Introduction to Stochastic Multi-Armed Bandits

Cynthia Rudin (with Stefano Tracá and Tianyu Wang)

The name "multi-armed bandit" (MAB) comes from the name of a gambling machine. You can choose one of the arms (levers) of the machine at each round, and get a reward based on which arm you choose. The rewards for each arm are iid from a distribution, and each arm has its own distribution. If one of the arms is better than the rest it would be good to always pull that arm, but you don't know which one it is! So you need to divide your time between *exploring* arms that you think might be good with *exploiting* arms that you know are good.

There are many applications for MAB, including recommender systems. For instance, the New York Times uses MAB to determine which news articles to show you on your cellular phone. One could also argue that contextual MAB are the algorithms that are leading to the demise of our current society as we know it! These are the algorithms that are really good at figuring out how to serve you advertisements and social media posts that you are most likely to click on. These are some of the algorithms that can keep you addicted to social media, going down its rabbit holes. But they are actually very simple optimization algorithms that are also used for many scientific purposes and clinical trials and so on. They are probably not dangerous unless you are running a big social media company! I am hopeful that some of you, after learning how these algorithms work, will figure out how to adapt them for the good of society, to optimize long term health of humanity rather than simply using them to optimize clicks and viewing time! One could envision many ways to do this, for instance, prioritizing articles that are more likely to be truthful and less likely to incite rage. Or perhaps to promote articles that encourage educational topics (broadly construed).

Usually MAB is considered to be an alternative to massive A-B testing. Say you want to optimize the look of your website, but there are many possible website options to consider. To determine which one is the best, you might try each option several times and perform pairwise hypothesis testing between all pairs (this is a hypothesis test between option "A" and option "B," hence the terminology "A-B testing"). This will take a huge amount of time, so you might want to run a MAB instead, which conducts all the tests at once, eliminating testing options that are bad once we are pretty sure they are bad, and focusing on options that might be the best. Clinical trials also can use MAB. We can give many different drugs to the patients, and we can use MAB to find the best drugs, based on the performance of these drugs over the course of the trial, without having to do pairwise tests and waste our resources testing drugs that we know early on are not performing well.

In contextual MAB, we also consider the context of each trial. So, for instance, when the social media companies optimize which advertisement to show you, they might not just use information about the general popularity of each ad, they would also use a context vector (a feature vector) that they created about you (e.g., this person is an introvert, who likes machine learning, Dungeons and Dragons, Minecraft, and is a student at a prestigious university in NC, with a political stance that leans to the left, who stays up late looking at dating sites – yes they have that level of detail about you, and no, it is not too hard to figure that information out if they know what you do online).

Formally, the stochastic multi-armed bandit problem is a game played in n rounds. At each round t the player chooses an action among a finite set of m possible choices called arms. When arm j is played ($j \in \{1, \cdots, m\}$) a random reward $X_j(t)$ is drawn from an unknown distribution. In the case of online advertising, the reward is often whether someone clicked on something. The distribution of $X_j(t)$ does not change with time (the index t is just used to indicate in which turn the reward was drawn). At the end of each turn the player can update her estimate of the mean reward of arm j:

$$\widehat{X}_{j,t} = \frac{1}{T_j(t-1)} \sum_{s=1}^{t-1} X_j(s) \mathbb{1}_{\{I_s = j\}}, \tag{1}$$

where $T_j(t-1)$ is the number of times arm j has been played before round t starts, and $\mathbb{1}_{\{I_t=j\}}$ is an indicator function equal to 1 if arm j is played at time t (otherwise its value is 0). After a while, this empirical mean will be close to the arm's mean reward. Updating these estimates after each round will help the player in choosing a good arm in the next round.

At each turn, the player suffers a possible regret from not having played the best arm. If they had chosen the best arm, their reward would have been $X^*(t)$ where notation * means the best arm. The total regret at the end of the game is given by

$$R_n^{(\text{raw})} = \sum_{t=1}^{n} \sum_{j=1}^{m} [X^*(t) - X_j(t)] \mathbb{1}_{\{I_t = j\}},$$

where $X^*(t)$ is the reward of the best arm at time t if it would have been played at time t.

We don't usually define the regret this way though when doing theory. We usually assign the regret to be based on the means of the arms distributions. So let's try it again:

$$R_n = \sum_{t=1}^{n} \sum_{j=1}^{m} [\mu_* - \mu_j] \mathbb{1}_{\{I_t = j\}},$$

where μ_j is the expected payoff of arm j. The mean regret for having played arm j is given by $\Delta_j = \mu_* - \mu_j$, where μ_* is the mean reward of the best arm and μ_j is the mean reward obtained when playing arm j. So the regret is now:

$$R_n = \sum_{t=1}^{n} \sum_{j=1}^{m} \Delta_j \mathbb{1}_{\{I_t = j\}},$$

The strategies presented in the following sections aim to minimize the expected cumulative regret $\mathbb{E}[R_n]$, where the expectation is over the random draw of the arms. (The algorithm reacts to these random draws, so the choice of arms I_t also then becomes random.)

$$\mathbb{E}[R_n] = \mathbb{E} \sum_{t=1}^{n} \sum_{j=1}^{m} \Delta_j \mathbb{1}_{\{I_t = j\}} = \sum_{j=1}^{m} \Delta_j \mathbb{E}\left[T_j(n)\right], \tag{2}$$

where $T_j(n)$ is the number of times arm j is played up to time n.

A complete list of the symbols used can be found in Appendix D.

1 ε-greedy algorithm

The first algorithm we consider is called ε-greedy, and it is in Algorithm 2. The idea is very simple: with some small probability, play an arm uniformly at random. Otherwise, pick the arm that we think is the best.

Algorithm 2: ε-greedy algorithm

Input : number of rounds n, number of arms m, a constant k such that $k > \max\{10, \frac{4}{\min_j \Delta_j^2}\}$, sequence $\{\varepsilon_t\}_{t=1}^n = \min\left\{1, \frac{km}{t}\right\}$

Initialization: play all arms once and initialize $\widehat{X}_{j,t}$ (defined in (1)) for each $j = 1, \cdots, m$

for $t = m+1$ **to** n **do**

With probability ε_t play an arm uniformly at random (each arm has probability $\frac{1}{m}$ of being selected), otherwise (with probability $1 - \varepsilon_t$) play ("best") arm j such that

$$\widehat{X}_{j,t-1} \geq \widehat{X}_{i,t-1} \ \forall i.$$

Get reward $X_j(t)$;

Update $\widehat{X}_{j,t}$;

end

You will notice that there are some interesting terms in the algorithm, defining the choice of ε_t. They are chosen that way so that we can get a tight bound on the regret of the algorithm.

Since we select k to be larger than both 1 and $\frac{4}{\min_j \Delta_j^2}$, it means that the algorithm explores for a while before it does any exploitation. To see this, by the definition of ε_t in the algorithm, $\varepsilon_t = 1$ until km/t is less than 1, which takes a while when k is large. In these early iterations t when $\varepsilon_t = 1$, the algorithm just plays arms uniformly at random. As I mentioned earlier, the whole idea of these MAB algorithms is to balance between exploration of arms to reduce uncertainty and exploiting arms that we know are good. So it's useful to explore for a while to know which arms are good before trying to exploit.

Theorem 1.1 shows that the regret of ε-greedy is bounded by a quantity that is at most logarithmic in n. You can see this because the bound consists of a sum of n terms (a term for each t), each of which is less than t^{-1}, and $\log n$ would be a bound on the sum (integral) of these terms. Specifically, ε_t is order $\theta\left(1/t\right)$, while the $\beta_j(t)$ term is $o\left(1/t\right)$. To see this, you need the assumptions we made about ε_t, for instance that $k > 10$ so that the first exponent of β_j is sufficiently negative, and that

$k > 4/\Delta_j^2$ for all j so that the second exponent of β_j is sufficiently negative.

