CS 747 (Spring 2025) Mid-semester Examination

6.00 p.m. – 8.00 p.m., February 27, 2025, LA 001 and LA 002

Name: ______________________________ Roll number: _______________

Note. There are three questions in this test. Provide your answer to each question in the space following the question (and *before* the next question if one exists). You can use any blank space in the paper for rough work by drawing a line (either vertical or horizontal), writing "Rough work" on one side of it, and using the demarcated space for rough work. In addition, the remainder of this page as well as a separate page at the end have been provided for rough work.

Rough work ↓

Question 1. For this question, consider the set of bandit instances $\mathcal{I}$ with $n \geq 2$ arms, each arm yielding Bernoulli rewards with means in $[0, 1]$. For $\epsilon \in (0, 1)$, an algorithm is said to be ϵ-fair if the algorithm necessarily allots each arm at least $\left\lfloor \frac{\epsilon t}{n} \right\rfloor$ pulls out of the total number of pulls t; this property must hold for all $t \geq 1$. An alternative statement of algorithm L being ϵ-fair is as follows. For $t \geq 1$, let $m(t)$ denote the number of pulls given by L to an arm that it has pulled the fewest among all arms in the first t pulls. L is ϵ-fair if and only if $m(t) \geq \left\lfloor \frac{\epsilon t}{n} \right\rfloor$ for all $t \geq 1$. For example, suppose $\epsilon = 0.26, n = 5$. Then we require $m(20) \geq 1, m(39) \geq 2, m(58) \geq 3$, and so on.

For algorithm L, bandit instance $I \in \mathcal{I}$, and horizon $T \geq 1$, let $\text{rew}(L, I, T)$ denote the expected cumulative reward of L on I over T pulls. Let $\mathcal{L}_\epsilon$ be the set of all ϵ-fair algorithms. For $\epsilon \in (0, 1)$, bandit instance $I \in \mathcal{I}$, and horizon $T \geq 1$, let $\text{rew}^\star(I, T)$ denote the maximum expected cumulative reward achievable by any ϵ-fair algorithm on I over T pulls: that is,

$$\text{rew}^\star(\epsilon, I, T) = \max_{L \in \mathcal{L}_\epsilon} \text{rew}(L, I, T).$$

Consider the following statement S.

S: For $\epsilon \in (0, 1)$, there exists an algorithm $L^\star_\epsilon \in \mathcal{L}_\epsilon$ such that for every $I \in \mathcal{I}$,

$$\lim_{T \to \infty} \frac{\text{rew}(L^\star_\epsilon, I, T)}{\text{rew}^\star(\epsilon, I, T)} = 1.$$

Informally, S claims that for every fixed ϵ, there exists an ϵ-fair algorithm that "succeeds" (in the sense of asymptotic optimality) on all bandit instances. Is S true or false? Provide a proof to justify your answer. [5 marks]

Answer 1.

There is a minor mathematical error in the framing of the question: the cumulative reward of any algorithm is exactly 0 if the input bandit instance I has 0 as mean for each arm. Consequently the ratio referred to in S is not well-defined. The error is regretted; credit to Yashwanth Reddy for catching it. We proceed to our answer by assuming that $\mathcal{I}$ *does not* contain the all-zero-mean bandit instance: that is, $\mathcal{I} \overset{\text{def}}{=} [0, 1]^n \setminus 0^n$.

Let us consider an arbitrary bandit instance $I \in \mathcal{I}$, in which the arms' means are

$$p_1 = p_2 = p_3 = \cdots = p_m > p_{m+1} \geq p_{m+2} \geq p_{m+3} \geq \ldots p_n$$

for some $m \geq 1$. In other words, there are m optimal arms and $n - m$ suboptimal arms. Let the mean of arm i be p_i, and let $p^\star$ be the mean of an optimal arm.

First we characterise $\text{rew}^\star(\epsilon, I, T)$. If $m = n$ (that is, all means are $p^\star$), then clearly $\text{rew}^\star(\epsilon, I, T) = Tp^\star$. If $m < n$, we observe that each suboptimal arm must have been pulled at least $\left\lfloor \frac{\epsilon T}{n} \right\rfloor$ times. Even if an optimal arm is pulled for all time steps other than the suboptimal pulls required to meet the ϵ-fairness requirement, we have

$$\text{rew}^\star(\epsilon, I, T) \leq \sum_{i=m+1}^{n} \left\lfloor \frac{\epsilon T}{n} \right\rfloor p_i + (T - (n - m) \left\lfloor \frac{\epsilon T}{n} \right\rfloor) p^\star. \tag{1}$$

Now, we describe an ϵ-fair algorithm $L^\star_\epsilon$, which asymptotically matches (as a fraction) $\mathrm{rew}^\star(\epsilon, I, T)$. $L^\star_\epsilon$ functions as follows for each $t = 1, 2, \ldots$:

- If $m(t) = \lfloor \frac{\epsilon T}{n} \rfloor$, then pull any arm that has exactly $\lfloor \frac{\epsilon T}{n} \rfloor$ pulls.//Call these "fair pulls".
- If $m(t) > \lfloor \frac{\epsilon T}{n} \rfloor$, then pull any arm with the highest empirical average.//Call these "exploit pulls".

