CS 747 (Autumn 2020): Weekly Quizzes

Instructor: Shivaram Kalyanakrishnan

December 3, 2020

Note. Provide justifications/calculations/steps along with each answer to illustrate how you arrived at the answer. You will not receive credit for giving an answer without sufficient explanation.

Submission. Write down your answer by hand, then scan and upload to Moodle. Write clearly and legibly. Be sure to mention your roll number.

Week 12

Question. In general, Experience Replay is used on large tasks, when function approximation is needed. Nonetheless, in this question, we consider applying it on a finite MDP with a small number of states and actions—say 10 states and 2 actions. In this case, $\hat{Q}$ can be represented as a table.

Suppose we have gathered a data set D with $L = 10,000$ transitions of the form (s, a, r, s'). In the form described in the week's lecture, Experience Replay requires drawing a large number of random samples from D and performing "Q-learning"-type updates. Say this number of updates is $M = 10^6$—implying roughly 100 visits to each sample in D. Assume that the combination of M and a small learning rate α ensures that $\hat{Q}$ converges (in a practical sense) at the end of the Experience Replay phase. Denote the converged value $\hat{Q}_{\text{output}}$.

Can you think of a faster way to compute $\hat{Q}_{\text{output}}$ from D—taking time in the order of $\theta(L)$ rather than $\theta(M)$? Use the fact that the MDP is finite and small. No need for pseudocode or precise calculations; a high-level sketch will do.

Solution. The answer is essentially the model-based algorithm presented in Week 8 and in Class Note 3. We build an empirical model, associating with each state-action-state pair a transition probability and a reward, both estimated empirically from D. The time taken to populate this model is linear in L. Thereafter we solve for the optimal action value function on the model. This function is going to be the same as $\hat{Q}_{\text{output}}$ (recall that we had established a similar equivalence in the context of prediction with Batch TD(0)). The time taken to plan is expected to be negligible with only 10 states and 2 actions, leading to a faster computation in aggregate compared to M Experience Replay updates.

On realistic tasks that require function approximation, it is not always possible to learn/represent sufficiently good models. It is also not straightforward (as it is in the tabular case) to plan with generalisation over a a large state space.

Week 11

Question. In this week's lecture, we observed Tsitsiklis and Van Roy's counterexample. We claimed that its demonstration crucially depended on the conjunction of three factors: off-policy updating, bootstrapping, and generalisation. In this question, we consider the effect of removing one of these factors: specifically we replace the off-policy update with an on-policy update. Refer to the MDP in the counterexample on Slide 9 of the lecture. Suppose that episodes always start at state s_1. Since there is a deterministic transition to s_2, the number of time steps per episode in s_1 is exactly $T(s_1) = 1$. Similarly, what is $T(s_2)$, the *expected* number of time steps per episode in s_2? Naturally $T(s_2)$ must depend on ϵ; assume $\epsilon \in (0, 1)$. We use the same linear architecture as described in the lecture.

For $k \geq 0$, the new update rule we propose is

$$w_{k+1} \leftarrow \operatorname*{argmin}_{w \in \mathbb{R}} \sum_{s} T(s) \left(\mathbb{E}_{\pi}[r + \gamma \hat{V}(w_k, x(s'))] - \hat{V}(w, x(s)) \right)^2.$$

Show that whatever be the initialisation $w_0 \in \mathbb{R}$, we shall achieve $\lim_{k \to \infty} w_k = 0$.

