CUB Reinforcement Learning, Fall 2024

Answers for Test Exam

Task 1. General policy improvement result says that any policy $\tilde{\pi}(s)$ is better than policy $\pi(s)$ if $Q^\pi(s, \tilde{\pi}(s)) > Q^\pi(s, \pi(s))$ at least for one s. So let's find Q function for some state that can improve the given policy.

Let's consider the state b and find V function for this state using Bellman equation:

$$V^\pi(s) = \mathbb{E}_{\pi(a|s)}(r(s,a) + \gamma \mathbb{E}_{p(s'|s,a)} V^\pi(s')).$$

Here with probability 1/2 we get a reward 2 and remain in state b and with probability 1/2 get a reward 4 and terminate. So

$$V^\pi(b) = \frac{1}{2}(2 + \gamma V^\pi(b)) + \frac{1}{2}4 = 3 + \frac{1}{4}V^\pi(b).$$

Hence $V^\pi(b) = 4$. Now let's compute Q function for this state using Bellman equation

$$Q^\pi(s,a) = r(s,a) + \gamma \mathbb{E}_{p(s'|s,a)} V^\pi(s').$$

For blue action $Q(b, \text{blue}) = 2 + \gamma V^\pi(b) = 2 + \frac{1}{2}4 = 4$. For red action $Q(b, \text{red}) = 4$. So both actions in state b have the same value of Q function, and hence the policy cannot be improved for this state.

Let's do similar computations for the state c. Here under the policy π we always take the red action, hence $V^\pi(c) = -1$. For Q function $Q(c, \text{blue}) = 0 + \gamma(0.8 * V^\pi(b) + 0.2 * V^\pi(c)) = \frac{1}{2}(0.8 * 4 + 0.2 * (-1)) = \frac{3}{2}$ and $Q(c, \text{red}) = -1$. Here we see that choosing blue action is always beneficial and hence the new policy $\tilde{\pi}$ that in the state c choose blue action with probability one would be a policy improvement.

For state a we can get the following:

$$\begin{aligned}
&V^\pi(a) = 0.25 * (0 + \gamma * (0.75 * 0 + 0.25 * 4)) + 0.75 * (0 + \gamma * (-1)) = -\frac{1}{4},\\
&Q^\pi(a, \text{blue}) = 0 + \gamma * (0.75 * 0 + 0.25 * 4) = \frac{1}{2},\\
&Q^\pi(a, \text{red}) = 0 + \gamma * (-1) = -\frac{1}{2}.
\end{aligned}$$

So here we see that taking blue action is more beneficial, hence the new policy $\tilde{\pi}$ that in the state a choose blue action with probability one would be a policy improvement.

Task 2. For tabular MDP:

- ☐ Optimal policy may not exist;
- ☐ There exists exactly one optimal (and deterministic) policy;
- ☐ There exists at least one optimal policy, but all of them are deterministic;
- ☑ There exists at least one optimal policy and among these policies at least one is deterministic;
- ☐ None of the above.

Task 3.

$$\begin{aligned}
&Q^*(s,a) = r(s,a) + \gamma \mathbb{E}_{p(s'|s,a)} \max_{a'} Q^*(s', a'),\\
&V^*(s) = \max_a (r(s,a) + \gamma \mathbb{E}_{p(s'|s,a)} V^*(s')),\\
&Q^*(s,a) = r(s,a) + \gamma \mathbb{E}_{p(s'|s,a)} V^*(s'),\\
&V^*(s) = \max_a Q^*(s,a).
\end{aligned}$$

Task 4. Here let's compute optimal V function from the Bellman equation:

$$V^*(s) = \max_a \mathbb{E}_{p(s'|s,a)}(r(s,a,s') + \gamma V^*(s')).$$

Note that in the given MDP the reward is dependent on state, action and the next state. That is why the expectation w.r.t. transition probability comes before the reward. Any policy that provides this maximum is optimal for the current state.

For the state c we have

$$V^*(c) = \max(1, 0.8 * (0 + \gamma V^*(c)) + 0.2 * (3)) = \max(1, 0.4 * V^*(c) + 0.6).$$

Suppose that the maximum is attained for the second case. Then $V^*(c) = 0.4 * V^*(c) + 0.6$ and hence $V^*(c) = 1$. If the maximum is attained for the first case, then again $V^*(c) = 1$. As a result, taking blue and red actions with arbitrary probabilities would be an optimal policy for the state c.

For the state b:

$$V^*(b) = \max(1 + \gamma V^*(b), 4).$$

If maximum is attained for the first case, then $V^*(b) = 1 + 0.5 * V^*(b)$ and hence $V^*(b) = 2$, that is, less than 4. Hence for the state b the optimal policy would be always choose the red action.

