ECE 8803: Online Decision Making in Machine Learning **Fall 2021**

Lecture 12: October 4

Lecturer: Vidya Muthukumar

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

Last lecture, we introduced the setting of limited-information feedback and motivated what we called the *multi-armed bandit* problem through a classic application of drug discovery through clinical trials. This example goes all the way back to William Thompson in 1933 (Thompson, 1933). Today, we will more formally review the multi-armed bandit (MAB) problem and the metric of *pseudo-regret* that we introduced at the end of last lecture. Then, we will introduce the popular *upper-confidence-bound* (UCB) algorithm for this problem, and visualize its performance on random MAB instances.

12.1. Recap: The multi-armed bandit problem

Last lecture, we introduced K-armed bandit problem with reward means given by $(\mu_1, \dots, \mu_K)$. The best arm was denoted by $a^* = \arg\max_{a \in [K]} \mu_a$, and its corresponding reward was given by $\mu^* := \mu_{a^*}$. In our drug discovery example, we considered $\mu_1 = 0.2, \mu_2 = 0.7$, so $a^* = 2$ and $\mu^* = 0.7$, and Bernoulli "rewards" (although we did not have to do this: the rewards could as easily just be continuous-valued between $[0, 1]$).

Our goal is to select actions A_t at every round t, based on past information, to maximize the *expected* long-term reward, given by

$$\mathbb{E}[G_T] := \mathbb{E}\left[\sum_{t=1}^{T} G_{t,A_t}\right],$$

over T rounds of decision-making. It is important to understand why we are taking an expectation here: it is two-fold:

- For one, the rewards themselves are random, i.e. $\mathbb{E}[G_{t,a}] = \mu_a$ at any round t and for any action $a \in \{1, \dots, K\}$. Recall that these rewards in our drug example denoted the effect of a drug on a patient which we model as a random variable. Since all patients are from the same population, the rewards are iid.
- For another, the actions A_t that are taken at time t in general will depend on the realizations of the rewards in past rounds. This means that the actions A_t will also be random variables that depend on past outcomes (although they are often deterministic *conditioned* on the past).

Last lecture, we evaluated a number of different approaches to do this, which we will now recap.

12.1.1 Pseudo-regret, and what it means

Let us think about the policy that would maximize the expected reward $\mathbb{E}[G_T]$ as defined above. What would this look like? It would simply pick the best action a^* on all rounds, giving us $G_{\max} = T\mu^*$. Of course, this policy is one that we will not know about apriori, since the means are unknown to us. However, it constitutes the most sensible benchmark for performance; in particular, we would like to study how much less reward we get than this optimal benchmark. This constitutes the metric of *pseudo-regret*, which is defined as

$$\overline{R}_T := T\mu_{a^*} - \mathbb{E}[G_T]. \tag{12.1}$$

Note the "bar" that we have placed on top of the pseudo-regret quantity in order to distinguish it from the notions of regret that we studied previously in the class for online prediction and optimization with full-information feedback. In the stochastic MAB problem that we are studying, the benchmark is a *learning-based* one: in particular, how well we could have done had we known the means $(\mu_1, \ldots, \mu_K)$ beforehand (or just the identity of the optimal arm).

It turns out to be convenient to write the pseudo-regret directly in terms of the number of times each of the suboptimal arms is sampled. Let us say that our choice of algorithm sampled a suboptimal arm, denoted by a, $N_a(T)$ times, where $N_a(T)$ is a random variable that depends on the actual outcomes of the rewards. (For example, recall that in our drug discovery example from last time, $a = 1$ was the suboptimal arm. We saw that the greedy algorithm could yield two very different outcomes: one where we had $N_a(T) = 1$, and the other where we had $N_a(T) = T - 1$!) Concretely, we can write the pseudo-regret equivalently as follows.

