CS 747 (Autumn 2021): Weekly Quizzes

Instructor: Shivaram Kalyanakrishnan

September 2, 2021

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 5

Question. Consider an MDP (S, A, T, R, γ) in which the set of states is $S = \{1, 2, \ldots, n\}$. This question relates to a *variation* of value iteration. Concretely, let us denote the variant given in class CV (for "class variant"), and the one given by the pseudocode below QV (for "quiz variant"). In QV, each state is initialised with value 0. Subsequently the value function V is updated through T iterations; in each iteration all n state values get updated.

> **QV**
> $V \leftarrow n$-dimensional $\mathbf{0}$ vector.
> For $i = 0, 1, \ldots, T-1$:
>   For $s = 1, 2, \ldots, n$:
> $V(s) \leftarrow \max_{a \in A} \sum_{s' \in S} T(s, a, s')\{R(s, a, s') + \gamma V(s')\}$.
> Return V.

a. Pay close attention to the pseudocode. In qualitative terms, what is the main difference between QV and CV? In quantitative terms, derive an operator $B : (S \to \mathbb{R}) \to (S \to \mathbb{R})$ that is being implemented by QV. In other words, for what B can we describe the code above as T successive applications of B to the initial value vector? Feel free to use the Bellman optimality operator $B^\star$, as well as recursion, in your definition of B. [2 marks]

b. QV is used quite commonly in practice, and is known to converge to $V^\star$ (as $T \to \infty$). Your job is to prove this to be the case. To that end, show that (1) B is a contraction mapping in a Banach space, and (2) its fixed point is $V^\star$. [4 marks]

Week 4

Question. This question is about the probability of transitioning between the states of an MDP over an extended period of time. Taking notations as usual, assume that you are given

- an MDP (S, A, T, R, γ),
- a start state $s_{\text{start}} \in S$,
- a policy $\pi : S \to A$,
- a non-negative integer $t \geq 0$ specifying the number of time steps elapsed (the same as the number of actions taken), and
- an arbitrary state $s_{\text{finish}} \in S$.

For $s \in S$, $t \geq 0$, let $X[t][s]$ denote the probability that the agent is in state s at time t, assuming it is in s_{start} at $t = 0$, and it takes actions according to π at every step. By initialisation, we have

$$X[0][s] = \begin{cases} 1 & s = s_{\text{start}}, \\ 0 & \text{otherwise.} \end{cases}$$

Provide pseudocode to compute $X[t][s_{\text{finish}}]$. You can refer to any subset of the input parameters—S, A, T, R, γ, s_{start}, π, t, s_{finish}—in your pseudocode, using notations as introduced in class (such as $T(s, a, s')$ to denote a transition probability and $\pi(s)$ to denote an action). To obtain full marks, you must provide an algorithm whose running time scales at most polynomially in t. [3 marks]

Solution. For $i \geq 0$, the agent is in state s at time $i + 1$ if and only if it made a transition into s from the state s' in which it was at time i. Hence we obtain a recursive relationship between $X[i + 1][\cdot]$ and $X[i][\cdot]$, which involves the transition probabilities under π. This recurrence can be used to compute $X[t][s_{\text{finish}}]$. Pseudocode is provided below.

For $s \in S$:
    $X[0][s] \leftarrow 0$.
$X[0][s_{\text{start}}] \leftarrow 1$.

For $i = 0, 1, \ldots, t - 1$:
    For $s \in S$:
        $X[i + 1][s] \leftarrow \sum_{s' \in S} X[i][s']T(s', \pi(s'), s)$.
Return $X[t][s_{\text{finish}}]$.

The sum in the recurrence can be performed using a for-loop or a vector dot product. Notice that $O(|S|^2)$ arithmetic operations are performed to obtain $X[i+1]$ from $X[i]$, and there are t such iterations in total. There is no dependence on $|A|$.

Can you think of a way to reduce the dependence on t from linear to *logarithmic*? The answer lies in writing down $X[t]$ as the product of a t-step transition *matrix* (of dimension $|S| \times |S|$) and $X[0][\cdot]$. This matrix is itself the t-th power of a single-step transition matrix, and can be computed in $O(\log(t + 1))$ matrix multiplications. The dependence on $|S|$ remains polynomial.