Theorem 1.1 (Regret-bound for ε-greedy algorithm – adapted from Auer et al. (2002))**.** *The bound on the mean regret* $\mathbb{E}[R_n]$ *at time* n *is given by*

$$\mathbb{E}[R_n] \leq ekm^2 + \sum_{t=e^2km+1}^{n} \sum_{j:\mu_j<\mu_*} \Delta_j \left(\varepsilon_t \frac{1}{m} + (1-\varepsilon_t)\beta_j(t) \right) \tag{3}$$

where

$$\beta_j(t) = k \left(\frac{t}{mke} \right)^{-\frac{k}{10}} \log \left(\frac{t}{mke} \right) + \frac{4e^{\frac{1}{2}}}{\Delta_j^2} \left(\frac{t}{mke} \right)^{-\frac{k\Delta_j^2}{4}}. \tag{4}$$

The first term in (3) is a bound on mean regret during the "starting phase" of Algorithm 1. For the rounds after the starting phase, the quantity in the parenthesis of (3) is an upper bound on the probability of playing arm j. In the bound, $\beta_j(t)$ is an upper bound on the probability that our algorithm thinks arm j is the best arm at round t, and $1/m$ is the probability of choosing arm j when the choice is made at random. Proof in Appendix A.

2 The UCB algorithm

The UCB algorithm is also very simple. It creates a confidence interval on the mean reward. At each round, it chooses the arm with the highest upper confidence interval. This is because any arm with a high upper confidence bound could be the best arm.

Algorithm 3: UCB algorithm

Input : number of rounds n, number of arms m
Initialization: play all arms once and initialize $\widehat{X}_{j,t}$ (as defined in (1)) for each $j = 1, \cdots, m$
for $t = m+1$ **to** n **do**
  play arm j with the highest upper confidence bound on the mean estimate:

$$\widehat{X}_{j,t-1} + \sqrt{\frac{2\log(t)}{T_j(t-1)}};$$

  Get reward X_j;
  Update $\widehat{X}_{j,t}$;
end

The bound for UCB also grows logarithmically in n. Again, this can be seen because the terms in the sum decay faster that $1/t$.

Theorem 2.1 (Regret-bound of the UCB algorithm – adapted from Auer et al. (2002))**.** *The bound on the mean regret* $\mathbb{E}[R_n]$ *at time* n *is given by*

$$\mathbb{E}[R_n] \leq \sum_{j=1}^{m} \Delta_j + \sum_{j:\mu_j<\mu_*} \frac{8}{\Delta_j} \log(n) + \sum_{j=1}^{m} \Delta_j \left(1 + \sum_{t=m+1}^{n} 2t^{-4}(t-1-m)^2 \right). \tag{5}$$

Proof in Appendix B.

3 Instance-independent regret bound

In previous sections, we have discussed regret bounds of the ϵ-greedy and the UCB algorithm, which are logarithmic in number of steps – n. In fact, such logarithmic regret rate is the best we can hope for *in the asymptotic sense* (Lai and Robbins, 1985). Yet it is important to point out that the aforementioned bounds depend on Δ_j (the gap of expected reward of the optimal arm and arm j). This means different problem instances exhibit different regret rates. In particular, when the optimality gaps Δ_j is small, the regret rate may be large. A natural question to ask is: is there a form of regret bound that hold true for any problem instance? The answer is positive.

Corollary 3.1. *Fix an arbitrary positive integer n. For the UCB algorithm, the mean regret satisfies*

$$\mathbb{E}\left[R_n\right] \leq \sqrt{\sum_{j:\Delta_j>0}\left(8\log(n)+1+2\sum_{t=m+1}^{n} t^{-4}(t-1-m)^2\right)}\sqrt{n}, \tag{6}$$

where m is number of arms. For the ϵ-greedy algorithm, the mean regret satisfies

$$\mathbb{E}\left[R_n\right] \leq \sqrt{\sum_{j:\Delta_j>0}\left(k\left(\frac{t}{mke}\right)^{-\frac{k}{10}}\log\left(\frac{t}{mke}\right)+4e^{\frac{1}{2}}\left(\frac{t}{mke}\right)^{-\frac{k\Delta_j^2}{4}}\right)}\cdot\sqrt{n}, \tag{7}$$

where m is the number of arms, and k is an algorithm parameter specified in Algorithm 2.

With proofs for Theorems 1.1 and 2.1, the proof of the above corollary is fairly straightforward, and is in Appendix C. Its counterpart for the ϵ-greedy algorithm follows the same procedure.

In fact, the bound in Corollary 3.1 for ϵ-greedy and the UCB algorithm are also optimal *in the worst case sense*. In fact, the rate in Corollary 3.1 is worst-case optimal. The proof for asymptotic optimality (Lai and Robbins, 1985) uses a similar (but a bit more involved) mechanism, and is left as optional reading. Textbooks covering this topic include those of Bubeck and Cesa-Bianchi (2012); Slivkins (2019); Lattimore and Szepesvári (2020).

4 Contextual Bandits

A natural extension to the standard multi-armed bandit is to make decisions with "side information." In this setting, at each time, the agent observes some contextual information, and chooses an arm based not only on the arms' histories, but also on the contextual information.

This setting is used in important real-world scenarios. For example, in online item recommendation, a company may use the context to describe the browsing user (age, gender, geographical location, etc.) and construct a feature vector to be the user's profile.

$$\text{user\_in\_context} = [1 \text{ if introvert},\ 1 \text{ if likes Jazz},\ 1 \text{ it is now between 12am and 6am},\ 1 \text{ if browsing dating sites}, ...]$$

After observing this context vector, an algorithm (the recommendation system) chooses an arm (item to recommend, e.g., a video game ad) to display the to the user. The choice of which arm would be more desirable really can depend on the context: what the user wants after midnight could be totally different than what she wants at lunchtime! Also, different people respond to very different ads, based on their interests. The algorithm could get a reward of 1 if the users clicks on the item, or a zero reward otherwise. Or, the reward could be "conversion," that is, whether the user purchased the item that is being advertised.

We will formulate (a preliminary version of) the problem, define a performance metric, and present one algorithm for this problem (out of many possible algorithms). The problem is described by a set of contexts $\mathcal{C} := [0,1]^{d_\mathcal{S}}$ ($d_\mathcal{S}$ is the dimension of the context vector), a continuous set of arms $\mathcal{A} := [0,1]^{d_\mathcal{A}}$ ($d_\mathcal{A}$ is the dimension of the arm space), and a (stochastic) reward $f(z,a)+\epsilon,\ \forall(z,a)\in\mathcal{C}\times\mathcal{A}$, where f is the mean reward function, and ϵ is a zero-mean and bounded noise. In addition, we assume that the "mean reward" is Lipschitz: For all $(z,a),(z',a')\in\mathcal{C}\times\mathcal{A}$, we assume $|f(z,a)-f(z',a')|\leq L\|(z,a)-(z',a')\|_2$, where L is the Lipschitz constant of the mean reward. For time $t=1,2,\cdots,n$, the environment reveals a context z_t, the agent chooses an arm a_t, and receives a random reward $X_t=f(z_t,a_t)+\epsilon$. (It's confusing, but now X_t is the label, whereas z_t and a_t act like the features!) In other words, the mean regret changes smoothly over context-arm space. This smoothness is precisely what allows us to interpolate from past situations to the present situation. Perhaps we have seen many contexts that are similar to the current context (but not exactly the same), and we know what arms were shown in previous

contexts. We can use this information to estimate rewards, even to a new arm and new context we have never seen before, as long as they are similar to past cases.

