Midterm Exam

ELEN E6885: Introduction to Reinforcement Learning

October 18, 2019

Problem 1 (16 Points, 2 Points each)

True or False. No explanation is needed.

1. An MDP is a mathematical formalization of an agent. Solution: False
2. Both ϵ-greedy policy and Softmax policy will balance between exploration and exploitation. Solution: True
3. One drawback of policy iteration is that each of its iterations involves policy evaluation, which may itself be an iterative computation process which requires multiple sweeps through the state space. Solution: True
4. In order to guarantee that Sarsa algorithm converges to the optimal policy, we only need to require that all state-action pairs are visited an infinite number of times. Solution: False
5. Consider the value functions v_k and v_{k+1} from two iterations of value iteration. Let π_k and π_{k+1} be the policies that are greedy with respect to v_k and v_{k+1}, respectively. It is always true that $\pi_{k+1} \geq \pi_k$, i.e., $v_{\pi_{k+1}}(s) \geq v_{\pi_k}(s)$ for any state s. Solution: False
6. Every finite MDP with bounded rewards and discount factor $\gamma \in [0, 1)$ has a unique optimal policy. Solution: False
7. For an episode $S_0, A_0, R_1, S_1, A_1, R_2, S_2, \cdots$ following policy π, $\sum_{k=0}^{\infty} \gamma^k R_{k+1}$ is an unbiased estimator of $v_\pi(S_0)$. Solution: True
8. Using ϵ-greedy policy improvement with $\epsilon = 0.1$, Q-Learning always achieves higher total reward per episode than Sarsa. Solution: False

Problem 2 (28 Points, 4 Points each)

Short-answer questions.

1. Explain what the purpose of the discount factor in infinite horizon problems is. What effect does a small discount factor have?

   Solution: The purpose of the discount factor is to give a higher value to plans that reach a reward sooner. It also bounds the utility of a state, ensuring it does not reach infinity. A small discount factor discounts future rewards more heavily. A solution with a small discount factor will be greedy, meaning that it will prefer immediate rewards more than long-term rewards.

   Rubrics: 2 points for giving any purpose of the discount factor listed above; 2 more points for answering that "small discount factor prefers immediate rewards more than long-term rewards" or something alike.

2. In real-world application, we often encounter reinforcement learning problems that are effectively non-stationary (e.g. the mean value of the random process is time-varying). In such cases, why do we usually use a constant step-size in the incremental update of an reinforcement learning algorithm, e.g. $Q_{k+1} = Q_k + \alpha[r_{k+1} - Q_k]$?

   Solution: For a non-stationary environment, we would better weight recent rewards more heavily than long-past ones.

   Rubrics: 2 points for mentioning convergence to the evolving value; 4 points for answering "put higher weights on the most recent rewards" or something alike.

3. Why do we say TD-method is a *bootstrapping* method?

   Solution: TD method bases its update in part on existing estimates.

   Rubrics: 1 point for only mentioning "TD use incomplete episode" (or not the entire episode, unterminated, etc.); 2 points for only writing down the update formula.

4. Is Q-learning on-policy or off-policy learning method? Explain your answer. The Q-learning update is given by $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$.

   Solution: Off-policy. It always use greedy policy to do update.

   Rubrics: 2 points for answering off-policy; 2 more points for the explanation.

5. In Sarsa, what happens if actions are chosen greedily with respect to the action-value function rather than nearly greedily?

   Solution: The agent can get stuck assuming that some actions are worse than the one it is currently taking. It will not retry these other actions, and so it cannot learn that these other actions are actually better.

   Rubrics: 1 points for only writing down the Sarsa update formula.

6. Why do we use the action-value function instead of the state-value function in the generalized policy iteration framework for Monte Carlo and TD control?

Solution: To find out the greedy action based on the state-value function requires knowledge of state transition function and reward function of the underlying MDP, which is not available in the model-free setting.

Rubrics: 2 points for only mentioning model free; 4 points to explain that "the state-value function requires knowledge of the MDP".

7. What is the definition of one-step TD error δ_t using the state-value function? What is $\mathsf{E}[\delta_t | S_t = s]$ if δ_t uses the true state-value function v_π?