Solution. The expected number of time steps per episode spent at s_2 is

$$T(s_2) = \epsilon(1) + (1-\epsilon)\epsilon(2) + (1-\epsilon)^2\epsilon(3) + \cdots = \frac{1}{\epsilon}.$$

Hence, we have

$$\begin{aligned}
w_{k+1} &= \operatorname*{argmin}_{w \in \mathbb{R}} \left((2\gamma w_k - w)^2 + \frac{1}{\epsilon}(2\gamma w_k(1-\epsilon) - 2w)^2 \right) \\
&= \operatorname*{argmin}_{w \in \mathbb{R}} \left(w^2 \left(1 + \frac{4}{\epsilon}\right) + w \left(-4\gamma w_k - \frac{8\gamma w_k(1-\epsilon)}{\epsilon} \right) \right) \\
&= \operatorname*{argmin}_{w \in \mathbb{R}} \left(w^2 - 2w \frac{2\gamma\epsilon w_k + 4\gamma w_k(1-\epsilon)}{\epsilon + 4} \right) \\
&= \operatorname*{argmin}_{w \in \mathbb{R}} \left(w - \gamma w_k \frac{4 - 2\epsilon}{4 + \epsilon} \right)^2 \\
&= \frac{4 - 2\epsilon}{4 + \epsilon} \gamma w_k = \left(\frac{4 - 2\epsilon}{4 + \epsilon} \gamma \right)^k w_0.
\end{aligned}$$

For $w_0 \in \mathbb{R}, \epsilon \in (0, 1), \gamma \in [0, 1]$, it is clear that $\lim_{k \to \infty} w_k = 0$.

Week 10

Question. In the preceding weeks, we have stated the following two results without proofs. Both relate to the prediction task; say policy π is being evaluated on MDP (S, A, T, R, γ). Assume the MDP is continuing and ergodic; also assume standard conditions for annealing the learning rate.

R1. $TD(0)$ in the tabular setting (that is, with a separate entry for each state) converges to the underlying value function V^π.

R2. Linear $TD(\lambda)$, for $\lambda \in [0, 1]$, which computes the estimate $\hat{V}$ as a dot product of a d-dimensional feature vector of state and learned weight vector w, converges to w_λ^∞ satisfying

$$MSVE(w_\lambda^\infty) \leq \frac{1-\gamma\lambda}{1-\gamma} \min_{w \in \mathbb{R}^d} MSVE(w).$$

Show that R2 implies R1.

Solution. The tabular representation can be interpreted as a linear function "approximation" scheme using a *one-hot encoding* scheme. Herein, the number of features d is equal to the number of states $|S|$. For each state $s \in S$, there is a corresponding feature in the feature vector $x(s)$ that is set to 1; all the other features in $x(s)$ are 0. For example, if there are three states, their feature vectors are $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$.

Observe that with this approach, the weight w_s corresponding to (the feature corresponding to) a state s is essentially the value estimate $\hat{V}(s)$, since $\hat{V}(s) = w \cdot x(s) = w_s$. Also observe that $TD(0)$ with a tabular representation is identical to Linear $TD(0)$ with the one-hot encoding representation, in terms of the update made after each step.

Now consider the weight vector $w^\star$, which, for every $s \in S$, has $w^\star(s) = V^\pi(s)$. It is clear that $MSVE(w^\star) = 0$. It follows from R2 that Linear $TD(0)$ using the proposed linear architecture must converge to w_0^∞ satisfying $MSVE(w_0^\infty) = 0$, in turn meaning $w_0^\infty(s) = V^\pi(s)$ for $s \in S$. The convergence of the $TD(0)$ estimate $\hat{V}$ to V^π—that is, R1—is a consequence.

Week 9

Question. A learning agent interacts with an MDP (S, A, T, R, γ), where $S = \{s_1, s_2, s_3\}$ and $A = \{a_1, a_2, a_3\}$. No discounting is used ($\gamma = 1$).

The agent begins with the Q-table given below as initialisation $\hat{Q}^0$.

$\hat{Q}^0(s, a)$			
s	a		
	a_1	a_2	a_3
s_1	2	-3	1
s_2	2	2	2
s_3	0	-1	1

The agent uses ϵ-greedy exploration with $\epsilon = 0.15$, and a learning rate $\alpha = 0.1$ (both are constants, not annealed over time). Starting from state s_2, suppose that the agent's first transition is $(s_2, a_1, 2, s_3)$ (the next state is s_3 and the reward 2). From s_3, the agent decides to take action a_2. Thus, the agent's trajectory is $s_2, a_1, 2, s_3, a_2, \ldots$. What is $\hat{Q}^1$—the Q-table after making the first learning update? Give your answer for each of Q-learning, Sarsa, and Expected Sarsa being used for making the update. In each case provide the complete 3×3 table for $\hat{Q}^1$. In the absence of ties, note that a 0.15-greedy policy will pick the "argmax" action with probability 0.9, and each of the other two actions with probability 0.05.