(Note that if the set of arms is finite, then we need the rewards only to be Lipschitz with respect to the context, and we can look at the past history of that particular arm. In fact, there are many variations of this problem!)

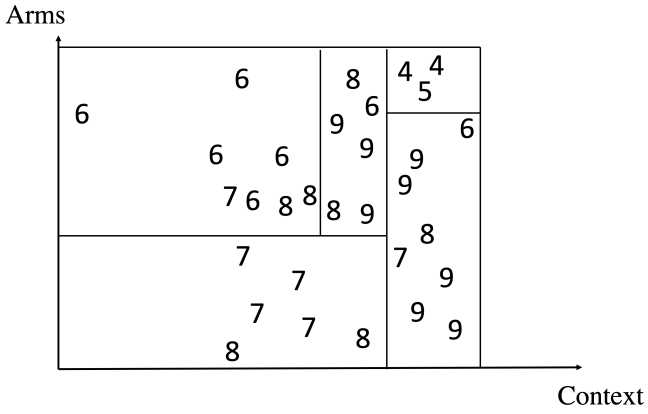
The performance is now measured by *contextual regret*:

$$R_n := \sum_{t=1}^{n} \left(\max_{a \in \mathcal{A}} f(z_t, a) - f(z_t, a_t) \right). \tag{8}$$

This is how much you lost in rewards from choosing the arm a_t in context z_t, rather than choosing the best arm for that context.

The UCB algorithm can be altered to solve contextual bandit problems with slight modification in order to accommodate context. To do this, we can create parametric models that estimate $f(z_t, a_t)$ with confidence intervals. You can use almost any kind of model to estimate f as long as it has an upper confidence bound. For instance, you can use decision trees that group the past contexts and arms we have seen so far into leaves of a tree. Then you can calculate the upper confidence bound of rewards that fall into that leaf. To use UCB, you just need to be able to compute upper confidence bounds for the value of f at any given context point z_t.

Let us provide a simple contextual bandit algorithm that iteratively partitions the context and arm space into leaves of a tree. Over iterations, we maintain a partition $\mathcal{P}_t$ of the context-arm space, and define a mean and confidence with respect to each "leaf" in this partition. This partition could be learned using decision tree splitting, but one could also define it in other ways. We would typically have the partition grow finer and finer as we gather more information about contexts, arms and rewards. In the figure below, we show an example of a partition that we could have constructed using a decision tree. Each number in the figure represents a trial, where the number's position on the plot represents the context and arm that was pulled, and the value (e.g., 4,5,6,7,8,9) is the reward that was obtained after pulling the arm.



Given partition $\mathcal{P}_t$, function $p_t : \mathcal{C} \times \mathcal{A} \to \mathcal{P}_t$ is called a Region Selection Function with respect to $\mathcal{P}_t$ if for any $(z, a) \in \mathcal{C} \times \mathcal{A}$, $p_t(z, a)$ is the region in $\mathcal{P}_t$ containing (z, a). In other words the Region Selection Function tells us which region ("bin") a given context-arm pair is in. With this function defined, we now define the mean estimate and confidence bound in each partition bin. Let $\{(z_1, a_1), x_1, (z_2, a_2), x_2, \cdots, (z_t, a_t), x_t\}$ be the (context-arm, reward) observations up to time t. Define function n_t to be the count of points (each corresponding to a historical arm pull) in the same bin as context-arm pair (z, a) (but if there are no points, we count that as 1):

$$n_t(z, a) = \max \left\{ 1, \sum_{s=1}^{t} \mathbb{1}_{\{(z_s, a_s) \in p_t(z,a)\}} \right\}. \tag{9}$$

Define m_t as the mean of the rewards in the bin:

$$m_t(z, a) = \begin{cases} \frac{\sum_{s=1}^{t} x_s \mathbb{1}_{\{(z_s, a_s) \in p_t(z,a)\}}}{n_t(z,a)}, \text{ if } \sum_{s=1}^{t} \mathbb{1}_{\{(z_s, a_s) \in p_t(z,a)\}} > 0; \\ \infty, \quad \text{otherwise (\textit{no observations in bin } } p_t(z, a)). \end{cases} \tag{10}$$

Now we can define the "UCB index" (that is, the upper confidence bound) for this problem: $\forall (z, a) \in \mathcal{C} \times \mathcal{A}$,

$$I_t(z, a) = m_t(z, a) + C_{UCB} \sqrt{\frac{\log t}{n_t(z, a)}} + \mathcal{D}_L(p_t(z, a)), \tag{11}$$

where C_{UCB} is a parameter that controls how much we would like to explore, because it scales the upper confidence bound and thus controls exploration. Here, $\mathcal{D}_L(p_t(z,a))$ describes how large the region $p_t(z,a)$ is: $\mathcal{D}_L(p_t(z,a)) := L \max_{w,w' \in p_t(z,a)} \|w - w'\|_2$. This last term is useful due to the fact that the reward is smooth (Lipschitz). As long as the reward is smooth and the region is not too large, the function f must stay relatively constant in the region $p_t(z,a)$. In that case, we have a tighter UCB because we are more confident that our estimate of the mean reward represents all of the points in the bin. However, if the region is large, then even if we have estimated the mean and its UCB correctly, there could be significant variation in the mean reward $f(z,a)$ over this large bin. In that case, we raise the upper confidence bound based on the diameter of the bin. This strategy is summarized below. Please see Wang et al. (2020) (and references therein) for more information on (contextual) Lipschitz bandits. When we update the partition, we create more partition bins by splitting

Algorithm 4: UCB-Tree algorithm

Input : number of rounds n, arms $\mathcal{A}$, exploration parameter C_{UCB}, Lipschitz constant L, tree fitting (partition maintenance) rule $\mathcal{R}$.

for $t = 1$ **to** n **do**

Observe context z_t.;

Compute m_t and n_t using (10) and (9);

Play arm a_t with the highest upper confidence bound index: $\max_a$ UCB where $UCB = m_t(z_t,a) + C_{UCB}\sqrt{\frac{\log t}{n_t(z_t,a)}} + \mathcal{D}_L(p_t(z_t,a))$;

Get reward x_t;

Update partition $\mathcal{P}_t$ using rule $\mathcal{R}$.

end

them. We split bins when we have enough data that we would get a sufficiently good estimate of the mean reward in each new bin, after we split. Since we will mostly be choosing bins with very high mean reward, this sequential splitting step will let us zoom in on the arms that have the highest rewards for each context.

5 Other types of bandit problems

There are a huge variety of bandit problems. For instance, there are *sleeping bandits*, where the bandits disappear for a while and then reappear (think about an online sale that appears and disappears), there are *mortal bandits* where the bandits appear at various times and disappear and never come back! (Here, you can think about news articles.) There are bandits where the rewards are delayed (so you're playing blindly for a while when you start playing a new arm). There are bandits where arms lock for a period of time, so that if you choose an arm, you can't change it for a while (think about pricing items online where you are not allowed to change the price too often or it would frighten the customers away).

Some of the most interesting MAB problems involve non-stationary time series, where the expected reward of an arm changes over time. This happens a lot in reality, for instance, demand for many products has a weekly or yearly cycle, with a spike in demand for Christmas!