   Solution: One-step TD error is defined as $\delta_t = R_{t+1} + \gamma v(S_{t+1}) - v(S_t)$. If the true state-value function is used in δ_t, we have

$$
\begin{aligned}
\mathsf{E}[\delta_t | S_t = s] &= \mathsf{E}\left[R_{t+1} + \gamma v_\pi(S_{t+1}) - v_\pi(S_t) | S_t = s\right] \\
&\overset{(a)}{=} v_\pi(s) - v_\pi(s) \\
&= 0,
\end{aligned}
$$

   where (a) is due to Bellman expectation equation.

   Rubrics: 2 points to give the correct one-step TD error formula; 1 more point to correctly write down $\mathsf{E}[\delta_t | S_t = s]$ using the true state-value function; 1 more point for answering $\mathsf{E}[\delta_t | S_t = s] = 0$.

Problem 3 (**28 Points**)

Consider a simple random walk MDP shown in Fig. 1. From states s_1 and s_2, the agent can move to the left (a_0) or right (a_1). Rewards are given upon taking an action from a state. Taking an action from the goal state G earns a reward of $r = 1$ and the agent still stays in G. Other moves have zero reward ($r = 0$). Assume the discount factor $\gamma = 0.5$.

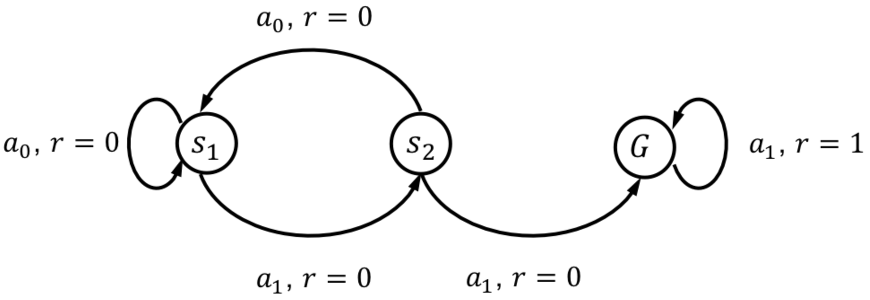

Figure 1: Simple Random Walk MDP

1. [6 Pts] Suppose the agent follows the random policy (i.e., take actions with equal probability) in states s_1 and s_2. Solve the Bellman expectation equation to find out the value function for all states (including the goal state G).

Solution: Let $v(s_1), v(s_2)$ and $v(G)$ denote the state-value function of states s_1, s_2 and G under the random policy, respectively. The Bellman expectation equation is as follows:

$$
\begin{aligned}
v(s_1) &= \gamma \left(\frac{1}{2} v(s_1) + \frac{1}{2} v(s_2) \right) = \frac{1}{4} v(s_1) + \frac{1}{4} v(s_2), \\
v(s_2) &= \gamma \left(\frac{1}{2} v(s_1) + \frac{1}{2} v(G) \right) = \frac{1}{4} v(s_1) + \frac{1}{4} v(G), \\
v(G) &= 1 + \gamma v(G) = 1 + \frac{1}{2} v(G).
\end{aligned}
$$

Solving the equations above, we have

$$v(s_1) = \frac{2}{11}, v(s_2) = \frac{6}{11}, v(G) = 2.$$

Rubrics: 3 points for writing down the correct Bellman equation either in matrix or non-matrix form; 1 more point for each correct value.

2. [6 Pts] Suppose the initial values of all states (including the goal state G) are 0. The agent follows the random policy (i.e., take actions with equal probability) in states s_1 and s_2. Find the values of all states (including the goal state G) in the first two steps of iterative policy evaluation using *synchronous* backup.

   Solution: Using Bellman expectation backup, the values of states s_1, s_2, G are $0, 0, 1$ after the first step, and $0, 0.25, 1.5$ after the second step.

   Rubrics: 1 point for each correct value.

3. [4 Pts] Following the results in the previous question, update the values of all states (including the goal state G) via one step of value iteration method using *synchronous* backup.