At each time step t, each arm has been pulled at least $\lfloor \frac{\epsilon t}{n} \rfloor$ times. We use this lower bound to argue that the empirical mean of each arm will be "close enough" to its true mean with sufficiently high probability, so that the probability of an exploit pull going to a suboptimal arm will be small. Define $\Delta = \frac{p_m - p_{m+1}}{2}$. Now consider any arm a at time step t. The arm has been pulled $\lfloor \frac{\epsilon t}{n} \rfloor$ or $\lfloor \frac{\epsilon t}{n} \rfloor + 1$ or $\lfloor \frac{\epsilon t}{n} \rfloor + 2$ or $\ldots$ or t times. By Hoeffding's inequality and a union bound, the probability that its empirical mean diverges from the true mean by Δ or more (in any one direction) is at most

$$\sum_{u=\lfloor \frac{\epsilon t}{n} \rfloor}^{t} \exp(-2u\Delta^2) \leq \sum_{u=\lfloor \frac{\epsilon t}{n} \rfloor}^{\infty} \exp(-2u\Delta^2) \leq \alpha \exp(-\beta t)$$

for some positive quantities α and β that could depend on n, ϵ, and Δ (but not on t). Consequently, the probability that there exists an arm that deviates (in any one direction) by more than Δ at time step t is at most $n\alpha \exp(-\beta t)$. Note that if every suboptimal arm has its empirical mean below $\frac{p_m + p_{m+1}}{2}$ and every optimal arm has its mean above $\frac{p_m + p_{m+1}}{2}$, then pulling an empirical optimal arm is equivalent to pulling a true optimal arm. Thus, the expected number of exploit pulls of suboptimal arms up to T steps is at most

$$\sum_{t=1}^{T} n\alpha \exp(-\beta t) \leq \sum_{t=1}^{\infty} n\alpha \exp(-\beta t) \leq \gamma$$

for some positive quantity γ that could depend on n, ϵ, and Δ (but not on T).

Clearly the number of fair pulls given to any arm by $L^\star_\epsilon$ is at most $\lfloor \frac{\epsilon T}{n} \rfloor + 1$, and so the expected number of pulls of optimal arms is at least $T - (n-m)(\lfloor \frac{\epsilon T}{n} \rfloor + 1) - \gamma$. Hence we have

$$\mathrm{rew}(L^\star_\epsilon, I, T) \geq \sum_{i=m+1}^{n} \left\lfloor \frac{\epsilon T}{n} \right\rfloor p_i + (T - (n-m)(\left\lfloor \frac{\epsilon T}{n} \right\rfloor + 1) - \gamma) p^\star. \tag{2}$$