A Regret-bound of the ε-greedy algorithm

The mean regret at round n is given by

$$R_n = \sum_{t=1}^{n} \sum_{j=1}^{m} \Delta_j \mathbb{1}_{\{I_t = j\}},$$

where $\mathbb{1}_{\{I_t=j\}}$ is an indicator function equal to 1 if arm j is played at time t (otherwise its value is 0) and $\Delta_j = \mu^* - \mu_j$ is the difference between the mean of the best arm's reward distribution and the mean of the j's arm reward distribution. By taking the expectation we have that

$$\mathbb{E}[R_n] = \sum_{t=1}^{n} \sum_{j=1}^{m} \Delta_j \mathbb{P}(\{I_t = j\})$$

which can be rewritten as

$$\mathbb{E}[R_n] \quad = \quad \sum_{t=1}^{n} \sum_{j=1}^{m} \Delta_j \left[\varepsilon_t \frac{1}{m} + (1 - \varepsilon_t) \mathbb{P}(\widehat{X}_{j,T_j(t-1)} \geq \widehat{X}_{i,T_i(t-1)} \ \ \forall i) \right], \tag{12}$$

where notation $\widehat{X}_{i,T_i(t-1)}$ is the estimated mean for arm i after it has been chosen $T_i(t-1)$ times up to time $t-1$. The first term is the probability that we choose arm j by exploring. We explore with probability ε_t and if we explore, we chose j at random, that is, with probability $1/m$. If we chose j while exploiting, which happens with prob $1-\varepsilon_t$, then its average reward is above that of all the other arms.

For this proof, we assume the rewards are bounded, say between 0 and 1. If they are bounded by something bigger than 1, we would have an extra constant scaling factor in the theorem.

STEP 1: Conditions when we think arm j is the best at time t. If we think arm j is the best at time t, then either we overestimated its mean reward, or we underestimated the reward of the best arm, which is called arm $*$. If neither of those things occurred, arm j's rewards would have been below those of arm $*$ and thus we would not think that arm j is the best when it isn't. In the first inequality below, we consider the probability arm j has average reward above all the other arms, and this is less than the probability that arm j has reward greater than just one of those arms (in particular, arm $*$).

$$\begin{aligned}\mathbb{P}(\widehat{X}_{j,T_j(t-1)} \geq \widehat{X}_{i,T_i(t-1)} \ \forall i) &\leq \mathbb{P}(\widehat{X}_{j,T_j(t-1)} \geq \widehat{X}_{*,T_*(t-1)}) \\ &\leq \mathbb{P}\left(\widehat{X}_{j,T_j(t-1)} \geq \mu_j + \frac{\Delta_j}{2}\right) + \mathbb{P}\left(\widehat{X}_{*,T_*(t-1)} \leq \mu_* - \frac{\Delta_j}{2}\right), \end{aligned} \tag{13}$$

where the last inequality follows from the fact that either we must have underestimated arm $*$ or overestimated arm j:

$$\left\{\widehat{X}_{j,T_j(t-1)} \geq \widehat{X}_{*,T_*(t-1)}\right\} \subset \left(\left\{\widehat{X}_{*,T_*(t-1)} \leq \mu_* - \frac{\Delta_j}{2}\right\} \cup \left\{\widehat{X}_{j,T_j(t-1)} \geq \mu_j + \frac{\Delta_j}{2}\right\}\right).$$

Aside: To show this, suppose that there exist an event $\omega \in \left\{\widehat{X}_{j,T_j(t-1)} \geq \widehat{X}_{*,T_*(t-1)}\right\}$ that does not belong to $\left(\left\{\widehat{X}_{*,T_*(t-1)} \leq \mu_* - \frac{\Delta_j}{2}\right\} \cup \left\{\widehat{X}_{j,T_j(t-1)} \geq \mu_j + \frac{\Delta_j}{2}\right\}\right)$. Then, we would have that

$$\begin{aligned}\omega &\in \left(\left\{\widehat{X}_{*,T_*(t-1)} \leq \mu_* - \frac{\Delta_j}{2}\right\} \cup \left\{\widehat{X}_{j,T_j(t-1)} \geq \mu_j + \frac{\Delta_j}{2}\right\}\right)^C \\ &= \left\{\widehat{X}_{*,T_*(t-1)} > \mu_* - \frac{\Delta_j}{2}\right\} \cap \left\{\widehat{X}_{j,T_j(t-1)} < \mu_j + \frac{\Delta_j}{2}\right\}, \end{aligned} \tag{14}$$

but from the intersection of events given in (14) it follows that $\widehat{X}_{*,T_*(t-1)} > \mu_* - \frac{\Delta_j}{2} = \mu_j + \frac{\Delta_j}{2} > \widehat{X}_{j,T_j(t-1)}$ which contradicts $\omega \in \left\{\widehat{X}_{j,T_j(t-1)} \geq \widehat{X}_{*,T_*(t-1)}\right\}$. Therefore, all events where $\left\{\widehat{X}_{j,T_j(t-1)} \geq \widehat{X}_{*,T_*(t-1)}\right\}$ belong to the set of events where:

$$\left(\left\{\widehat{X}_{*,T_*(t-1)} \leq \mu_* - \frac{\Delta_j}{2}\right\} \cup \left\{\widehat{X}_{j,T_j(t-1)} \geq \mu_j + \frac{\Delta_j}{2}\right\}\right).$$

STEP 2: Let us bound the probability of overestimating sub-optimal arm j at time t. Let us consider the first term of (13). The computations for the second term are basically identical.

$$\begin{aligned}\mathbb{P}\left(\widehat{X}_{j,T_j(t-1)} \geq \mu_j + \frac{\Delta_j}{2}\right) &= \sum_{s=1}^{t-1} \mathbb{P}\left(T_j(t-1) = s, \widehat{X}_{j,s} \geq \mu_j + \frac{\Delta_j}{2}\right) \\ &= \sum_{s=1}^{t-1} \mathbb{P}\left(T_j(t-1) = s | \widehat{X}_{j,s} \geq \mu_j + \frac{\Delta_j}{2}\right) \mathbb{P}\left(\widehat{X}_{j,s} \geq \mu_j + \frac{\Delta_j}{2}\right) \\ &\leq \sum_{s=1}^{t-1} \mathbb{P}\left(T_j(t-1) = s | \widehat{X}_{j,s} \geq \mu_j + \frac{\Delta_j}{2}\right) e^{-\frac{\Delta_j^2}{2}s}, \end{aligned} \tag{15}$$

where in the last inequality we used the Chernoff-Hoeffding bound. The second term will be small when s is large, so that term will be sufficient to handle whatever the first term brings when s is large. When s is small, the first term could be problematic since it will be large. We are going to separate this sum into large s and small s and handle them separately. Here, small s means less than x_0, where we define it as:

$$x_0 := \frac{1}{2m} \sum_{s=1}^{t} \varepsilon_s.$$

Then

$$(15) \quad \leq \quad \sum_{s=1}^{\lfloor x_0 \rfloor} \mathbb{P}\left(T_j(t-1) = s | \widehat{X}_{j,s} \geq \mu_j + \frac{\Delta_j}{2}\right) \cdot 1 + \sum_{s=\lfloor x_0 \rfloor + 1}^{t-1} 1 \cdot e^{-\frac{\Delta_j^2}{2} s}.$$

Here, we split the sum into two pieces and bounded one of the terms by 1.