Solution. Under all three methods, the only entry that changes is $\hat{Q}(s_2, a_1)$; all the other entries are carried forward from $\hat{Q}^0$ to $\hat{Q}^1$.

- Under Q-learning, we have

$$\hat{Q}^1(s_2, a_1) = \hat{Q}^0(s_2, a_1)(1 - \alpha) + \alpha\{2 + \max_{a \in A} \hat{Q}^0(s_3, a)\} = 2 \times 0.9 + 0.1 \times (2 + 1) = 2.1.$$

- Under Sarsa, we have

$$\hat{Q}^1(s_2, a_1) = \hat{Q}^0(s_2, a_1)(1 - \alpha) + \alpha\{2 + \hat{Q}^0(s_3, a_2)\} = 2 \times 0.9 + 0.1 \times (2 - 1) = 1.9.$$

- Under Expected Sarsa, the policy π used to pick an action at s_3 is reflected in the update. Since it is 0.15-greedy, we have

$$\begin{aligned}\hat{Q}^1(s_2, a_1) &= \hat{Q}^0(s_2, a_1)(1 - \alpha) + \alpha\{2 + \sum_{a \in A} \pi(s_3, a)\hat{Q}^0(s_3, a)\} \\ &= 2 \times 0.9 + 0.1 \times (2 + (0.05 \times 0 + 0.05 \times -1 + 0.9 \times 1)) = 2.085.\end{aligned}$$

The tables below compare results from the three update rules.

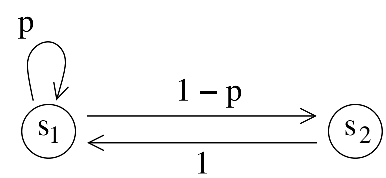
$\hat{Q}^1$, Q-learning			
s	a		
	a_1	a_2	a_3
s_1	2	-3	1
s_2	2.1	2	2
s_3	0	-1	1

$\hat{Q}^1$, Sarsa			
s	a		
	a_1	a_2	a_3
s_1	2	-3	1
s_2	1.9	2	2
s_3	0	-1	1

$\hat{Q}^1$, Expected Sarsa			
s	a		
	a_1	a_2	a_3
s_1	2	-3	1
s_2	2.085	2	2
s_3	0	-1	1

Week 8

Question. The figure below shows a *Markov chain*, which is defined by its states and transition probabilities. A Markov chain is what we get by fixing a policy for a given MDP (and ignoring the rewards and discount factor). The Markov chain in the figure has two states, s_1 and s_2. State s_1 loops back to itself with probability $p \in [0,1]$, and transitions to s_2 with probability $1-p$. State s_2 deterministically transitions to s_1. This question examines the ergodicity of the Markov chain.



Arrows are marked with transition probabilities.

Assume that for $t \geq 0$, s^t is the state occupied at time step t. For $t \geq 0$, $i,j \in \{1,2\}$, define $x^t_{i,j} \stackrel{\text{def}}{=} \mathbb{P}\{s^t = s_i | s^0 = s_j\}$; in other words, $x^t_{i,j}$ is the probability of being in state s_i at step t given the agent was at s_j at step 0. Based on the definition and the fact that the agent will be in either s_1 or s_2 at any time step $t \geq 0$, observe that we have

$$x^0_{11} = 1 \text{ and } x^0_{22} = 1; \tag{1}$$

$$x^t_{11} + x^t_{21} = 1 \text{ and } x^t_{12} + x^t_{22} = 1. \tag{2}$$

Write down x^t_{11}, x^t_{21}, x^t_{12}, and x^t_{22} as functions of p and t (to do this write down the variables for step $t+1$ in terms of those at t, and then solve the recurrence). Show that for $p \in (0,1)$,

$$\lim_{t\to\infty} x^t_{11} = \lim_{t\to\infty} x^t_{12} \text{ and } \lim_{t\to\infty} x^t_{21} = \lim_{t\to\infty} x^t_{22}.$$

Do these limits exist for $p = 0$ and for $p = 1$?

