Midterm Exam

ELEN E6885: Introduction to Reinforcement Learning

Problem 1 (**20 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
9. In partially observable Markov decision process (POMDP), it is likely that an agent cannot identify its current state. So the best choice is to maintain a probability distribution over the states and actions. and then updates this probability distribution based on its real-time observations.Solution: False
10. Almost all off-policy methods utilize importance sampling, a general technique for estimating expected values under one distribution given samples from another. Solution: True

Problem 2 (24 Points, 4 Points each)

Short-answer questions.

1. Is the MDP framework adequate to usefully represent all goal-directed learning tasks? Can you think of any clear exceptions? The main thing about the MDP is Markov property. There are tasks where this does not hold. For instance, in Poker, the previous states will determine what is in the deck and what is not. This does not obey Markov property.

   Rubrics: 2 point for the first question. 2 points for the example.

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. For Q-learning to converge we need to correctly manage the exploration vs. exploitation tradeoff. What property needs to be hold for the exploration strategy?

   Solution: In the limit, every action needs to be tried sufficiently often in every possible state. This can be guaranteed with a sufficiently permissive exploration strategy.

   Rubrics: 2 points for "sufficiently often" or equivalent. 2 points for "every state"

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".

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 $\gamma = 0.5$.

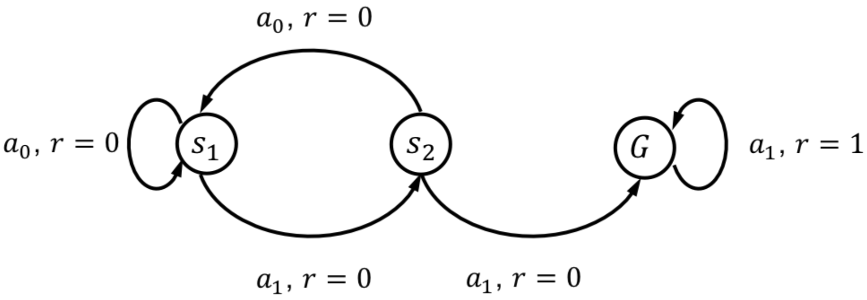

Figure 1: Simple Random Walk MDP

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:

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

   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)

For this problem, we give full marks to the solution using 0.9 or 1

See Fig.2 where the states are grid squares, identified by their row and column numbers (row first). The agent always starts in state (1.1), marked with the letter S. There are two terminal states $(2, 3)$ with reward $+5$ and $(1, 3)$ with reward -5. Rewards are 0 in non-terminal states. (The reward for a state is received as the agent moves into the state.) The transition function is such that the intended agent movement (North, South, West, or East) happens with probability 0.8. With probability 0.1 each, the agent ends up in one of the states perpendicular to the intended direction. If a collision with a wall happens, the agent stays in the same state.

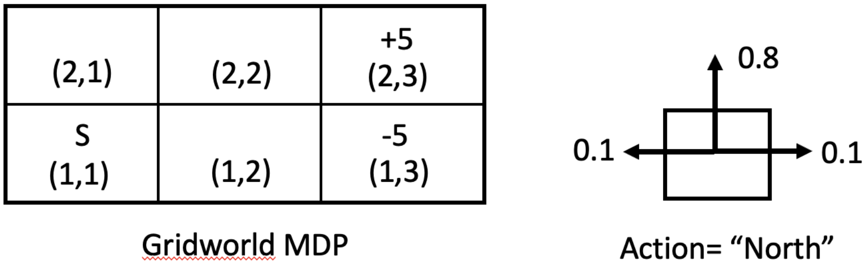

Figure 2: Gridworld MDP

1. [4 Pts] Draw the optimal policy for this grid, i.e., $(1, 1), (1, 2), (2, 1), (2, 2)$.

   Solution: (1,1): Up; (1,2): Left; (2,1): Right; (2,2):Right

   Rubrics: 1 point each

2. [8 Pts] Suppose the agent knows the transition probabilities as shown in Fig.2. Compute the first two rounds of value iteration updates for each state (state “$(1, 1), (1, 2), (2, 1), (2, 2)$”),

with a discount of 0.9. (Assume V_0 is 0 everywhere and compute V_i for time $i = 1, 2$). The Bellman backup equation for value iteration is

$$V_{i+1}(s) = \max_a (\sum_{s'} T(s, a, s')(R(s, a, s') + \gamma V_i(s')))$$

Solution: $V_1(1,1) = V_1(1,2) = V_1(2,1) = 0$, $V_1(2,2) = 0.8 \times 5 = 4$
$V_2(1,1) = 0$; $V_2(1,2) = 0.9 \times 0.8 \times 4 + 0.1 \times (-5) = 2.38$; $V_2(2,1) = 0.8 \times 0.9 \times 4 = 2.88$;
~~$V_2(2,2) = 0.8 \times 5 = 4$.~~ **Should be -> V2(2,2) = 0.8 * 5 + 0.1 * 0.9 * 4 + 0.1 * 0.9 * 0 = 4.36**

Rubrics: 1 point each

3. [4 Pts] Suppose the agent does not know the transition probabilities. What does it need to be able do (or have available) in order to learn the optimal policy?

   Solution: The agent must be able to explore the world by taking actions and observing the effects.

4. [4 Pts] The agent starts with the policy that always chooses to go right, and executes the following three trials: 1) $(1,1)$–$(1,2)$–$(1,3)$, 2) $(1,1)$–$(1,2)$–$(2,2)$–$(2,3)$, and 3) $(1,1)$–$(2,1)$–$(2,2)$–$(2,3)$. What are the Monte Carlo estimates for states $(1,1)$ and $(2,2)$, given these traces?

   Solution: To compute the estimates, average the rewards received in the trajectories that went through the indicates states.

$$V(1,1) = \frac{-5 + 5 + 5}{3} = 1.666$$

$$V(2,2) = \frac{(5 + 5)}{2} = 5$$

   Rubrics: 2 point each

5. [8 Pts]Using a learning rate of $\alpha = 0.1$ and assuming initial values of 0, what updates does the TD-learning agent make after trials 1 and 2, above? The general TD-learning update is $V(s) = V(s) + \alpha(r + \gamma V(s') - V(s))$.

   Solution:

   After trial 1, the updates will be:

$$V(1,1) = V(2,1) = V(2,2) = 0$$

$$V(1,2) = 0 + 0.1(-5 + 0.9 \times 0 - 0) = -0.5$$

   After trial 2, the updates will be:

$$V(1,1) = 0 + 0.1(0 + 0.9 \times -0.5 - 0) = -0.045$$

$$V(1,2) = -0.5 + 0.1 * (0 + 0.9 \times 0 + 0.5) = -0.45$$
$$V(2,2) = 0 + 0.1(5 + 0.9 \times 0 - 0) = 0.5$$
$$V(2,1) = 0$$

Rubrics: 1 point each