Algorithmic Foundations of Learning **Lecture 15**

Stochastic Multi-Armed Bandit Problem and Algorithms

Lecturer: Patrick Rebeschini *Version: December 1st 2019*

15.1 Introduction

We now look at the setting of online statistical learning, and, in particular, at the celebrated problem of stochastic multi-armed bandit. We first recall the general setting of online statistical optimization from Lecture 1, then describe two natural algorithms to solve this problem in the limited information (a.k.a. bandit) case: Explore-then-Commit and Greedy. We illustrate the reason why these algorithms fall short in achieving a good performance. This discussion will lead us to the development of an optimal algorithm: the Upper Confidence Bound (UCB) algorithm.

In previous lectures we have seen how concentration inequalities are a fundamental tool to *analyze* the behavior of optimization algorithms (in the offline statistical learning setting), establishing convergence bounds that hold with high probability. Today we will see how concentration inequalities are also a fundamental tool to *develop* algorithms (in the online statistical learning setting): UCB is based on confidence bounds for functions of many random variables, i.e., on concentration inequalities!

15.2 Multi-Armed Bandit Problem

Let us recall from Lecture 1 the setting of online statistical optimization.

Algorithm 1: Online Statistical Learning

At every time step $t = 1, 2, \dots, n$:

1. Choose an action $A_t \in \mathcal{A}$ (possibly using some external source of randomness), where $\mathcal{A}$ is a set of admissible actions.
2. A data point $Z_t \in \mathcal{Z}$ is sampled from an *unknown* distribution. The setting where Z_t is revealed to the player is called the *full information* setting. The setting where Z_t is not revealed to the player is called the *limited information* setting, or *bandit* setting.
3. Suffer a loss $\ell(A_t, Z_t)$ where $\ell : \mathcal{A} \times \mathcal{Z} \to \mathbb{R}_+$ is a given loss function. Let $r : \mathcal{A} \to \mathbb{R}_+$ be the expected/population risk, defined as the average loss function

$$\boxed{r(a) := \mathbf{E}\, \ell(a, Z)}$$

   Define the *normalized pseudo-regret* as follows

$$\boxed{\frac{1}{n} \sum_{t=1}^{n} r(A_t) - \inf_{a \in \mathcal{A}} r(a)}$$

Goal: Minimize and control the normalized pseudo-regret as a function of the sample size n and notions of "complexity" of the action set $\mathcal{A}$ of the loss function ℓ.

The classical stochastic multi-armed (finite-arm) bandit problem is obtained by making the following choices:

- The action set is finite with $\mathcal{A} = \{1, \dots, k\}$ and $|\mathcal{A}| = k$. Each action $a \in \mathcal{A}$ is regarded as an *arm* that the player can play/pull.
- At each time step t, the data point Z_t is a k-dimensional vector where each component is sampled independently from (possibly different) unknown distributions. For the sake of concreteness, henceforth we assume that the component distributions are supported on $[0, 1]$. Namely, $Z_t = (Z_{t,1}, \dots, Z_{t,k}) \in \mathcal{Z} = [0, 1]^k$ and for each $a \in \mathcal{A}$ the sequence $Z_{1,a}, \dots, Z_{n,a} \in \mathbb{R}$ is i.i.d. from a certain unknown distribution with mean μ_a. As we are in the bandit setting, Z_t is *not* revealed to the player.
- The loss associated by playing an arm in $\mathcal{A}$ is given by the opposite value of the coordinate of the data point associated with that arm, namely, $\ell(A_t, Z_t) = -Z_{t,A_t}$. We regard each component $Z_{t,a}$ of Z_t as the *reward* that the player gets by playing arm a (note that the player only observes the reward obtained by playing a given arm and does not observe the rewards from the arms that are not played).

In this case, the expected risk reads $r(a) = \mathbf{E}\,\ell(a, Z) = -\mathbf{E}Z_a := -\mu_a$, the normalized pseudo-regret reads $\max_{a\in\mathcal{A}} \mu_a - \frac{1}{n}\sum_{t=1}^n \mu_{A_t}$. If we let

$$a^\star \in \operatorname*{argmax}_{a\in\mathcal{A}} \mu_a$$

be one of the optimal arms (there could be many), the *pseudo-regret* (obtained by multiplying the previous quantity by the time horizon n) reads

$$\boxed{R_n := n\mu_{a^\star} - \sum_{t=1}^n \mu_{A_t}}$$

The goal is to find an algorithm (i.e., a set of actions $A_1, \dots, A_n \in \mathcal{A}$) that minimizes R_n, with the constraint (due to the online nature of the problem) that each action A_t can depend only on the information available prior to time t, namely, on $(A_s, \ell(A_s, Z_s))_{s\in[t-1]}$, or, equivalently, $(A_s, Z_{s,A_s})_{s\in[t-1]}$.