Solution. Based on the transition probabilities, we observe the recurrences

$$x^{t+1}_{11} = x^t_{11}(p) + x^t_{21} \text{ and } x^{t+1}_{22} = x^t_{12}(1-p) \tag{3}$$

for $t \geq 0$. Using (1), (2), and (3), we obtain

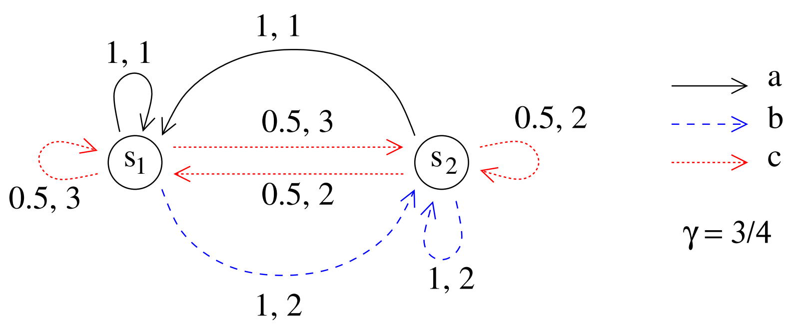
$$x^t_{11} = \frac{1 + (-1)^t(1-p)^{t+1}}{2-p}, \qquad x^t_{21} = \frac{1 - p - (-1)^t(1-p)^{t+1}}{2-p},$$
$$x^t_{12} = \frac{1 - (-1)^t(1-p)^t}{2-p}, \qquad x^t_{22} = \frac{1 - p + (-1)^t(1-p)^t}{2-p}.$$

for $t \geq 0$. For $p \in (0,1)$, we get $\lim_{t\to\infty} x^t_{11} = \lim_{t\to\infty} x^t_{12} = \frac{1}{2-p}$, and $\lim_{t\to\infty} x^t_{21} = \lim_{t\to\infty} x^t_{22} = \frac{1-p}{2-p}$. The same limits hold for $p = 1$, but the Markov chain is not irreducible in this case (hence not ergodic). The limits are not well-defined for $p = 0$ since the probabilities are exactly 0 or 1 depending on the parity of t.

Week 6

Question. This question calls for a straightforward application of definitions introduced in the Week 6 lecture. Consider the MDP shown in the figure below. It has two states: s_1 and s_2; and three actions: a, b, and c. Action a is deterministic, always leading to state s_1; action b is also deterministic, but always leading to state s_2. Action c, on the other hand, keeps the agent in the starting state with probability $1/2$, and moves the agent to the other state with probability $1/2$.

Action a merits a reward of 1 and action b a reward of 2 regardless of the state from which they are taken. Action c yields a reward of 3 if taken from s_1 and a reward of 2 if taken from s_2. Observe that all the rewards can be written in terms of the starting state and action alone, with no dependence on the next state. The MDP has a discount factor $\gamma = 3/4$.



Arrows are marked with "probability, reward"; transitions with zero probability are not shown.

Consider the policy $\pi =$ "ac", which takes action a from s_1 and action c from s_2. What are the improving actions for s_1 and s_2 under this policy? In other words, what are $\mathbf{IA}(ac, s_1)$ and $\mathbf{IA}(ac, s_2)$? Show the working to arrive at your answer.