Finally for the state a:

$$V^*(a) = \max(0 + \gamma V^*(c), 0.5 * (-1 + \gamma V^*(a)) + 0.5 * (\gamma V^*(b))) = \max(\frac{1}{2}, -\frac{1}{2} + \frac{1}{4}V^*(a) + \frac{1}{4}4) = \max(\frac{1}{2}, \frac{1}{2} + \frac{1}{4}V^*(a)).$$

For the second case $V^*(a) = \frac{1}{2} + \frac{1}{4}V^*(a)$ and hence $V^*(a) = \frac{2}{3}$, that is, higher then $\frac{1}{2}$. As a result, in the state a the blue action should be always chosen.

Task 5. Point out off-policy RL algorithms:

□ CEM (Cross-Entropy Method)

☑ DQN (Deep Q-network)

☑ Rainbow DQN

☑ QR-DQN (Quantile Regression DQN)

□ REINFORCE

□ A2C (Advantage Actor-Critic)

□ PPO (Proximal Policy Optimization)

☑ DDPG (Deep Deterministic Policy Gradient)

☑ TD3 (Twin Delayed DDPG)

☑ SAC (Soft Actor-Critic)

Task 6. Choose correct statements about GAE estimate:

☑ In this estimate all available N-step estimates are aggregated;

☑ In this estimate user should tune a hyperparameter λ, that is reponsible for bias-variance trade-off;

□ It is required to play full episode in order to be able to compute this estimate;

□ If in GAE estimate N-step estimates only for large N are used, then GAE value may degrade and for all values of λ may have too large variance;

☑ If in GAE estimate N-step estimates only for small N are used, then GAE value may degrade and for all values of λ may have too large bias.

Task 7. Let's first enumerate all possible returns R that we can get from the given MDP. We get the return $1 + \gamma * 2 = 2$, if from the state A we get to the state B and then choose the blue action. We get the return $1 + \gamma * 0 = 1$, if from the state A we get to the state B and then choose the red action. Finally we can get $0 + \gamma * 1 = \frac{1}{2}$ and $0 + \gamma * 0$, if we get to the state C and there take either blue or red action. The probabilities for that are:

$$\begin{aligned}
&R = 2,\ p = 0.6 * 0.5 = 0.3,\\
&R = 1,\ p = 0.6 * 0.5 = 0.3,\\
&R = \frac{1}{2},\ p = 0.4 * 0.25 = 0.1,\\
&R = 0,\ p = 0.4 * 0.75 = 0.3.
\end{aligned}$$

Task 8. GAE estimate for given rollout of length n can be expressed as

$$\mathrm{GAE}(\gamma, \lambda) = \delta_0 + (\gamma\lambda)\delta_1 + (\gamma\lambda)^2\delta_2 + \cdots + (\gamma\lambda)^{n-1}\delta_{n-1}.$$

Here δ_t is a one-step temporal difference estimate $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$. For the provided information:

$$\begin{aligned}
&\delta_0 = 1 + \gamma V(s_1) - V(s_0) = 1 + 1 - 1 = 1,\\
&\delta_1 = 0 + \gamma V(s_2) - V(s_1) = 0 + 0 - 1 = -1,\\
&\delta_2 = 1 - V(s_2) = 1,\\
&GAE = \delta_0 + (\lambda\gamma)\delta_1 + (\lambda\gamma)^2\delta_2 = 1 + \frac{1}{2}(-1) + \frac{1}{4}1 = \frac{3}{4}.
\end{aligned}$$

Task 9. PPO and TRPO advantages compared to simple policy gradient methods (A2C) are in the following:

□ They can learn using data collected by arbitrary behaviour policy;

□ They better address exploration-exploitation dilemma using comparison between two policies;

□ Optimization of lower bound instead of initial criterion function allows to get unbiased gradient estimate even with non-exact critic;

□ They can use ensembles of multi-step estimates;

☑ None of the above.

Task 10. Adding policy entropy term to cumulative reward (like in SAC method) leads to:

□ Set of optimal policies does not change;

☑ Optimal policy becomes unique;

☑ Optimal policy becomes stochastic;

□ Value functions coincide with the ones from original RL formulation;

□ New value function $U(s, a)$ is introduced;

□ None of the above.

Task 11. In which RL algorithms for actor's training a gradient of Q function w.r.t. actions is used?

□ DQN

□ Policy Gradient algorithms (A2C, PPO)

☑ DDPG

☑ TD3

☑ SAC (if reparameterization trick is applicable)

Task 12. What is computed by different heads of neural network in AlphaZero?

☑ policy for given state

☑ V function value

□ Q function value

□ Exploration Bonus;

□ probabilities for next states;