Week 3

Question. You are familiar with the UCB algorithm applied to Bernoulli bandits. The algorithm selects an arm to pull by being greedy with respect to the arms' upper confidence bounds. Consider arm a that has been pulled $u_a^t \geq 1$ time(s) out of a total of $t \geq 1$ pull(s), and has empirical mean $\hat{p}_a^t$. The upper confidence bound for this arm is given by

$$\text{ucb}_a^t = \hat{p}_a^t + \sqrt{\frac{1}{2u_a^t} \ln\left(\frac{1}{\delta(t)}\right)},$$

where the common choice is to set $\delta(t) = \frac{1}{t^4}$. The upper confidence bound used by KL-UCB is

$$\text{ucb-kl}_a^t = \text{the solution } q \in [\hat{p}_a^t, 1] \text{ that satisfies } u_a^t \text{KL}(\hat{p}_a^t, q) = \ln\left(\frac{1}{\delta'(t)}\right),$$

wherein we commonly take $\delta'(t) = \frac{1}{t \cdot (\ln t)^c}$ for some fixed $c \geq 3$. Recall that for $x, y \in [0,1]$, $\text{KL}(x, y)$ denotes the KL-divergence between Bernoulli distributions with means x and y, respectively.

a. Show that if $0 < \delta(t) \leq \delta'(t)$, then regardless of the number of pulls u_a^t and empirical mean $\hat{p}_a^t$, we are guaranteed that $\text{ucb-kl}_a^t \leq \text{ucb}_a^t$. You can use the well-known Pinsker's Inequality, which states that for $x, y \in [0,1]$, $\text{KL}(x,y) \geq 2(x-y)^2$. [2 marks]

b. The intuition behind KL-UCB incurring lower regret than UCB is that it uses a "tighter" upper confidence bound, as established in part a. Extending this logic, suppose we propose an even tighter quantity

$$\text{ucb-proposed}_a^t = \frac{1}{2}\left(\hat{p}_a^t + \text{ucb-kl}_a^t\right),$$

which clearly satisfies $\text{ucb-proposed}_a^t \leq \text{ucb-kl}_a^t$. May we expect an algorithm that is greedy with respect to ucb-proposed to incur even lower regret than ucb-kl_a^t? Explain. [1 mark]

Solution.
a. By definition, $u_a^t \text{KL}(\hat{p}_a^t, \text{ucb-kl}_a^t) = \ln\left(\frac{1}{\delta'(t)}\right)$. Applying Pinsker's Inequality to the LHS and and the relation between δ and δ' to the RHS, we get $2u_a^t(\text{ucb-kl}_a^t - \hat{p}_a^t)^2 \leq \ln\left(\frac{1}{\delta(t)}\right)$, which, in turn, can be rearranged to obtain

$$\text{ucb-kl}_a^t \leq \hat{p}_a^t + \sqrt{\frac{1}{2u_a^t} \ln \frac{1}{\delta(t)}} = \text{ucb}_a^t.$$

b. Note that ucb_a^t and ucb-kl_a^t are both genuine upper confidence bounds: with probability $\delta(t)$ or $\delta'(t)$, respectively, there is a guarantee that the true mean does not exceed them. And we have formal proofs that for mistake probability $\delta(t)$ or $\delta'(t)$, the corresponding algorithm achieves logarithmic regret. On the other hand, it is not clear that ucb-proposed will be an upper bound on the mean with sufficiently high probability. It runs the risk of not exploring at a sufficient rate. An ideal upper confidence bound would be one that achieves the required "mistake probability", while still being as tight as possible.

Another perspective on the question would be that since KL-UCB is proven to be asymptotically optimal, it is not possible for any other algorithm to improve upon it substantively. However, note that there remains room for an algorithm to always achieve lower regret than KL-UCB, but only, say, by at most a constant amount. Such an occurrence would not contradict known theory.

Week 2

Question. Since the UCB algorithm achieves logarithmic regret on every bandit instance, we may infer that it satisfies the GLIE conditions. In this question, you are to argue from first principles that indeed UCB performs an infinite amount of exploration. To simplify our argument, we only consider a 2-armed bandit instance with arms 1 and 2. Suppose that the algorithm is (1) initialised by pulling each arm once, and (2) thereafter it is greedy with respect to the arms' upper confidence bounds at each time step, (3) breaking ties uniformly at random.

Adopting the usual notation, let u_a^t and $\hat{p}_a^t$ denote the number of pulls and the empirical mean of arm $a \in \{1, 2\}$ after $t \geq 2$ pulls (which ensures that the empirical means are well-defined). We consider an arbitrary t-length history h, summarised by t, u_1^t, $\hat{p}_1^t$, u_2^t, $\hat{p}_2^t$. We contemplate: *is it possible that one of the arms will never get pulled after encountering h?* Your task is to show that on the contrary, there exists a finite integer T (which can be defined in terms of t, u_1^t, $\hat{p}_1^t$, u_2^t, $\hat{p}_2^t$ or some subset of them) such that the T pulls following h are *guaranteed* to have at least one pull of each arm. It is okay of you unable to work out an explicit formula for T, but are still able to formally argue for its existence. Support your claims with rigorous justification, rather than appealing to "intuition" and informal observations. [4 marks]