Solution. The Bellman equations for policy ac are:

$$V^{ac}(s_1) = 1 + \gamma V^{ac}(s_1);\ \text{and}\ V^{ac}(s_2) = \frac{1}{2}\{2 + \gamma V^{ac}(s_1)\} + \frac{1}{2}\{2 + \gamma V^{ac}(s_2)\},$$

solving which we obtain $V^{ac}(s_1) = 4$; $V^{ac}(s_2) = 28/5 = 5.6$. Using V^{ac}, we calculate Q^{ac} for the actions not taken at each state:

$$Q^{ac}(s_1, b) = 2 + \gamma V^{ac}(s_2) = 31/5 = 6.2.$$
$$Q^{ac}(s_1, c) = \frac{1}{2}\{3 + \gamma V^{ac}(s_1)\} + \frac{1}{2}\{3 + \gamma V^{ac}(s_2)\} = 33/5 = 6.6.$$
$$Q^{ac}(s_2, a) = 1 + \gamma V^{ac}(s_1) = 4.$$
$$Q^{ac}(s_2, b) = 2 + \gamma V^{ac}(s_2) = 31/5 = 6.2.$$

Notice that $Q^{ac}(s_1, b)$ and $Q^{ac}(s_1, c)$ both exceed $V^{ac}(s_1)$, whereas $Q^{ac}(s_2, a) < V^{ac}(s_2) < Q^{ac}(s_2, b)$. Consequently we have

$$\mathbf{IA}(ac, s_1) = \{b, c\};$$
$$\mathbf{IA}(ac, s_2) = \{b\}.$$

Week 5

Question. For an MDP (S, A, T, R, γ), let $V_0 : S \to \mathbb{R}$ be an initial guess of the optimal value function $V^\star$. Suppose that this guess is progressively updated using Value Iteration: that is, by setting $V_{t+1} \leftarrow B^\star(V_t)$ for $t = 0, 1, 2, \ldots$. Recall that $B^\star$ is the Bellman optimality operator.

In this question, we examine the design of a stopping condition for Value Iteration. As usual, let $\|\cdot\|_\infty$ denote the max norm. We would like that our computed solution, V_u for some $u \in \{1, 2, \ldots\}$, be within ϵ of $V^\star$ for some given tolerance $\epsilon > 0$. In other words, we would like to stop after u applications of $B^\star$, so long as we can *guarantee* $\|V_u - V^\star\|_\infty \leq \epsilon$. Naturally, we cannot use $V^\star$ itself in our stopping rule, since it is not known! Show that it suffices to stop when

$$\|V_u - V_{u-1}\|_\infty \leq \frac{\epsilon(1-\gamma)}{\gamma}.$$

and thereafter return V_u as the answer.

You are likely to find two results handy: (1) that $B^\star$ is a contraction mapping with contraction factor γ, and (2) the triangle inequality: for $X : S \to \mathbb{R}, Y : S \to \mathbb{R}, \|X + Y\|_\infty \leq \|X\|_\infty + \|Y\|_\infty$.

Solution. Let $\epsilon' = \frac{\epsilon(1-\gamma)}{\gamma}$. We are given $\|V_u - V_{u-1}\|_\infty \leq \epsilon'$; by successive application of the result that $B^\star$ is a contraction mapping with contraction factor γ, we get

$$\begin{aligned}
\|V_u - V_{u-1}\|_\infty &\leq \epsilon', \\
\|B^\star(V_u) - B^\star(V_{u-1})\|_\infty &\leq \epsilon'\gamma, \\
\|(B^\star)^2(V_u) - (B^\star)^2(V_{u-1})\|_\infty &\leq \epsilon'\gamma^2, \\
&\vdots \\
\|(B^\star)^k(V_u) - (B^\star)^k(V_{u-1})\|_\infty &\leq \epsilon'\gamma^k
\end{aligned}$$

for all $k \geq 0$. By using the triangle inequality, we obtain

$$\|(B^\star)^k(V_u) - V_u\|_\infty \leq \sum_{j=1}^{k} \|(B^\star)^k(V_u) - (B^\star)^k(V_{u-1})\|_\infty \leq \epsilon'(\gamma + \gamma^2 + \cdots + \gamma^k)$$

for all $k \geq 0$. Taking the limit as $k \to \infty$ yields $\|V^\star - V_u\|_\infty \leq \frac{\epsilon'\gamma}{1-\gamma} = \epsilon$.