Clearly, the pseudo-regret R_n is upper-bounded by a function that increases linearly with n. Hence, in order for learning to occur, we aim to find an algorithm that achieves a bound that increases only sub-linearly with n. Henceforth, we look at bounds in expectations: we consider various algorithms and derive upper bounds for the quantity $\mathbf{E}R_n$.

How can we design an algorithm to solve this problem? The player plays the arms iteratively, collect the outcome of each play (the rewards), and use this information to choose which arm to pull next. Within this scheme, one might naturally think that the sample mean of the rewards obtained per each arm is a useful statistics for deciding which arm to pull next.

The number of times that arm a is pulled up to time t is given by

$$N_{t,a} := \sum_{s=1}^t \mathbf{1}_{A_s=a}.$$

The sample mean of the rewards obtained by playing arm a up to time t is given by

$$M_{t,a} := \frac{1}{N_{t,a}} \sum_{s=1}^t Z_{s,a}\mathbf{1}_{A_s=a}.$$

Let $\Delta_a := \mu_{a^\star} - \mu_a$ be the sub-optimality gap of arm a. The pseudo-regret R_n can be written as a function of the (deterministic) sub-optimality gaps $\{\Delta_a\}_{a\in\mathcal{A}}$ and of the (random) number of times that each arm is pulled by the end of the time horizon n, i.e., the quantities $\{N_{n,a}\}_{a\in\mathcal{A}}$.

Proposition 15.1 *We have*

$$\boxed{R_n = \sum_{a \in \mathcal{A}} \Delta_a N_{n,a}}$$

Proof: The proof follows as $n = \sum_{a \in \mathcal{A}} N_{n,a}$ and $\sum_{t=1}^{n} \mu_{A_t} = \sum_{a \in \mathcal{A}} \mu_a N_{n,a}$. ■

This result shows that the randomness of R_n is exclusively related to the number of times that each sub-optimal arm is pulled by the end of the game.

15.3 Explore-then-Commit

We start by considering a simple algorithm, which first explores all the possible actions for a prescribed number of times $\varepsilon \in \mathbb{N}_+$ (exploration phase), and then keeps playing the action that has performed the best in the initial phase (exploitation phase), where the notion of "best performance" is related to maximizing the empirical mean. The parameter ε is the number of times each action is played in the exploration phase before moving to the exploitation phase. This parameter controls the exploration/exploitation trade-off: the greater ε is, the more exploration is performed. Recall that k is the total number of arms.

Algorithm 2: Explore-then-Commit(ε)

Input: $\varepsilon \in \mathbb{N}_+$;
for $t = 1, \ldots, \varepsilon k$ **do**
  set $A_t = (t-1) \pmod{k} + 1$;
end
for $t = \varepsilon k + 1, \ldots, n$ **do**
  set $A_t \in \operatorname{argmax}_{a \in \mathcal{A}} M_{\varepsilon k, a}$;
end

This algorithm yields a pseudo-regret that is linear in n. This should be intuitive: as the exploration phase has a fixed length, there is always (aside from trivial pathological cases) a constant probability of not choosing the best arm in the exploration phase, which yields a linear pseudo-regret. If the pseudo-regret R_n is linear with a constant probability, then its expectation is also linear.

Proposition 15.2 *For any $\varepsilon \in \mathbb{N}_+$, there exists a stochastic multi-armed bandit problem so that Explore-then-Commit(ε) yields*

$$\boxed{\mathbf{E}R_n = cn + \tilde{c}}$$

for some constants $c, \tilde{c} \in \mathbb{R}_+$ that do not depend on n.

Proof: Take the case of two arms, $k = 2$. Let arm a have a fix reward equal to $\mu_a < 1$, and the other (optimal) arm have a Bernoulli (0 or 1) reward with mean $\mu_{a^\star} > \mu_a$. In this case, the probability of not choosing the best arm in the exploration phase is strictly positive and reads:

$$p = \mathbf{P}(M_{2\varepsilon,a^\star} < M_{2\varepsilon,a}) = \mathbf{P}(\text{Binomial}(\varepsilon, \mu_{a^\star}) < \varepsilon \mu_a) > 0.$$

The number of times that the sub-optimal arm is played after the exploration phase is equal to $n - \varepsilon k$ with probability p (the suboptimal arm is played ε times during the exploration phase). By Proposition 15.1,

$$\mathbf{E}R_n = \Delta_a \mathbf{E}N_{n,a} = \Delta_a(\varepsilon + (n - \varepsilon k)p) = \Delta_a pn + \Delta_a \varepsilon(1 - kp).$$

■

15.4 Greedy

The Explore-Then-Commit algorithm suffers from linear pseudo-regret as the exploration phase is finite. One might try to overcome this behavior by forcing the algorithm to keep exploring. One way to achieve this is by introducing some randomness in the algorithm; more specifically, a coin flip that with probability $\varepsilon \in (0,1)$ plays an arm uniformly sampled among all possible choices (exploration), and with probability $1-\varepsilon$ plays the arm with the largest sample mean (exploitation). The parameter ε controls the exploration/exploitation trade-off: the greater ε is, the more exploration is performed.