Definition 1 *Let $\Delta_a := \mu^* - \mu_a$ for each suboptimal arm $a \neq a^*$. Then, the pseudo-regret of any algorithm can be equivalently written as*

$$\overline{R}_T := \sum_{a \neq a^*} \Delta_a \cdot \mathbb{E}[N_a(T)]. \tag{12.2}$$

Equation (12.2) has an intuitive meaning: the less number of times a suboptimal arm is pulled, the less the pseudo-regret will be. See Lemma 4.5 in Lattimore and Szepesvári (2020) for a formal proof of this fact. Henceforth, all of our intuitive discussion as well as formal proofs will use this definition of pseudo-regret.

12.1.2 Suboptimal algorithms

Now, we quickly recall the suboptimal algorithms that we had introduced last lecture, and mention without proof their pseudo-regret guarantees. Homework 3 has you explore and show these guarantees for yourself in a hands-on manner.

- Greedy (only exploit): $A_t = \arg\max_a \widehat{\mu}_{t,a}$. This turns out to incur linear pseudo-regret in T because of the very bad and non-vanishing possibility of pulling the suboptimal arm all the time.
- Only explore: select all arms in a round-robin fashion. This can easily be verified to incur linear pseudo-regret as well.

- Explore-then-commit for T_0 rounds: the homework has you show that you can expect sub-linear regret with this, with some caveats.
- ϵ-greedy with a randomizing schedule given by $\{\epsilon_t\}_{t\geq 1}$: the homework has you show that you can also expect sub-linear regret with this, but the guarantee may not be optimal.

12.2. The upper-confidence-bounding algorithm

ETC and ϵ-greedy both constitute *heuristic* approaches to trading off exploration and exploitation in our algorithm. While they make some headway in ensuring sublinear pseudo-regret, they turn out to not be optimal. The reason, at a high level, is because they do not adapt the tradeoff of exploration-vs-exploitation to the properties of the data at hand. For example, it is intuitive to see that the case where Drug A is 20% effective and drug B is 70% effective will require more exploration than a hypothetical case in which Drug A is 10% effective and Drug B is 90% effective, simply because the former case has more randomness, and the mean efficacies of each drug are "closer" to one another — so we need to collect more samples of each to offset possible adverse effects due to randomness in the rewards.

We will now explore an algorithm that turns out to trade off exploration and exploitation in precisely this way. This is called the *upper-confidence-bound* algorithm, and is formally defined below.

Definition 2 *At round t, let $\widehat{\mu}_{a,t-1}$ denote the sample mean of arm a, and $N_{t-1}(a)$ denote the number of times that arm a was sampled until then. (Note that both of these are random variables owing to the random reward feedback.) Then, UCB selects the action A_t as:*

$$A_t = \arg\max_a \left[\widehat{\mu}_{a,t-1} + \sqrt{\frac{2\log(1/\delta)}{N_{t-1}(a)}} \right]. \tag{12.3}$$

We denote the RHS of the above for each a as $\mathsf{UCB}(a,t)$ as shorthand; then, we have $A_t = \arg\max_a \mathsf{UCB}(a,t)$. Above, δ is a parameter that will dictate the width of the confidence interval, as well as the probability that the true mean lies within the interval: we will discuss this more at length shortly.

12.2.1 Mathematical principle: incentivizing exploration *and* exploitation

It is instructive to look at Equation (12.3) and ask what factors would lead to the objective becoming large for a given arm $a \in \{1,\ldots,K\}$. Essentially, there are two factors:

- *Large sample mean:* the larger the values of $\widehat{\mu}_{a,t-1}$, the larger the objective will be. This term encourages exploitation.
- *Small number of samples thus far:* notice the inverse dependence on $N_{t-1}(a)$. If this is small, i.e. arm a has been sampled very few times until now, it makes sense to increase the objective to *incentivize exploration.*

Clearly, the choice of δ crucially determines the operating point on the exploration-exploitation tradeoff. It is instructive to consider two extremes:

- The case where $\delta = 1$, which yields the greedy algorithm.
- The case where $\delta = 0$, which can be verified to do pure exploration (as only the second term will always dominate).