Week 4

Question. In this week's lecture, we derived Bellman equations for policy evaluation. If $M = (S, A, T, R, \gamma)$ is our input MDP, we showed for every policy $\pi : S \to A$ and state $s \in S$:

$$V^{\pi}(s) = \sum_{s' \in S} T(s, \pi(s), s')\{R(s, \pi(s), s') + \gamma V^{\pi}(s')\}.$$

This question considers four variations in our definitions or assumptions regarding the input MDP M and policy π. In each case write down Bellman equations after making appropriate modifications. The set of equations for each case will suffice; no need for additional explanation.

a. The reward function R does not depend on the next state s'; it is given to you as $R : S \times A \to \mathbb{R}$.

b. The reward function R depends only on the next state s'; it is given to you as $R : S \to \mathbb{R}$.

c. The policy π is stochastic: for $s \in S, a \in A$, $\pi(s, a)$ denotes the probability with which the policy takes action a from state s.

d. The underlying MDP M is deterministic. Hence, the transition function T is given as $T : S \times A \to S$, with the semantics that $T(s, a)$ is the next state $s' \in S$ for $s \in S, a \in A$.

Solution. Answers are given below for all policies π and states $s \in S$.

a. $V^{\pi}(s) = R(s, \pi(s)) + \gamma \sum_{s' \in S} T(s, \pi(s), s') V^{\pi}(s')$.

b. $V^{\pi}(s) = \sum_{s' \in S} T(s, \pi(s), s')\{R(s') + \gamma V^{\pi}(s')\}$.

c. $V^{\pi}(s) = \sum_{a \in A} \pi(s, a) \sum_{s' \in S} T(s, a, s')\{R(s, a, s') + \gamma V^{\pi}(s')\}$.

d. $V^{\pi}(s) = R(s, \pi(s), T(s, \pi(s))) + \gamma V^{\pi}(T(s, \pi(s)))$.

Week 3

Question. A 2-armed bandit instance I has as the mean rewards of its arms $p_1, p_2 \in [0, 1]$, where $|p_1 - p_2| = \Delta > 0$. Both arms produce 0 and 1 rewards (that is, from Bernoulli distributions).

Suppose we are given Δ, but we do not know which arm has the higher mean reward. Our aim is to determine the optimal arm with probability at least $1 - \delta$. In order to do so, we pull each arm N times, and declare as our answer the arm which registers the higher empirical mean (breaking ties uniformly at random).

Show that it suffices to set

$$N = \theta\left(\frac{1}{\Delta^2}\log\left(\frac{1}{\delta}\right)\right)$$

in order to indeed give the correct answer with probability at least $1 - \delta$.

Solution. Without loss of generality, let arm 1, with mean p_1, be the optimal arm, and arm 2, with mean p_2, be the sub-optimal arm. Intuition suggests that as N becomes larger, the probability that arm 1 is returned increases. We will build a proof assuming N is sufficiently large—and take it to the point that the proof itself suggests to us how N must be set.

After N pulls each, let the empirical means of the arms be $\hat{p}_1$ and $\hat{p}_2$, respectively. Consider the mid-point between these means, $\frac{p_1+p_2}{2}$, as a "boundary", in the sense that the answer is guaranteed to be correct if neither empirical mean "crosses" the boundary. In other words, if each empirical mean stays within $\frac{\Delta}{2}$ of the true mean on its corresponding side, then $\hat{p}_1$ must exceed $\hat{p}_2$, thereby yielding the right answer. We invoke Hoeffding's Inequality to bound the deviation probability.