Let us work on the second term. We will now use the fact that $\sum_{s=\lfloor x_0 \rfloor+1}^{\infty} e^{-bs} \leq \frac{1}{b} e^{-b\lfloor x_0 \rfloor}$, where in our case $b = \frac{\Delta_j^2}{2}$.

$$(15) \quad \leq \quad \sum_{s=1}^{\lfloor x_0 \rfloor} \mathbb{P}\left(T_j(t-1) = s | \widehat{X}_{j,s} \geq \mu_j + \frac{\Delta_j}{2}\right) + \frac{2}{\Delta_j^2} e^{-\frac{\Delta_j^2}{2} \lfloor x_0 \rfloor}.$$

Now comes a trick. Let us define $T_j^R(t-1)$ as the number of times arm j is played when we are performing exploration. Note that $T_j^R(t-1) \leq T_j(t-1)$ and that $T_j^R(t-1) = \sum_{s=1}^{t-1} B_s$ where B_s is a Bernoulli r.v. with parameter ε_s/m (this is the probability that we explore times the probability that we choose arm j when exploring, so ϵ_s times $1/m$. In that case, $T_j^R(t-1)$ equals a values less than s but we don't know which one. Luckily we're constructing upper bounds. So we add up all possibilities for it.

$$(15) \quad \leq \quad \sum_{s=1}^{\lfloor x_0 \rfloor} \mathbb{P}\left(T_j^R(t-1) \leq s | \widehat{X}_{j,s} \geq \mu_j + \frac{\Delta_j}{2}\right) + \frac{2}{\Delta_j^2} e^{-\frac{\Delta_j^2}{2} \lfloor x_0 \rfloor}.$$

Now things are good, since the number of times we explore to choose arm j, $T_j^R(t-1)$, does not depend on the estimate of the mean for arm j. The number of terms in the sum is $\lfloor x_0 \rfloor$:

$$(15) \quad \leq \quad \lfloor x_0 \rfloor \mathbb{P}\left(T_j^R(t-1) \leq \lfloor x_0 \rfloor\right) + \frac{2}{\Delta_j^2} e^{-\frac{\Delta_j^2}{2} \lfloor x_0 \rfloor}. \tag{16}$$

Recall $T_j^R(t-1) = \sum_{s=1}^{t-1} B_s$ where B_s are independent Bernoulli random variables with $\mathbb{P}(B_s = 1) = \frac{\epsilon_s}{m}$. The Bernstein inequality states, for (independent) Bernoulli random variables,

$$\mathbb{P}\left(\sum_{s=1}^{t-1} B_s \leq \mathbb{E}\left[\sum_{s=1}^{t-1} B_s\right] - a\right) \leq \exp\left(-\frac{\frac{1}{2}a^2}{\sum_{s=1}^{t-1} Var(B_s) + \frac{1}{3}a}\right).$$

Also, we have (using the formula for the variance of a Bernoulli random variable):

$$Var(B_s) = \frac{\varepsilon_s}{m}\left(1 - \frac{\varepsilon_s}{m}\right) \leq \frac{\varepsilon_s}{m}. \tag{17}$$

Thus, applying Bernstein's inequality to the Bernoulli random variables B_s with $a = x_0 = \frac{1}{2}\mathbb{E}\left[T_j^R(t-1)\right]$ gives

$$\begin{aligned}
\mathbb{P}(T_j^R(t-1) \leq \lfloor x_0 \rfloor) &\leq \mathbb{P}(T_j^R(t-1) \leq x_0) \\
&= \mathbb{P}\left(T_j^R(t-1) \leq \mathbb{E}[T_j^R(t-1)] - \frac{1}{2}\mathbb{E}[T_j^R(t-1)]\right) \\
&\leq \exp\left\{-\frac{\frac{1}{8}(\mathbb{E}[T_j^R(t-1)])^2}{\sum_{s=1}^{t-1} Var(B_s) + \frac{1}{6}\mathbb{E}[T_j^R(t-1)]}\right\} \\
&\leq \exp\left\{-\frac{\frac{1}{8}(\mathbb{E}[T_j^R(t-1)])^2}{\sum_{s=1}^{t-1} \frac{\varepsilon_s}{m} + \frac{1}{6}\mathbb{E}[T_j^R(t-1)]}\right\} && \text{(by Eq. 17)} \\
&= \exp\left\{-\frac{\frac{1}{8}(\mathbb{E}[T_j^R(t-1)])^2}{\mathbb{E}[T_j^R(t-1)] + \frac{1}{6}\mathbb{E}[T_j^R(t-1)]}\right\} && \text{(because } \mathbb{E}[T_j^R(t-1)] = \textstyle\sum_{s=1}^{t-1} \frac{\varepsilon_s}{m}.\text{)} \\
&= \exp\left\{-\frac{6}{7} \cdot \frac{1}{8}\mathbb{E}[T_j^R(t-1)]\right\} = \exp\left\{-\frac{3}{7} \cdot \frac{1}{2} x_0\right\} \\
&\leq \exp\left\{-\frac{1}{5} x_0\right\}. && (18)
\end{aligned}$$

STEP 3: To upper bound (18), let us find a lower bound on $\lfloor x_0 \rfloor$. Let us define $n' = \lfloor km \rfloor + 1$ (where k was defined in the algorithm statement, remember that it is more than 10), then

$$\begin{aligned} x_0 &= \frac{1}{2m}\sum_{s=1}^{t}\varepsilon_s \\ &= \frac{1}{2m}\sum_{s=1}^{t}\min\left\{1, \frac{km}{s}\right\} \\ &= \frac{1}{2m}\sum_{s=1}^{n'}1 + \frac{km}{2m}\sum_{s=n'+1}^{t}\frac{1}{s} \\ &= \frac{n'}{2m} + \frac{k}{2}\left(\sum_{s=1}^{t}\frac{1}{s} - \sum_{s=1}^{n'}\frac{1}{s}\right). \end{aligned}$$

Here we will use some properties of harmonic sequences, namely $\sum_{t=1}^{n}\frac{1}{t} \le \log n + 1$ and $\sum_{t=1}^{n}\frac{1}{t} > \int_1^{n+1}\frac{1}{t}dt = \ln(n+1)$. Continuing from the previous line,

$$\begin{aligned} x_0 \ge\ & \frac{n'}{2m} + \frac{k}{2}\left(\log(t+1) - (\log(n') + \log(e))\right) \\ =\ & \frac{k}{2}\frac{n'}{mk} + \frac{k}{2}\left(\log\frac{(t+1)}{n'e}\right) \\ \ge\ & \frac{k}{2}\log\left(\frac{n'}{mk}\right) + \frac{k}{2}\log\left(\frac{t}{n'e}\right) \qquad \text{(because } \log x \le x\text{)} \\ =\ & \frac{k}{2}\log\left(\frac{t}{mke}\right). \end{aligned} \tag{19}$$

Using (19) combined with (18) in (16), we get the following:

$$\begin{aligned} (15) &\le \lfloor x_0 \rfloor \mathbb{P}\left(T_j^R(t-1) \le \lfloor x_0 \rfloor\right) + \frac{2}{\Delta_j^2}e^{-\frac{\Delta_j^2}{2}\lfloor x_0 \rfloor} \qquad \text{(copying (16))} \\ &\le \lfloor x_0 \rfloor \exp\left\{-\frac{1}{5}x_0\right\} + \frac{2}{\Delta_j^2}e^{-\frac{\Delta_j^2}{2}\lfloor x_0 \rfloor} \qquad \text{(from (18))} \\ &\le x_0 \exp\left\{-\frac{1}{5}x_0\right\} + \frac{2}{\Delta_j^2}e^{-\frac{\Delta_j^2}{2}(x_0-1)} \\ &= x_0 \exp\left\{-\frac{1}{5}x_0\right\} + \frac{2}{\Delta_j^2}e^{\frac{1}{2}\Delta_j^2}e^{-\frac{\Delta_j^2}{2}x_0} \\ &\le x_0 \exp\left\{-\frac{1}{5}x_0\right\} + \frac{2}{\Delta_j^2}e^{\frac{1}{2}}e^{-\frac{\Delta_j^2}{2}x_0} \qquad \text{(since } \Delta_j \in [0,1]\text{)} \\ &\le x_0 \exp\left\{-\frac{1}{5}x_0\right\} + \frac{2e^{\frac{1}{2}}}{\Delta_j^2}\left(\frac{t}{mke}\right)^{-\frac{k\Delta_j^2}{4}} \qquad \text{(from (19)).} \end{aligned} \tag{20}$$