Thus, larger values of δ would encourage exploitation, while lower values of δ would encourage exploration. We will now see a statistical interpretation of the value of δ through the notion of *confidence intervals*.

12.2.2 Cognitive principle: optimism in the face of uncertainty

There is also an interesting cognitive principle at work with the UCB algorithm, which is the principle of *optimism in the face of uncertainty*: when we don't know much about an action, we take the *upper-confidence-bound*, rather than the *lower-confidence-bound*, in a display of optimism. There is some preliminary evidence that humans tend to make their decisions in this way in the face of uncertainty (but this remains a matter of extensive debate among cognitive scientists and psychologists).

To see this picture at work, we now consider the value of the hyperparameter δ and its connection to the notion of a confidence interval. We discuss this *informally* here, and briefly touch upon formal caveats at the end. Suppose that we had seen n samples of arm a before round t. Then, an application of Hoeffding's lemma tells us that

$$\begin{aligned}\mathbb{P}\left[\widehat{\mu}_{a,t-1} - \mu_a > \sqrt{\frac{2\log(1/\delta)}{n}}\right] &\leq \exp\left(-\frac{n \cdot 2\log(1/\delta)}{2n}\right) \\ &= \delta.\end{aligned}$$

This tells us that with probability at least $1 - \delta$, our sample mean of arm a would would not be *too* much larger than the true mean given by μ_a. This is called a $(1-\delta)$-*confidence interval*, and essentially is the reason why the UCB algorithm is named so! Furthermore, the extent of closeness decays with the number of times the arm is sampled, n, in a $1/\sqrt{n}$ fashion. In other words, the $(1-\delta)$-confidence intervals *shrink in width* as more samples are drawn of a particular arm.

This formalizes the notion of *confidence intervals*, and will be important for the proof technique that we will introduce in the next lecture. Moreover, it lends interpretational value to the hyperparameter δ of choice. If $\delta = 1$, all bets are off: the confidence widths shrink to 0, and we cannot guarnatee anything! This constitutes the overly high-risk "greedy" approach that purely uses the sample means. On the other hand, if $\delta = 0$, we want to be excessively certain about the sample means and have confidence widths that accommodate the true means with probability 1! This will never be possible unless we make the widths arbitrarily large, and turns out to lead to a very conservative approach of over-exploring.

12.3. Demonstration of performance

Next lecture, we will see how to set this parameter δ, and also show a remarkable ability of the UCB algorithm to automatically tailor the tradeoff between exploration and exploitation to the instance. We spend the rest of this lecture *demonstrating* the performance of UCB

on various random instances. In these notes, Figure 12.1 shows a snapshot of one such execution of the demo on a random 5-armed bandit instance.