$$\begin{aligned}
\mathbb{P}\{\text{Wrong answer given}\} &\leq \mathbb{P}\{\hat{p}_1 \leq \hat{p}_2\} \\
&\leq \mathbb{P}\left\{\hat{p}_1 \leq \frac{p_1+p_2}{2} \text{ or } \hat{p}_2 \geq \frac{p_1+p_2}{2}\right\} \\
&\leq \mathbb{P}\left\{\hat{p}_1 \leq \frac{p_1+p_2}{2}\right\} + \mathbb{P}\left\{\hat{p}_2 \geq \frac{p_1+p_2}{2}\right\} \\
&= \mathbb{P}\left\{\hat{p}_1 \leq p_1 - \frac{\Delta}{2}\right\} + \mathbb{P}\left\{\hat{p}_2 \geq p_2 + \frac{\Delta}{2}\right\} \\
&\leq e^{-2N(\Delta/2)^2} + e^{-2N(\Delta/2)^2}.
\end{aligned}$$

Suppose we had set N such that $2e^{-2N(\Delta/2)^2} \leq \delta$, we would have an acceptable proof to go with that choice! Observe that it suffices to take $N = \lceil \frac{2}{\Delta^2}\ln(\frac{2}{\delta})\rceil$.

Week 2

Question. In this question, we consider bandit instances in which the number of arms $n = 10$; assume the set of arms is $A = \{0, 1, 2, \ldots, 9\}$. Each arm yields rewards from a Bernoulli distribution whose mean is strictly less than 1. Call this set of bandit instances $\bar{\mathcal{I}}$.

Now consider a family of algorithms $\mathcal{L}$ that operate on $\bar{\mathcal{I}}$, wherein each algorithm $L \in \mathcal{L}$ satisfies the following properties.

- L is deterministic.
- In the first n pulls made by L (on steps $0 \leq t \leq n-1$), each arm is pulled exactly once.
- For $t = n, n+1, n+2, \ldots$: if t is *not* a prime number, then the arm pulled by L on the t-th step has the highest empirical mean among all the arms at that step.

In other words, each $L \in \mathcal{L}$ is a deterministic algorithm that begins with round-robin sampling for n pulls, and thereafter *exploits* on every step t that is not a prime number. You can assume ties are broken arbitrarily. The chief difference between the elements of $\mathcal{L}$ arises from the decisions they make on steps t that are prime numbers—there is no restriction on the choice made on such steps.

a. Show that there exists $L_{\text{good}} \in \mathcal{L}$ such that L_{good} achieves sub-linear regret on all $I \in \bar{\mathcal{I}}$.

a. Show that there exists $L_{\text{bad}} \in \mathcal{L}$ such that L_{bad} does not achieve sub-linear regret on all $I \in \bar{\mathcal{I}}$.

Your arguments can be informal: no need for the dense notation of Class Note 1. You can use the fact that the number of prime numbers smaller than any natural number N is $\theta(\frac{N}{\log(N)})$.

Solution. For part (a), it suffices to show that there exists $L_{\text{good}} \in \mathcal{L}$ that is GLIE. For every prime number t, let $m(t)$ denote the number of prime numbers smaller than t. Thus $m(2) = 0, m(3) = 1, m(5) = 2, \ldots$. Take L_{good} as an algorithm that on every step t that is a prime number, pulls arm $m(t) \mod n$. It is clear that L_{good} will pull each arm infinitely often in the limit. Furthermore, the number of "exploit" steps up to horizon T is at least $T - \theta(\frac{T}{\log(T)})$. For $I \in \bar{\mathcal{I}}$, we have

$$\lim_{T \to \infty} \frac{\mathbb{E}_{L_{\text{good}}, I}[exploit(T)]}{T} = \lim_{T \to \infty} \left(1 - \theta\left(\frac{1}{\log(T)}\right)\right) = 1,$$

implying that L_{good} is greedy in the limit.