   Solution: Using Bellman optimality backup, after one step of value iteration method, the values of states s_1, s_2, G are $0.125, 0.75, 1.75$, respectively.

   Rubrics: 1 point for mentioning Bellman optimality backup or writing down the right formula; 1 more point for each correct value.

4. [4 Pts] Let v_1^* and v_2^* denote the optimal value of state s_1 and s_2, respectively. Write down the Bellman optimality equation for v_1^* and v_2^*.

   Solution: Since the value of state G is always 2, the Bellman optimality equation for v_1^* and v_2^* are as follows:

$$v_1^* = \gamma \max(v_1^*, v_2^*) = \frac{1}{2} \max(v_1^*, v_2^*), \tag{1}$$

$$v_2^* = \gamma \max(v_1^*, 2) = \frac{1}{2} \max(v_1^*, 2). \tag{2}$$

Rubrics: 2 points for the correct equation for v_1^* and v_2^*, respectively; 0.5 point deducted if writing down (without any argument) $v_1^* = 0.5v_2^*$ and $v_2^* = 1$, respectively.

5. [4 Pts] Based on your answer to the previous question, prove that $v_1^* < v_2^* < 2$.

   Proof: If $v_1^* \geq v_2^*$, from (1) we have

   $$v_1^* = \frac{1}{2}\max(v_1^*, v_2^*) = \frac{1}{2}v_1^*.$$

   Thus, $v_1^* = 0$, which implies $v_2^* = 1$ from (2). It is contradictory to the assumption that $v_1^* \geq v_2^*$. Therefore, we have $v_1^* < v_2^*$.

   In addition, if $v_2^* \geq 2$, from (2) we have $v_1^* \geq 2$ and $v_2^* = \frac{1}{2}v_1^* < v_1^*$, which is contradictory with $v_1^* < v_2^*$. Therefore, we have $v_2^* < 2$.

   Rubrics: 2 points to a correct proof of $v_1^* < v_2^*$ and $v_2^* < 2$, respetively.

6. [4 Pts] Find out the optimal value function of states s_1 and s_2 from the results of previous two questions. And write down the optimal policy of this random walk MDP.

   Solution: Since $v_1^* < v_2^* < 2$, from (1) and (2) we have

   $$v_1^* = \frac{1}{2}v_2^*,\ v_2^* = \frac{1}{2} \cdot 2 = 1,$$

   i.e., $v_1^* = 0.5$ and $v_2^* = 1$. The optimal policy from states s_1 and s_2 is to move right (take action a_1).

   Rubrics: 2 points for finding the correct optimal value function; 2 more point to correctly write down the optimal policy.

Problem 4 (28 Points)

Consider an *undiscounted* MDP with two non-terminal states A, B and a terminal state C. The transition function and reward function of the MDP are unknown. However, we have observed the following two episodes:

$$A, a_1, -3, A, a_1, +1, A, a_2, +3, C,$$
$$A, a_2, +1, B, a_3, +2, C,$$

where a_1, a_2, a_3 are actions, and the number after each action is an immediate reward. For example, $A, a_1, -3, A$ means that the agent took action a_1 from state A, received an immediate reward -3 and ended up in state A.

1. [4 Pts] Use first-visit Monte-Carlo evaluation, estimate the state-value function of $V(A)$ and $V(B)$.

   Solution:

$$
\begin{aligned}
V(A) &= \frac{(-3+1+3)+(1+2)}{2} = 2, \\
V(B) &= \frac{2}{1} = 2.
\end{aligned}
$$

   Rubrics: 2 points for $V(A)$ and $V(B)$, respectively.

2. [4 Pts] Use every-visit Monte-Carlo evaluation, estimate the state-value function of $V(A)$ and $V(B)$.

   Solution:

$$
\begin{aligned}
V(A) &= \frac{(-3+1+3)+(1+3)+3+(1+2)}{4} = 2.75, \\
V(B) &= \frac{2}{1} = 2.
\end{aligned}
$$

   Rubrics: 2 points for $V(A)$ and $V(B)$, respectively.