Next, from the first order derivative test we know that the function $x_0 \exp\left\{-\frac{1}{5}x_0\right\}$ is decreasing on $[5, \infty)$. Thus when $\frac{k}{2}\log\left(\frac{t}{mke}\right) \ge 5$, (and from (19) we have $x_0 \ge \frac{k}{2}\log\left(\frac{t}{mke}\right)$) we have (plugging in $\frac{k}{2}\log\left(\frac{t}{mke}\right)$ for x_0 in the expression $x_0 \exp\left\{-\frac{1}{5}x_0\right\}$):

$$x_0 \exp\left\{-\frac{1}{5}x_0\right\} \le \frac{k}{2}\left(\frac{t}{mke}\right)^{-\frac{k}{10}}\log\left(\frac{t}{mke}\right). \tag{21}$$

Since we choose $k \ge \max\left\{10, \frac{4}{\min_j \Delta_j^2}\right\}$, we know $\frac{k}{2} \ge 5$. Thus $t \ge e^2 km$ is sufficient to ensure $x_0 \ge \frac{k}{2}\log\left(\frac{t}{mke}\right) \ge 5$. Combining the above results in (20) and (21) gives: when $t \ge e^2 km$,

$$\text{the first term in (13)} = \text{the left side of (15)} \le \frac{k}{2}\left(\frac{t}{mke}\right)^{-\frac{k}{10}}\log\left(\frac{t}{mke}\right) + \frac{2e^{\frac{1}{2}}}{\Delta_j^2}\left(\frac{t}{mke}\right)^{-\frac{k\Delta_j^2}{4}}. \tag{22}$$

STEP 4: Let us bound the probability of underestimating sub-optimal arm j at time t. Since the computations for the second term in (13) are essentially identical, by removing the $1/2$ factor we get this bound on $\mathbb{P}\left(\widehat{X}_{j,T_j(t-1)} \geq \widehat{X}_{i,T_i(t-1)} \ \ \forall i\right)$ (when $t \geq e^2 km$):

$$\beta_j(t) = k\left(\frac{t}{mke}\right)^{-\frac{k}{10}} \log\left(\frac{t}{mke}\right) + \frac{4e^{\frac{1}{2}}}{\Delta_j^2}\left(\frac{t}{mke}\right)^{-\frac{k\Delta_j^2}{4}}. \tag{23}$$

STEP 5: Let us bound the probability of playing suboptimal arm j. We have now an upper bound for

$$\mathbb{P}(\widehat{X}_{j,T_j(t-1)} \geq \widehat{X}_{i,T_i(t-1)} \ \ \forall i),$$

the left hand side of (13). We will plug this into (12) which yields the following bound on the mean regret at time n. First, we just split the sum over t in (12) into two parts.

$$\begin{aligned}\mathbb{E}[R_n] \quad \leq \quad & \sum_{t=1}^{\lfloor ekm \rfloor} \sum_{j=1}^{m} \Delta_j \left[\varepsilon_t \frac{1}{m} + (1-\varepsilon_t)\mathbb{P}(\widehat{X}_{j,T_j(t-1)} \geq \widehat{X}_{i,T_i(t-1)} \ \ \forall i)\right] \\ & + \sum_{t=\lfloor ekm \rfloor + 1}^{n} \sum_{j=1}^{m} \Delta_j \left[\varepsilon_t \frac{1}{m} + (1-\varepsilon_t)\mathbb{P}(\widehat{X}_{j,T_j(t-1)} \geq \widehat{X}_{i,T_i(t-1)} \ \ \forall i)\right],\end{aligned} \tag{24}$$

The first term has an upper bound, since all probabilities are at most 1, and all Δ_j's are at most 1:

$$\begin{aligned}\sum_{t=1}^{\lfloor ekm \rfloor} \sum_{j=1}^{m} \Delta_j \left[\varepsilon_t \frac{1}{m} + (1-\varepsilon_t)\mathbb{P}(\widehat{X}_{j,T_j(t-1)} \geq \widehat{X}_{i,T_i(t-1)} \ \ \forall i)\right] \quad & \leq \quad \sum_{t=1}^{\lfloor ekm \rfloor} \sum_{j=1}^{m} 1 \cdot \left[\varepsilon_t \frac{1}{m} + (1-\varepsilon_t)\cdot 1\right] \\ & \leq \quad \sum_{t=1}^{\lfloor ekm \rfloor} \sum_{j=1}^{m} 1 \\ & \leq \quad ekm^2\end{aligned} \tag{25}$$

Thus, including into (24) the upper bound for the first term (25) and the upper bound for the probability in the second term (23), we obtain:

$$\mathbb{E}[R_n] \quad \leq \quad ekm^2 + \sum_{t=\lfloor e^2 km \rfloor + 1}^{n} \sum_{j:\mu_j < \mu_*} \Delta_j \left(\varepsilon_t \frac{1}{m} + (1-\varepsilon_t)\beta_j(t)\right),$$

This proves the theorem.

B The regret bound of the UCB algorithm

The regret at round n is given by

$$R_n \quad = \quad \sum_{j=1}^{m} \Delta_j + \sum_{t=m+1}^{n} \sum_{j=1}^{m} \Delta_j \mathbb{1}_{\{I_t = j\}}$$

The expected regret $\mathbb{E}[R_n]$ at round n is bounded by

$$\mathbb{E}[R_n] \leq \sum_{j=1}^{m} \Delta_j + \sum_{j=1}^{m} \Delta_j \mathbb{E}[T_j(n)]. \tag{26}$$

where $T_j(n) = \sum_{t=1}^{n} \mathbb{1}_{\{I_t = j\}}$ is the number of times arm j has been chosen up to round n. Recall that

$$\widehat{X}_{j,t} = \frac{1}{T_j(t-1)} \sum_{s=1}^{T_j(t-1)} X_j(s).$$

Let's suppose the rewards are bounded, say between 0 and 1.

STEP 1: Let us bound the probability of overestimating or underestimating suboptimal arm j.
From the Chernoff-Hoeffding Inequality we have that

$$\mathbb{P}\left(\frac{1}{T_j(t-1)}\sum_{i=1}^{T_j(t-1)} X_j(i) - \mu_j \leq -\varepsilon\right) \leq \exp\{-2T_j(t-1)\varepsilon^2\},$$

and

$$\mathbb{P}\left(\frac{1}{T_j(t-1)}\sum_{i=1}^{T_j(t-1)} X_j(i) - \mu_j \geq \varepsilon\right) \leq \exp\{-2T_j(t-1)\varepsilon^2\}.$$

By selecting $\varepsilon = \sqrt{\frac{2\log(t)}{T_j(t-1)}}$ we have

$$\mathbb{P}\left(\widehat{X}_{j,t} + \sqrt{\frac{2\log(t)}{T_j(t-1)}} \leq \mu_j\right) \leq t^{-4}, \tag{27}$$

and

$$\mathbb{P}\left(\widehat{X}_{j,t} - \sqrt{\frac{2\log(t)}{T_j(t-1)}} \geq \mu_j\right) \leq t^{-4}. \tag{28}$$

STEP 2: Let us bound the number of times we play arm j.
For each t, we consider the events such that the UCB of j is higher than that of $*$. These are events that could potentially happen due to the randomness in the draws of each arm at each time until t.

$$\left\{\widehat{X}_{j,T_j(t-1)} + \sqrt{\frac{2\log(t)}{T_j(t-1)}} \geq \widehat{X}_{*,T_*(t-1)} + \sqrt{\frac{2\log(t)}{T_*(t-1)}}, T_j(t-1) \geq u\right\} \subset$$
$$\left\{\max_{s_j\in\{u,\ldots,T_j(t-1)\}} \widehat{X}_{j,s_j} + \sqrt{\frac{2\log(t)}{s_j}} \geq \min_{s_*\in\{1,\ldots,T_*(t-1)\}} \widehat{X}_{*,s_*} + \sqrt{\frac{2\log(t)}{s_*}}\right\} \tag{29}$$

