Midterm Exam: Examples of Questions

An Open-Notes Exam

CS4771 Reinforcement Learning (Spring 2026)

Other important notes:

- Write your answer in the **answer boxes**. Anything written outside the answer boxes or ~~crossed out~~ will not be graded.
- Besides the answer box, there are several questions with a block of **Detailed calculation**. It is NOT necessary to provide detailed calculation. However, that could earn you partial credit if your answer is incorrect.
- **Fractions** need NOT be simplified or converted to decimals in your answer.
- The number of correct answers in **multiple choice questions** can range from 0 to 5. They will be graded as 5 independent true-or-false questions. If you think none of the choices are correct, put "None" in the answer box; leaving it blank will result in zero points.

1. In this problem, we use Markov decision process to model the operation of a machine.

The machine can be in one of two states: B (Bad condition), G (Good condition). In each state, there are two possible actions: N (Do nothing) and R (Repair). The machine operates indefinitely with a discount factor $\gamma = 0.9$. The rewards and transition probabilities are defined in the tables below.

s	a	$R(s,a)$
B	N	-5
B	R	-13
G	N	+10
G	R	+8

Rewards

s	a	s'	$P(s' \mid s,a)$
B	N	B	1.0
B	R	G	1.0
G	N	G	0.8
G	N	B	0.2
G	R	G	1.0

Transition Probabilities

Rationale behind the reward and transition functions are as follows:

- **Do nothing** in **Bad condition**: Incur a loss of -5 due to lost productivity and remain in bad condition.
- **Repair** in **Bad condition**: Incur a total cost of -13 (loss of productivity and repair cost) and transition to good condition.

- **Do nothing** in **Good condition**: Earn $+10$ from normal production, with a 20% chance of transitioning to bad condition.
- **Repair** in **Good condition**: Pay a maintenance cost of -2, resulting in a net reward of $+8$, and ensure the machine stays in good condition.

Suppose we start from $V_0(s) = Q_0(s,a) = 0$ for all s, a. Let $V_k(s)$ and $Q_k(s,a)$ be the value functions after performing value iterations for k times. Let $V^\star(s) = \lim_{k\to\infty} V_k(s)$ and $Q^\star(s,a) = \lim_{k\to\infty} Q_k(s,a)$.

1.1. Write the Bellman equations (the relations that should be satisfied for $V^\star$ and $Q^\star$) for this MDP, using the specific numbers provided in the tables.

Answer:

$$
\begin{aligned}
Q^\star(B,N) &= -5 + 0.9V^\star(B) \\
Q^\star(B,R) &= -13 + 0.9V^\star(G) \\
V^\star(B) &= \max\{Q^\star(B,N), Q^\star(B,R)\} \\
Q^\star(G,N) &= 10 + 0.9\left(0.2V^\star(B) + 0.8V^\star(G)\right) \\
Q^\star(G,R) &= 8 + 0.9V^\star(G) \\
V^\star(G) &= \max\{Q^\star(G,N), Q^\star(G,R)\}
\end{aligned}
$$

1.2. Starting with $V_0(B) = 0$ and $V_0(G) = 0$, perform two iterations of value iteration. Compute $V_1(B)$, $V_1(G)$, $V_2(B)$, and $V_2(G)$.

$V_1(B) =$ -5 $V_1(G) =$ 10 $V_2(B) =$ -4 $V_2(G) =$ 17

$V_1(B) = \max\{-5 + 0.9V_0(B), -13 + 0.9V_0(G)\} = -5.$
$V_1(G) = \max\{10 + 0.9(0.2V_0(B) + 0.8V_0(G)), 8 + 0.9V_0(G)\} = 10.$
$V_2(B) = \max\{-5 + 0.9V_1(B), -13 + 0.9V_1(G)\} = \max\{-5 + 0.9 \times (-5), -13 + 0.9 \times 10\} = -4.$
$V_2(G) = \max\{10 + 0.9(0.2V_1(B) + 0.8V_1(G)), 8 + 0.9V_1(G)\} = \max\{10 + 0.9(0.2 \times -5 + 0.8 \times 10), 8 + 0.9 \times 10\} = 17.$