Solution. Suppose that for some $x > 1$, the x pulls following h are all of the same arm, which, without loss of generality, we may take as arm 1. Then we have

$$\begin{aligned}
ucb_2^{t+x} - ucb_1^{t+x} &= \hat{p}_2^{t+x} + \sqrt{\frac{2\ln(t+x)}{u_2^t}} - \hat{p}_1^{t+x} - \sqrt{\frac{2\ln(t+x)}{u_1^t + x}} \\
&\geq 0 + \sqrt{\frac{2\ln(t+x)}{t}} - 1 - \sqrt{\frac{2\ln(t+x)}{x}} \\
&= -1 + \sqrt{\frac{2\ln(t+x)}{t}}\left(1 - \sqrt{\frac{t}{x}}\right).
\end{aligned}$$

If we choose any $x \geq e^t + 16$, we have $\ln(t+x) > t$ and $\sqrt{t/x} \leq \sqrt{2/(e^2+16)} < 0.2925$, using the fact that $\sqrt{\frac{t}{e^t+16}}$ is a decreasing function of t, which is maximised in our domain at $t = 2$. Consequently we get

$$ucb_2^{t+x} - ucb_1^{t+x} > -1 + \sqrt{2}(1 - 0.2925) > 0.$$

Thus, even if all x pulls have been of arm 1, it is clear that the next pull $(t + x + 1)$ must be of arm 2. For the choice of $T = \lceil e^t \rceil + 17$, we have established that the T pulls following h cannot all be of the same arm.

Week 1

Question. Consider the family of n-armed bandit instances, $n \geq 2$, in which each arm $a \in \{1, 2, \ldots, n\}$ generates a 1-reward with probability p_a and a 0-reward with probability $1 - p_a$. Thus, each instance of the family is fixed by a vector $(p_1, p_2, \ldots, p_n)$, where $p_a \in [0, 1]$ for $a \in \{1, 2, \ldots, n\}$.

A round-robin algorithm undertakes $m \geq 2$ passes over the set of arms; the sequence of pulls $1, 2, \ldots, n$ is repeated m times. For each arm $a \in \{1, 2, \ldots, n\}$, let s_a denote the number of 1-rewards (interpreted as "successes") from its m pulls, and let f_a denote the number of 0-rewards (interpreted as "failures") from its m pulls (hence $s_a + f_a = m$).

a. For a fixed bandit instance $(p_1, p_2, \ldots, p_n)$, what is the probability that $s_1 = s_2 = \cdots = s_n$? Give your answer in terms of $p_1, p_2, \ldots, p_n$, and m. [2 marks]

b. Denote the total number of successes after the m passes $S = s_1 + s_2 + \cdots + s_n$. What are the mean and variance of S? Again, your answer must be in terms of $p_1, p_2, \ldots, p_n$, and m. [2 marks]

It will help to view the reward given by each pull as a random variable, noting that it is independent of the $(nm - 1)$ others. This view can facilitate an easy computation of the variance of S in part b—in your answer, be sure to explain why.

Solution.

a. Each arm a is pulled m times. The probability that it gets s_a successes and f_a failures for $0 \leq s_a \leq m$, $s_a + f_a = m$, is $\binom{m}{s_a}(p_a)^{s_a}(1 - p_a)^{f_a}$. For any fixed number of successes $s \in \{0, 1, \ldots, m\}$, the probability that all n arms get s successes is

$$\prod_{a \in \{1,2,\ldots,n\}} \binom{m}{s}(p_a)^s(1 - p_a)^{m-s}.$$

The required probability takes into account all possible values of s, and is thus

$$\sum_{s=0}^{m} \prod_{a \in \{1,2,\ldots,n\}} \binom{m}{s}(p_a)^s(1 - p_a)^{m-s}.$$

b. S is seen to be the sum of nm Bernoulli variables $X_{a,l}$ for arm $a \in \{1, 2, \ldots, n\}$ and pass $l \in \{1, 2, \ldots m\}$. The mean of $X_{a,l}$ is p_a, and its variance is $p_a(1 - p_a)$. We use

$$\mathbb{E}[S] = \sum_{a=1}^{n} \sum_{l=1}^{m} \mathbb{E}[X_{a,l}] = m \sum_{a=1}^{n} p_a.$$

Since the variables are independent, we also have

$$\mathrm{Var}[S] = \sum_{a=1}^{n} \sum_{l=1}^{m} \mathrm{Var}[X_{a,l}] = m \sum_{a=1}^{n} p_a(1 - p_a).$$

Note that if random variables X and Y are *not* independent, it is not necessary that they satisfy $\mathrm{Var}[X + Y] = \mathrm{Var}[X] + \mathrm{Var}[Y]$. In typical bandit algorithms (such as ϵ-greedy sampling), the *arm* that is pulled at some fixed time step could itself be random, disallowing the decomposition of S into $\sum_{a=1}^{n} \sum_{l=1}^{m} X_{a,l}$, which makes our variance-calculation convenient.