Events on both the left and right sides of (29) are included in

$$\bigcup_{s_*=1}^{T_*(t-1)} \bigcup_{s_j=u}^{T_j(t-1)} \left\{\widehat{X}_{j,s_j} + \sqrt{\frac{2\log(t)}{s_j}} \geq \widehat{X}_{*,s_*} + \sqrt{\frac{2\log(t)}{s_*}}\right\}. \tag{30}$$

Thus, for any integer u, we may write

$$\begin{aligned}
T_j(n) &= 1 + \sum_{t=m+1}^{n} \mathbb{1}\{I_t = j\} \quad \text{(play arm } j \text{ once during the starting phase)} \\
&= u + \sum_{t=m+1}^{n} \mathbb{1}\{I_t = j, T_j(t-1) \geq u\} \quad \text{(split terms to separate out the first } u \text{ turns)} \\
&= u + \sum_{t=m+1}^{n} \mathbb{1}\left\{\widehat{X}_{j,T_j(t-1)} + \sqrt{\frac{2\log(t)}{T_j(t-1)}} \geq \widehat{X}_{*,T_*(t-1)} + \sqrt{\frac{2\log(t)}{T_*(t-1)}}, T_j(t-1) \geq u\right\} \qquad (31) \\
&\qquad \text{(play arm } j \text{ when its UCB is the highest)} \\
&\leq u + \sum_{t=m+1}^{n} \mathbb{1}\left\{\max_{s_j\in\{u,\ldots,T_j(t-1)\}} \widehat{X}_{j,s_j} + \sqrt{\frac{2\log(t)}{s_j}} \geq \min_{s_*\in\{1,\ldots,T_*(t-1)\}} \widehat{X}_{*,s_*} + \sqrt{\frac{2\log(t)}{s_*}}\right\} \\
&\qquad \text{(from (29))} \\
&\leq u + \sum_{t=m+1}^{n} \sum_{s_*=1}^{T_*(t-1)} \sum_{s_j=u}^{T_j(t-1)} \mathbb{1}\left\{\widehat{X}_{j,s_j} + \sqrt{\frac{2\log(t)}{s_j}} \geq \widehat{X}_{*,s_*} + \sqrt{\frac{2\log(t)}{s_*}}\right\} \text{ (from (30)).} \qquad (32)
\end{aligned}$$

STEP 3: Let us rewrite the event of playing arm j as a subset of the union of underestimating or overestimating arm j. When

$$\mathbb{1}\left\{\widehat{X}_{j,t} + \sqrt{\frac{2\,\log(t)}{T_j(t-1)}} \geq \widehat{X}_{*,t} + \sqrt{\frac{2\,\log(t)}{T_*(t-1)}}\right\} \quad \text{(when we play arm } j\text{)} \tag{33}$$

is equal to one, at least one of the following has to be true:

$$\widehat{X}_{*,t} \leq \mu_* - \sqrt{\frac{2\,\log(t)}{T_*(t-1)}}; \quad \text{(we underestimated arm } *\text{)} \tag{34}$$

$$\widehat{X}_{j,t} \geq \mu_j + \sqrt{\frac{2\,\log(t)}{T_j(t-1)}}; \quad \text{(we overestimated arm } j\text{)} \tag{35}$$

$$\mu_* < \mu_j + 2\sqrt{\frac{2\,\log(t)}{T_j(t-1)}}. \quad \text{(arm } j\text{'s UCB is higher than } \mu^*\text{)} \tag{36}$$

To prove this, suppose none of them hold. Then from (34) we would have that $\widehat{X}_{*,t} > \mu_* - \sqrt{\frac{2\log(t)}{T_*(t-1)}}$; then, by applying (36) (with opposite verse since we are assuming it does not hold) we get $\widehat{X}_{*,t} > \mu_j + 2\sqrt{\frac{2\log(t)}{T_j(t-1)}} - \sqrt{\frac{2\log(t)}{T_*(t-1)}}$ and then from (35) (again, with opposite verse) follows that $\widehat{X}_{*,t} > \widehat{X}_{j,t} + \sqrt{\frac{2\log(t)}{T_j(t-1)}} - \sqrt{\frac{2\log(t)}{T_*(t-1)}}$ which is in contradiction with (33). Now, if we set $u = \left\lceil \frac{8}{\Delta_j^2}\log(t)\right\rceil$, then for $T_j(t-1) \geq u$, we have seen arm j enough times that its confidence bound is less than Δ, which allows us to show that (36) does not hold, shown as follows:

$$\begin{aligned}
\mu_* - \mu_j - 2\sqrt{\frac{2\log(t)}{T_j(t-1)}} &\geq \mu_* - \mu_j - 2\sqrt{\frac{2\log(t)}{u}} \\
&= \mu_* - \mu_j - 2\sqrt{\frac{2\log(t)}{\left\lceil \frac{8}{\Delta_j^2}\log(t)\right\rceil}} \\
&\geq \mu_* - \mu_j - 2\sqrt{\frac{2\log(t)}{\frac{8}{\Delta_j^2}\log(t)}} \\
&\geq \mu_* - \mu_j - \Delta_j = 0,
\end{aligned}$$

therefore, with this choice of u, (36) cannot hold. So either (34) or (35) is true if we play arm j instead of arm $*$.

STEP 4: Let us bound the expected number of times we play arm j.
Using (32) and Step 3, we have that

$$\begin{aligned}
T_j(n) \leq & \left\lceil \frac{8}{\Delta_j^2}\log(n)\right\rceil \quad \text{(this is } u \text{ in (32))} \\
&+ \sum_{t=m+1}^{n}\sum_{s_*=1}^{T_*(t-1)}\sum_{s_j=u}^{T_j(t-1)} \mathbb{1}\left\{\widehat{X}_{*,s_*} \leq \mu_* - \sqrt{\frac{2\log(t)}{s_*}}\right\} \quad \text{(from (34))} \\
&+ \sum_{t=m+1}^{n}\sum_{s_*=1}^{T_*(t-1)}\sum_{s_j=u}^{T_j(t-1)} \mathbb{1}\left\{\widehat{X}_{j,s_j} \geq \mu_j + \sqrt{\frac{2\log(t)}{s_j}}\right\} \quad \text{(from (35))}
\end{aligned}$$

and by taking the expectation,

$$\begin{aligned}\mathbb{E}[T_j(n)] &\leq \left\lceil \frac{8}{\Delta_j^2}\log(n) \right\rceil \\ &+ \sum_{t=m+1}^{n} \sum_{s_*=1}^{T_*(t-1)} \sum_{s_j=u}^{T_j(t-1)} \mathbb{P}\left\{ \widehat{X}_{*,s_*} \leq \mu_* - \sqrt{\frac{2\log(t)}{s_*}} \right\} \\ &+ \sum_{t=m+1}^{n} \sum_{s_*=1}^{T_*(t-1)} \sum_{s_j=u}^{T_j(t-1)} \mathbb{P}\left\{ \widehat{X}_{j,s_j} \geq \mu_j + \sqrt{\frac{2\log(t)}{s_j}} \right\} \\ &\leq \frac{8}{\Delta_j^2}\log(n) + 1 + 2\sum_{t=m+1}^{n} t^{-4}(t-1-m)^2. \end{aligned} \tag{37}$$

where in the last step we created an upper bound for $T_*(t-1)$ by $(t-1-m)$ (this is the maximum number of cases where we have could have played the best arm, excluding the starting phase of m rounds). We similarly bounded $T_j(t-1)$. Therefore, by using (26),

$$\mathbb{E}[R_n] \leq \sum_{j=1}^{m} \Delta_j + \sum_{j:\mu_j<\mu_*} \frac{8}{\Delta_j}\log(n) + \sum_{j=1}^{m} \Delta_j \left(1 + 2\sum_{t=m+1}^{n} t^{-4}(t-1-m)^2\right).$$

Notice that the parenthesis is bound by $\mathcal{O}(1)$ because the terms in the sum decrease rapidly enough in t. We now have proven the theorem.