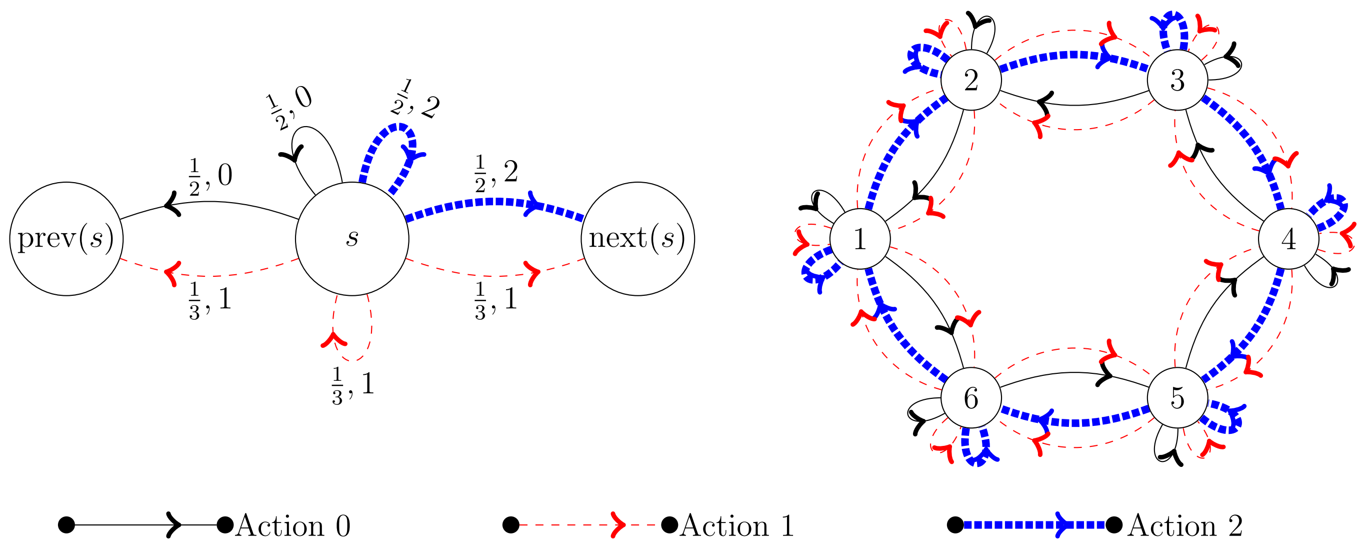
From (1) and (2), we get

$$\lim_{T \to \infty} \frac{\mathrm{rew}(L^\star_\epsilon, I, T)}{\mathrm{rew}^\star(\epsilon, I, T)} = 1.$$

Question 2. This question is about an MDP with a set of $n \geq 2$ states $S = \{1, 2, \ldots, n\}$, and a set of three actions $A = \{0, 1, 2\}$. There is cyclic dependence among the states in the MDP, in the sense that each state $s \in S$ has a "previous" state $\mathrm{prev}(s)$ and a "next" state $\mathrm{next}(s)$, defined by

$$\mathrm{prev}(s) = \begin{cases} n & \text{if } s = 1, \\ s-1 & \text{if } 2 \leq s \leq n; \end{cases} \qquad \mathrm{next}(s) = \begin{cases} s+1 & \text{if } 1 \leq s \leq n-1, \\ 1 & \text{if } s = n. \end{cases}$$

From each state s, the only possible transitions are to s itself, to $\mathrm{prev}(s)$, or to $\mathrm{next}(s)$. In particular, action 0 takes the agent from state s to either s or $\mathrm{prev}(s)$ with equal probability; action 1 takes the agent from state s to s, $\mathrm{prev}(s)$, or $\mathrm{next}(s)$ with equal probability; while action 2 takes the agent from state s to s or $\mathrm{next}(s)$ with equal probability. The reward for any transition is fully determined by the action: the reward is 0 from action 0; 1 from action 1; and 2 from action 2. The discount factor is $\gamma = \frac{4}{5}$. Below, on the left, is a diagram showing only the transitions emanating from state s, annotated with "probability, reward". On the right is the state-transition diagram for $n = 6$, with annotations not shown so as to avoid clutter.



Work out the optimal value function $V^\star$ and an optimal policy $\pi^\star$ for the given MDP; observe that both quantities are n-dimensional. Since there are 3^n policies (and n is a variable), you cannot possibly evaluate all policies. Begin with your intuition to propose the solution; thereafter use the structure of the MDP to provide formal justification that what you claim to be $V^\star$ is indeed the optimal value function and what you claim to be $\pi^\star$ is indeed an optimal policy. You can reuse any result that has been proven in class. [5 marks]

Answer 2.

There are multiple ways to reason about acting optimally in this MDP.

Approach 1. It strikes us that there is a symmetry with respect to the states. On close observation, we notice that action 2, which gives a reward at 2, is available from each state. Since 2 is the largest possible single-step reward possible, the maximum possible value from any starting state has to be at most

$$2 + \gamma \cdot 2 + \gamma^2 \cdot 2 + \cdots = \frac{2}{1-\gamma} = 10.$$

The policy that takes 2 at each time step indeed achieves this long-term reward from every state, and hence must be optimal.

Approach 2. Let π be any policy. Now consider the policy π' that is obtained by "shifting" the actions of π by one state: that is, for $s \in S$,

$$\pi'(s) \overset{\text{def}}{=} \pi(\text{prev}(s)).$$

We observe from the structure of the MDP that this means $V^{\pi'}(s) = V^{\pi}(\text{prev}(s))$ for $s \in S$. We can again shift π' to obtain π'', which would satisfy $V^{\pi''}(s) = V^{\pi}(\text{prev}(\text{prev}(s)))$ for $s \in S$. The implication of this property of shifting is that if a value of v is possible at some state, it is also possible at every other state. In turn this means that all optimal values must be the same. Let us call this common optimal value $v^\star$. By the Bellman optimality equations, we have for $s \in S$,

$$\begin{aligned} v^\star &= \max\left\{0 + \frac{1}{2}(\gamma v^\star + \gamma v^\star), 1 + \frac{1}{3}(\gamma v^\star + \gamma v^\star + \gamma v^\star), 2 + \frac{1}{2}(\gamma v^\star + \gamma v^\star)\right\} \\ &= \max\left\{0 + \gamma v^\star, 1 + \gamma v^\star, 2 + \gamma v^\star\right\} \\ &= 2 + \gamma v^\star. \end{aligned}$$

Again, this working yields $v^\star = \frac{2}{1-\gamma} = 10$; while the action delivering the "max" on the RHS is 2 for all states.