1.3. Determine the optimal policy $\pi^\star$ and the optimal discounted value function $V^\star$.

$\pi^\star(B) =$ R $\pi^\star(G) =$ R $V^\star(B) =$ 59 $V^\star(G) =$ 80

One can perform several iterations of value iteration. From iteration 2, the better action is Repair on every state. Hence, let's guess $\pi^\star(B) = \pi^\star(G) =$ R and use it to solve $V^\star$. The Bellman equation with $\pi^\star(B) = \pi^\star(G) =$ R will be

$$
\begin{aligned}
V^\star(B) &= -13 + 0.9V^\star(G) \\
V^\star(G) &= 8 + 0.9V^\star(G).
\end{aligned}
$$

Solving them gives $V^\star(G) = 80$ and $V^\star(B) = 59$. We can verify that our guess of $\pi^\star$ is correct because $-5 + 0.9V^\star(B) < -13 + 0.9V^\star(G)$ and $10 + 0.9\left(0.2V^\star(B) + 0.8V^\star(G)\right) < 8 + 0.9V^\star(G)$.

2. Which of the following statements about contextual bandits are true?

A. In contextual bandits, the action chosen by the learner may affect future contexts.

B. Unlike supervised learning, contextual bandits suffer from "bandit feedback," meaning the learner only observes the reward for the chosen action and not the rewards of the unchosen actions.

C. In policy-based methods, we construct unbiased reward estimators to encourage exploration.

D. With ϵ-greedy, we usually use decreasing ϵ over time to balance exploitation and exploration.

E. If the learner has a reward estimation $\hat{R}$ such that $|\hat{R}(x,a) - R(x,a)| \leq \Delta$ for all x, a (R is the true underlying reward function), then the greedy-policy $\pi(x) = \text{argmax}_a \hat{R}(x,a)$ guarantees $\max_a R(x,a) - R(x,\pi(x)) \leq 2\Delta$.

Answer: BDE

3. Recall the UCB algorithm for multi-armed bandits:

Algorithm 1 UCB

Parameters: fixed constant $c = 2$.

for $t = 1, 2, \ldots, T$ **do**

Select arm

$$a_t = \operatorname*{argmax}_a \left(\hat{R}_t(a) + c\sqrt{\frac{1}{N_t(a)}} \right) \tag{1}$$

where $\hat{R}_t(a)$ is the empirical mean of arm a up to time $t-1$, and $N_t(a)$ is the number of draws to arm a up to time $t-1$. In case of a tie, the algorithm selects the arm with the smallest index.

Suppose at time $t = 30$, the learner has collected the following statistics for three arms ($A = 3$):

Arm (a)	**Empirical Mean** ($\hat{R}_{30}(a)$)	**Number of Pulls** ($N_{30}(a)$)
1	0.85	16
2	0.70	9
3	0.40	4

Which arm a_{30} will the algorithm select? Justify your answer.

Answer:

Arm 3.

Calculate the objective in Eq. (1):

$$\hat{R}_{30}(1) + c\sqrt{\frac{1}{N_{30}(1)}} = 0.85 + 2\sqrt{\frac{1}{16}} = 1.35$$

$$\hat{R}_{30}(2) + c\sqrt{\frac{1}{N_{30}(2)}} = 0.70 + 2\sqrt{\frac{1}{9}} = 1.367$$

$$\hat{R}_{30}(3) + c\sqrt{\frac{1}{N_{30}(3)}} = 0.40 + 2\sqrt{\frac{1}{4}} = 1.40$$

The maximizer is arm 3.

4. Consider the KL-regularized policy update algorithm for multi-armed bandits:

$$\pi_{t+1} = \underset{\pi}{\operatorname{argmax}} \left\{ \langle \pi, r_t \rangle - \beta \mathrm{KL}(\pi, \pi_t) \right\}.$$

Here, $r_t(a) = R(a) + w_t(a)$ where $R(a)$ being the true expected reward of arm a and $w_t(a)$ is a zero-mean noise. Explain the effects of setting β too large or too small, respectively.

Answer:

Setting β too large makes the KL term dominate, forcing π_{t+1} to stay close to π_t. The updates become very small, so learning is slow and overly conservative.
Setting β too small weakens the regularization, so the update mainly maximizes $\langle \pi, r_t \rangle$. Because r_t contains noise, the policy may overreact those noises, leading to unstable behavior.