C Proof of Corollary 3.1

Proof. We will first prove the statement of the corollary for the UCB algorithm. Using previous results, we have

$$\begin{aligned}\mathbb{E}[R_n] &= \sum_{j=1}^{m} \Delta_j \mathbb{E}[T_j(n)] && \text{(by Eq. 2.)} \\ &= \sum_{j=1}^{m} \Delta_j \sqrt{\mathbb{E}[T_j(n)]} \cdot \sqrt{\mathbb{E}[T_j(n)]} \\ &\leq \sqrt{\sum_{j=1}^{m} \Delta_j^2 \mathbb{E}[T_j(n)]} \cdot \sqrt{\sum_{j=1}^{m} \mathbb{E}[T_j(n)]} && \text{(by Cauchy-Schwarz inequality.)} \\ &\leq \sqrt{\sum_{j=1}^{m} \Delta_j^2 \mathbb{E}[T_j(n)]} \cdot \sqrt{n} && \text{(since } \textstyle\sum_{j=1}^{m} \mathbb{E}[T_j(n)] \leq n\text{. We can't play more than } n \text{ arms in } n \text{ rounds.)} \\ &= \sqrt{\sum_{j:\Delta_j>0} \Delta_j^2 \mathbb{E}[T_j(n)]} \cdot \sqrt{n} \\ &\leq \sqrt{\sum_{j:\Delta_j>0} \Delta_j^2 \cdot \left(\frac{8}{\Delta_j^2}\log(n) + 1 + 2\sum_{t=m+1}^{n} t^{-4}(t-1-m)^2\right)}\sqrt{n} && \text{(from (37))} \\ &\leq \sqrt{\sum_{j:\Delta_j>0} \left(8\log(n) + \left(1 + 2\sum_{t=m+1}^{n} t^{-4}(t-1-m)^2\right)\Delta_j^2\right)}\sqrt{n} && \text{(multiplying } \Delta \text{ through)} \\ &\leq \sqrt{\sum_{j:\Delta_j>0} \left(8\log(n) + 1 + 2\sum_{t=m+1}^{n} t^{-4}(t-1-m)^2\right)}\sqrt{n}. && \text{(since } \Delta_j \leq 1\text{.)} \end{aligned}$$

This is the first line of the corollary. Onto ϵ-greedy.

The proof for ϵ-greedy is very similar, except for a different bound on $\mathbb{E}\left[T_j(n)\right]$: from Equations (12) and (23), we have

$$\begin{aligned}
\mathbb{E}[T_j(n)] &= \sum_{t=1}^{n} \mathbb{P}(\{I_t = j\}) \\
&\leq \sum_{t=1}^{n} \left[\varepsilon_t \frac{1}{m} + (1-\varepsilon_t)\mathbb{P}(\widehat{X}_{j,T_j(t-1)} \geq \widehat{X}_{i,T_i(t-1)} \ \ \forall i) \right] \\
&\leq k\left(\frac{t}{mke}\right)^{-\frac{k}{10}} \log\left(\frac{t}{mke}\right) + \frac{4e^{\frac{1}{2}}}{\Delta_j^2}\left(\frac{t}{mke}\right)^{-\frac{k\Delta_j^2}{4}}. \qquad (38)
\end{aligned}$$

Now we can repeat the same argument:

$$\begin{aligned}
\mathbb{E}\left[R_n\right] &= \sum_{j=1}^{m} \Delta_j \mathbb{E}\left[T_j(n)\right] && \text{(by Eq. 2.)} \\
&= \sum_{j=1}^{m} \Delta_j \sqrt{\mathbb{E}\left[T_j(n)\right]} \cdot \sqrt{\mathbb{E}\left[T_j(n)\right]} \\
&\leq \sqrt{\sum_{j=1}^{m} \Delta_j^2 \mathbb{E}\left[T_j(n)\right]} \cdot \sqrt{\sum_{j=1}^{m} \mathbb{E}\left[T_j(n)\right]} && \text{(by Cauchy-Schwarz inequality.)} \\
&\leq \sqrt{\sum_{j=1}^{m} \Delta_j^2 \mathbb{E}\left[T_j(n)\right]} \cdot \sqrt{n} && \text{(since } \textstyle\sum_{j=1}^{m} \mathbb{E}\left[T_j(n)\right] \leq n\text{. We can't play more than } n \text{ arms in } n \text{ rounds.)} \\
&= \sqrt{\sum_{j:\Delta_j>0} \Delta_j^2 \mathbb{E}\left[T_j(n)\right]} \cdot \sqrt{n}. \\
&\leq \sqrt{\sum_{j:\Delta_j>0} \Delta_j^2 \left(k\left(\frac{t}{mke}\right)^{-\frac{k}{10}} \log\left(\frac{t}{mke}\right) + \frac{4e^{\frac{1}{2}}}{\Delta_j^2}\left(\frac{t}{mke}\right)^{-\frac{k\Delta_j^2}{4}} \right)} \cdot \sqrt{n} && \text{(by Eq. 38.)} \\
&\leq \sqrt{\sum_{j:\Delta_j>0} \left(\Delta_j^2 k\left(\frac{t}{mke}\right)^{-\frac{k}{10}} \log\left(\frac{t}{mke}\right) + 4e^{\frac{1}{2}}\left(\frac{t}{mke}\right)^{-\frac{k\Delta_j^2}{4}} \right)} \cdot \sqrt{n} \\
&\leq \sqrt{\sum_{j:\Delta_j>0} \left(k\left(\frac{t}{mke}\right)^{-\frac{k}{10}} \log\left(\frac{t}{mke}\right) + 4e^{\frac{1}{2}}\left(\frac{t}{mke}\right)^{-\frac{k\Delta_j^2}{4}} \right)} \cdot \sqrt{n}. && \text{(since } \Delta_j \leq 1\text{)}
\end{aligned}$$

We have now proved the second inequality in the corollary. □

D Notation summary

- m: number of arms;
- n: number of rounds;
- $X_j(t)$: random reward for playing arm j;
- μ_*: mean reward of the optimal arm ($\mu_* = \max_{1 \leq j \leq m} \mu_j$);
- Δ_j: difference between the mean reward of the optimal arm and the mean reward of arm j ($\Delta_j = \mu_* - \mu_j$);
- $\hat{X}_j$: current estimate of μ_j;
- I_t: arm played at turn t;
- $T_j(t-1)$: number of times arm j has been played before round t starts;
- k: a constant greater than 10 such that $k > \frac{4}{\min_j \Delta_j}$ in Algorithm 1;
- $\beta_j(t)$: upper bound on the probability of considering suboptimal arm j being the best arm at round t when using Algorithm 1;
- n': particular time defined as km in the comparison between Algorithm 1 in Section 1;
- R_n: total regret at round n.

References

Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time analysis of the multi-armed bandit problem. Machine learning, 47(2-3):235–256.

Bubeck, S. and Cesa-Bianchi, N. (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. Foundations and Trends® in Machine Learning, 5(1):1–122.

Lai, T. L. and Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. Advances in applied mathematics, 6(1):4–22.

Lattimore, T. and Szepesvári, C. (2020). Bandit algorithms. Cambridge University Press.

Slivkins, A. (2019). Introduction to multi-armed bandits. Foundations and Trends® in Machine Learning, 12(1-2):1–286.

Wang, T., Ye, W., Geng, D., and Rudin, C. (2020). Towards practical Lipschitz bandits. In FODS.