Approach 3. If the "all 2" policy $\pi^\star$ is claimed to be optimal and its value (10 at each state) determined by solving its Bellman equations, a simple way to prove optimity is to show that actions 1 and 2 have strictly lower Q-values at each state. Notice that for $s \in S$,

$$Q^{\pi^\star}(s, 0) = 0 + \gamma \cdot 10 = 8;$$

$$Q^{\pi^\star}(s, 1) = 1 + \gamma \cdot 10 = 9.$$

Question 3. This question is about Tic-Tac-Toe, the popular game played on a 3×3 grid. To begin the game, the first player places an "X" in any of the cells; then the second player places an "O" in any other cell. The players alternate placing X's and O's on unoccupied cells until termination. The game terminates either when (1) one of the players has an entire row, column, or diagonal filled with their symbol—in which case that player is the winner, or (2) all 9 cells are filled up such that every row, column, and diagonal contains both players' symbols—in which case the outcome is a draw. Figure (a) below shows a terminal grid with the second player having won, while Figure (b) shows a possible grid for a draw.

Your aim is to help the first player maximise their probability of winning, when it is known that at each step the second player always selects an unoccupied cell *uniformly at random* to place their O. For example, if the grid in Figure (c) below is the result after three X's and two O's have been placed, then the four grids following it (figures (d), (e), (f), and (g)) are each the next grid with probability $\frac{1}{4}$.

X		X
O	O	O
X		X

(a)

X	X	O
O	O	X
X	O	X

(b)

O		X
	O	X
X		

(c)

O	O	X
	O	X
X		

(d)

O		X
O	O	X
X		

(e)

O		X
	O	X
X	O	

(f)

O		X
	O	X
X		O

(g)

Assume that you have access to an MDP solver. You need to design its input: an MDP such that an optimal policy for the MDP (which the solver will provide) can guide the first player in their action selection. Specify the various elements of your MDP, and describe how an optimal policy for it can be used by the first player to maximise their win probability (the first player does not distinguish losses from draws). You do not have to code up the rules of Tic-Tac-Toe mathematically; instead, use precise natural language to specify details about states, actions, transitions, rewards, and values. [5 marks]

Answer 3.

Since we are helping only the first player with their decisions, the set of (non-terminal or decision-making) states are board configurations in which the first player has to act. These would only include grids containing an equal number of X's and O's, and such that no row, column, or diagonal is completely occupied with the same symbol (among X and O). Terminal states (where no decisions have to be made) would include grids with all 9 cells filled, or an entire row, column, or diagonal occupied with the same symbol. A terminal state in which the first player has won would have one more X than O's, while a terminal state in which the second player has won would have an equal number of X's and O's. Incidentally, Figure (a) in the question has a typo: it must have three (rather than four) X's. Credit to Sai Deepthika for catching the typo. Apart from terminal states named "win", "draw", and "loss", we also define an additional terminal state named "illegal", explained below.

We associate an action with each cell, giving a total of 9 actions. Of course, some actions would be "illegal", but this can be handled using the transition- and reward functions, described next.

Suppose the first player is in state s. An illegal action a (placing an X on a filled cell) results in a terminal state labeled "illegal", and all transitions into "illegal" are given a reward of -1. On the other hand, for each legal action a, we consider the grid reached G_1 reached. G_1 must either be a position from which the second player must act, or it must be terminal. If G_1 is a terminal grid, then our MDP correspondingly transitions into a terminal state that is either "draw" or "win", depending on G_1. If G_1 is not terminal, then we know that the opponent places an O uniformly at random among the unoccupied cells of G_1. We construct the grids $G_{21}, G_{22}, \ldots, G_{2m}$ reached by the second player's possible actions, and construct a state out of each. If any of these is terminal, it is labelled "loss" (a move from the second player cannot result in a draw or a loss for the second player). A transition probability of $\frac{1}{m}$ is given to each triple (s, a, G_{2i}) for $i \in \{1, 2, \ldots, m\}$.

Rewards are 0 for all transitions other than those for illegal actions (which are -1) and when the next state reached is terminal. In the latter case, the reward is 1 if the terminal state is "win", and 0 otherwise. No discounting is used; that is, the discount factor is 1.

With the construction given above, if starting from the empty grid, the first player will necessarily reach a terminal state after at most 5 actions. For any given policy, the expected long-term reward will be exactly the probability of winning. Consequently, an optimal policy will provide the first player a choice of actions from each state such that it maximises the player's probability of winning.

Rough work ↓