, \pi(s_{t+1})) - Q(s_t, a_t)\right)$
Here $Q'(s', a')$ is the action-value of the next state and action estimated using the model, and r is the simulated reward obtained from the model.
Another way could be initialising the Q-Values such that we keep high negative values to avoid possible state-action pairs which have negative rewards(eg:cliff in cliffwalking) and have high positive Q-values for states near the goal state.

5. (4 marks) We discussed Q-MDPs in the class as a technique for solving the problem of behaving in POMDPs. It was mentioned that 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: By employing Q-MDP, we are only considering current partial observations and not the full history of observations; as such, it may act optimal for current input but might not be optimal for the actual POMDP. For example, in the grid world we saw in class where multiple states might

have similar representations, in such symmetric conditions, even if one state is closer to the goal and another state is further away from the goal, they will get the same Q-value updates and hence will lead to a sub-optimal policy.
It can be optimal in cases where the observation space is small, and the degree of uncertainty is low. Say in cliffwalking example if only the last row close to the cliff is taken as environment. Here all states will be symmetric and we will get an optimal policy.

6. (3 marks) This question requires you to do some additional reading. Dietterich specifies certain conditions for safe-state abstraction for the MaxQ framework. I had mentioned in class that even if we do not use the MaxQ value function decomposition, the hierarchy provided is still useful. So, which of the safe-state abstraction conditions are still necessary when we do not use value function decomposition.

Solution: Subtask Irrelevance and Leaf Irrelevance are the necessary conditions even when we don't use value function decomposition.

7. (4 marks) One of the goals of using options is to be able to cache away policies that caused interesting behaviors. These could be rare state transitions, or access to a new part of the state space, etc. While people have looked at generating options from frequently occurring states in a goal-directed trajectory, such an approach would not work in this case, without a lot of experience. Suggest a method to learn about interesting behaviors in the world while exploring. [*Hint: Think about pseudo rewards.*]

Solution: Just like in Dyna-Q and Intrinsic Motivation approaches, we can give pseudo rewards for rare action sequences or for reaching desired states.
For example, in the grid world where pits surrounded the shorter route to the goal (Hence negative Q-values to go to that side), by associating a high pseudo-reward for exploring such states, the agent can learn the optimal path instead of taking the longer route.