For part (b), it suffices to show that there exists $L_{\text{bad}} \in \mathcal{L}$ that is not GLIE: in particular we need only show that L_{bad} is not guaranteed to pull each arm infinitely often in the limit. Take L_{bad} to be an algorithm that only pulls arm 0 on steps t that are prime numbers. On any instance in which the means of arms are in increasing order of their index (hence arm 9 is the sole optimal arm), there is a non-zero probability that arm 9 will initially give a 0-reward, some other arm a 1-reward, and thereafter arm 9 will never get pulled by L_{bad}. On such an instance, L_{bad} incurs linear regret.

In summary: the prime number bound guarantees that each $L \in \mathcal{L}$ will be greedy in the limit, and it also allows for infinite exploration. Whether $L \in \mathcal{L}$ actually performs infinite exploration of each arm determines the sub-linearity of its regret.

Week 1

Question. Consider a 2-armed bandit instance B in which the rewards from the arms come from *uniform* distributions (recall that the lectures assumed they came from Bernoulli distributions). The rewards of arm 1 are drawn uniformly at random from $[a, b]$, and the rewards of arm 2 are drawn uniformly at random from $[c, d]$, where $0 < a < c < b < d < 1$. Observe that this means there is an overlap: both arms produce some rewards from the interval $[c, b]$.

An algorithm L proceeds as follows. First it pulls arm 1; then it pulls arm 2; whichever of these arms produced a higher reward (or arm 1 in case of a tie) is then pulled a further 20 times. In other words, the algorithm performs round-robin exploration for 2 steps and greedily picks an arm for the subsequent exploitation phase, during which that arm is blindly pulled 20 times. What is the expected cumulative regret of L on B after 22 pulls?

(If you have worked out an answer but are not sure about it, consider writing a small program to simulate L and run it many times for fixed a, b, c, d. Is the average regret from these runs close to your answer? The program is for your own sake; no need to submit or to explain to us.)

Solution. The mean reward of arm 1 is $p_1 = \frac{a+b}{2}$ and the mean reward of arm 2 is $p_2 = \frac{c+d}{2}$. Since $a < c$ and $b < d$, it is clear that arm 2 is optimal.

The expected cumulative regret from the 22 pulls is the sum of those from the first 2 pulls and from the subsequent exploitation phase (20 pulls). In the first two pulls, the expected cumulative regret is exactly $p_2 - p_1$, since arm 1 (the suboptimal arm) is pulled exactly once. In the exploitation phase, the expected cumulative regret is 0 in case arm 2 is played, and $20(p_2 - p_1)$ if arm 1 is pulled. The expected cumulative regret from exploitation is therefore $\mathbb{P}\{\text{arm 1 is selected after first 2 steps}\} \cdot 20 \cdot (p_2 - p_1)$.

What is the probability that arm 1 gets selected after the first two pulls? We know that each reward x_1 from arm 1 is drawn from $[a, b]$ according to pdf $\frac{1}{b-a}$. Similarly, the reward x_2 from arm 2 is drawn from $[c, d]$ according to pdf $\frac{1}{d-c}$. The probability that $x_1 \geq x_2$ is therefore

$$\mathbb{P}\{\text{arm 1 is selected after first 2 steps}\} = \int_{x_1=c}^{b} \int_{x_2=c}^{x_1} \frac{1}{(b-a)(d-c)} dx_2 dx_1 = \frac{(c-b)^2}{2(b-a)(d-c)}.$$

An alternative argument to obtain this probability is that (1) x_1 falls in $[c, b]$ with probability $\frac{c-b}{b-a}$, (2) x_2 falls in $[c, b]$ with probability $\frac{c-b}{d-c}$, and (3) conditioned on x_1 and x_2 both falling in $[c, b]$, each has a uniform distribution in that range, and thus the probability that one exceeds the other is $1/2$.

The expected cumulative regret from the 22 pulls is thus

$$(p_2 - p_1) + \mathbb{P}\{\text{arm 1 is selected after first 2 steps}\} \cdot 20 \cdot (p_2 - p_1) = \frac{c+d-a-b}{2} \left(1 + \frac{10(c-b)^2}{(b-a)(d-c)}\right).$$