3. [6 Pts] Using a learning rate of $\alpha = 0.1$, and assuming initial state values of 0, what updates to $V(A)$ does *on-line* TD(0) method make after the first episode?

   Solution: The update to $V(A)$ at each step of the first episode using on-line TD(0) is:

$$
\begin{aligned}
V(A) &= V(A) + \alpha(-3 + V(A) - V(A)) = 0 + 0.1 \times (-3) = -0.3, \\
V(A) &= V(A) + \alpha(1 + V(A) - V(A)) = -0.3 + 0.1 \times 1 = -0.2, \\
V(A) &= V(A) + \alpha(3 + V(C) - V(A)) = -0.2 + 0.1 \times 3.2 = 0.12.
\end{aligned}
$$

   Rubrics: 2 points for each update to $V(A)$.

4. [8 Pts] Draw a diagram of an MDP and policy π that can best fit these two episodes (i.e., the maximum likelihood Markov model). Write down the transition probability $P^a_{s,s'}$, the reward function R^a_s of the MDP you draw, and your estimated policy $\pi(a|s)$.

   Solution: The maximum likelihood Markov model is shown in the following Fig. 2.

   The transition probabilities and reward function are as follows (also shown in Fig. 2):

$$
\begin{aligned}
&P^{a_1}_{A,A} = 1,\ P^{a_2}_{A,B} = \frac{1}{2} = 0.5,\ P^{a_2}_{A,C} = \frac{1}{2} = 0.5,\ P^{a_3}_{B,C} = 1, \\
&R^{a_1}_A = \frac{-3+1}{2} = -1,\ R^{a_2}_A = \frac{3+1}{2} = 2,\ R^{a_3}_B = 2.
\end{aligned}
$$

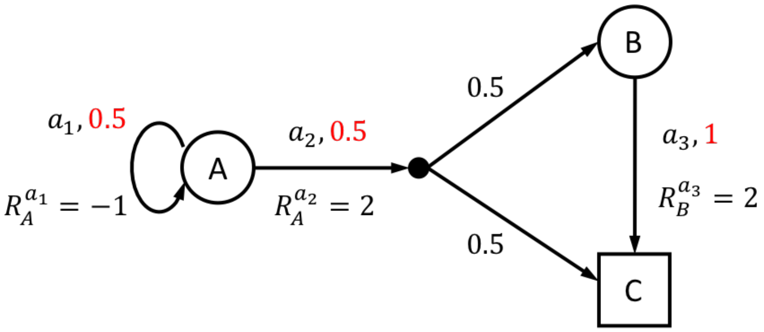

Figure 2: Maximum Likelihood Markov Model

And the estimated policy (marked in red in Fig. 2) is

$$\pi(a_1|A) = \frac{2}{4} = 0.5,\ \pi(a_2|A) = \frac{2}{4} = 0.5, \pi(a_3|B) = 1.$$

Rubrics: 3 points for the MDP diagram (no need to include all numbers, but states/actions should be the same as in Fig. 2; 0.5 more point each for a correct value of transition probability, reward function or the estimated policy.

5. [3 Pts] Based on your results in the previous question, solve Bellman equation to find out the state-value function of $V_\pi(A)$ and $V_\pi(B)$.

   Solution: The Bellman expectation equation for the MDP in Fig. 2 is as follows:

$$\begin{aligned} &V_\pi(A) = 0.5(-1 + V_\pi(A)) + 0.5(2 + 0.5V_\pi(B)), \\ &V_\pi(B) = 2. \end{aligned}$$

   We have $V_\pi(A) = 2, V_\pi(B) = 2$.

   Rubrics: 1 point for correct $V_\pi(B)$; 2 more points for correct $V_\pi(A)$.

6. [3 Pts] What value function would batch TD(0) find, i.e., if TD(0) was applied repeatedly to these two episodes?

   Solution: TD(0) will find the solution to the MDP in Fig. 2, which is $V_\pi(A) = 2, V_\pi(B) = 2$.

   Rubrics: 2 points for mentioning "TD(0) will find the solution to the MDP in Fig. 2"; 1 more point for correct values of $V_\pi(A)$ and $V_\pi(B)$.