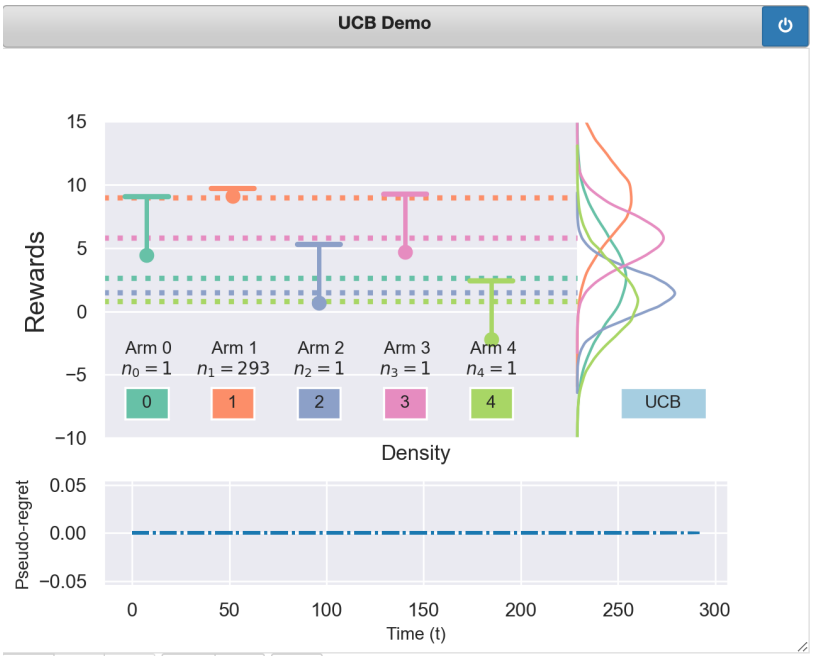

Figure 12.1: Snapshot of demo of UCB algorithm with parameter $\delta = 1/T^2$ (we will explain this choice next lecture) on a randomly chosen 5-armed bandit instance. Here, arm 1 is the best arm and is pulled disproportionately often by UCB. This demo is borrowed from the lectures in UC Berkeley's DS102 course, Fall 2019 iteration, and was originally created by graduate student Karl Krauth.

The demo shows consistently nice behavior with UCB: on all cases, it finds the optimal arm fairly quickly and while it does sample one or two of the suboptimal arms once in a while, it does so very rarely. It in fact turns out that we can show that the pseudo-regret of UCB is given by

$$\overline{R}_T = \mathcal{O}\left(\sum_{a \neq a^*} \frac{\log T}{\Delta_a}\right),$$

where we define suboptimality gaps $\Delta_a := \mu^* - \mu_a$. Now, how do we turn our observations into an analysis that reflects this? The discussion in Section 12.2.2 offers some initial hints: we will, repeatedly, use the fact that the sample means are concentrated around the true means in a way that depends on the number of times the arm has been sampled thus far. In particular, we notice that across all the random examples that we demonstrate in class, the following patterns present themselves:

- For the optimal arm, the *true* mean μ^* is always contained within its confidence interval; in other words, regardless of how many rounds have been played, we always

have $\mu^* < \mathsf{UCB}(a^*, t)$. Thus, even when the optimal arm has been sampled many times, its UCB always lies slightly above its *true mean*. This property is clearly visible in Figure 12.1 even after 293 samples of the optimal arm (1 in this example), and will prove to be useful to analyze UCB.

- After suboptimal arms have been sampled a minimal number of times, their *upper confidence bounds* tend to lie below the true mean of the optimal arm μ^*. This minimal number of times tends to depend on how big the gap was between μ^* and μ_a in the first place: the larger the gap, the fewer number of times arm a needs to be sampled before we observe this behavior. You can see this manifest in the extreme example in Figure 12.1: arms 2 and 4 are more suboptimal than arms 3 and 0 with respect to the optimal arm 1 — so even after just 1 pull, their UCB's will be below the optimal arm mean μ_1. This is not the case for arms 3 and 0, which will require more pulls to be conclusively ruled out.

In fact, this is essentially the reason for why regret tends to have an inverse dependence on the suboptimality gap between arms. Next lecture, we will recap these initial ideas, and use them to show formally that UCB achieves this regret guarantee.

12.4. Additional notes (brief)

- Chapters 4 and 7 of Lattimore and Szepesvári (2020) are an excellent reference for this lecture and the next. Their bibliographical notes are also worth a read to get a sense of the history of the UCB, explore-then-commit, and ϵ-greedy algorithms.
- Chapter 5 of Lattimore and Szepesvári (2020), in addition to our review on probability in the first week of this course, are also worth brushing up before Wednesday's lecture.
- The metric of pseudo-regret implicitly assumes that decision-makers want to maximize their *expected* reward or utility. As we know from behavioral economics, this is not always a reasonable model for human behavior: humans may be risk-seeking or risk-averse, and also use reference points for their decisions. Developing a theory of online decision-making for these more complex behavioral models is a fascinating topic, but outside the scope of this course.

References

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.