Algorithm 3: Greedy(ε)

Input: $\varepsilon \in (0,1)$;
for $t = 1, \dots, k$ **do**
  set $A_t = t$;
end
for $t = k+1, \dots, n$ **do**
  set $A_t \begin{cases} \in \operatorname{argmax}_{a \in \mathcal{A}} M_{t-1,a} & \text{with probability } 1-\varepsilon \\ \sim \text{Unif}\{1, \dots, k\} & \text{with probability } \varepsilon \end{cases}$
end

Unfortunately, this algorithm also suffers from a linear pseudo-regret.

Proposition 15.3 *For any $\varepsilon \in (0,1)$, Greedy(ε) yields*

$$\boxed{\mathbf{E}R_n = cn + \tilde{c}}$$

for some constants $c, \tilde{c} \in \mathbb{R}_+$ that do not depend on n.

Proof: The expected number of times that each arm $a \in \mathcal{A}$ is played is upper-bounded as follows:

$$\mathbf{E}N_{n,a} \geq 1 + \frac{\varepsilon}{k}(n-k).$$

In fact, after the initial phase when each arm is played once, there are still $n-k$ plays to be made and at every time step the probability that each arm is played is at least ε/k.

By Proposition 15.1,

$$\mathbf{E}R_n = \sum_{a \in \mathcal{A}} \Delta_a \mathbf{E}N_{n,a} \geq \frac{\varepsilon}{k}(n-k) \sum_{a \in \mathcal{A}} \Delta_a.$$

The proof is concluded as the pseudo-regret can not grow faster than n. ■

15.5 Upper Confidence Bound (UCB)

The algorithms so far discussed only achieve a linear pseudo-regret. From the ongoing discussion it appears that a good algorithm should always keep the doors open to exploration, but enforcing exploration with a mechanism that *does not depend on time* does not seem to yield good results. Indeed, one can modify the Greedy algorithm and consider a sequence $\{\varepsilon_t\}_{t \geq k+1}$ of exploration/exploitation parameters that decrease with time. It can be proved that this strategy yields pseudo-regrets that grow poly-logarithmically in n (in expectation) provided one knows a lower bound $\min_{a:\Delta_a > 0} \Delta_a$ [1]. This is a considerable improvement from

linear-growth regrets but it is not optimal (the lower bound only grows logarithmically and does not depend on $\min_{a:\Delta_a>0} \Delta_a$). As we see next, the key to success stems from designing a protocol that implements exploration *adaptively*, as a function of the number of times each armed is played.

The algorithm that we now consider works by modifying the arm selection criteria, i.e., the notion of "best performance": instead of playing the arm with the largest sample mean, we consider the arm with the largest upper confidence bound. The algorithm takes the name of Upper Confidence Bound (UCB).

Define the upper confidence bound as

$$U_{t,a} := M_{t,a} + \sqrt{\frac{\varepsilon \log t}{2N_{t,a}}}. \tag{15.1}$$

for a given $\varepsilon \in \mathbb{R}_+$.

Algorithm 4: UCB(ε)

Input: $\varepsilon \in \mathbb{R}_+$;
for $t = 1, \ldots, k$ **do**
  set $A_t = t$;
end
for $t = k+1, \ldots, n$ **do**
  set $A_t \in \operatorname{argmax}_{a \in \mathcal{A}} U_{t-1,a}$;
end

Note that A_t is a function of the upper confidence bounds $U_{t-1,a}$'s; as such, A_t depends on $(A_s, Z_{s,A_s})_{s\in[t-1]}$.

At each time step, the UCB algorithm estimates the mean reward of each arm *at some fixed confidence level* by creating an upper confidence bound, and, as a way to promote exploration, chooses the arm that looks the best under this estimate (not using the sample mean itself). The fact that the algorithm chooses the best arm in a random environment is often referred to as the principle of *optimism in the face of uncertainty*, and many algorithms for stochastic bandit problems rely on this principle. Note that the upper confidence bound $U_{t,a}$ for arm a increases with time, albeit only logarithmically, and it is inversely proportional to the number of times that the arm has been played.

The two terms on the right hand side of (15.1) determine the trade-off between exploration and exploitation. The term $M_{t,a}$ represents an estimate of the expected reward of arm a at time t: the largest this term is, the more exploitation is promoted. The term $\sqrt{\varepsilon \log t/(2N_{t,a})}$ represents an estimate of the uncertainty in the arm's estimate: the largest this term is, the more exploration is promoted. The parameter ε controls the exploration/exploitation trade-off: the greater ε is, the more exploration is performed.

15.6 Distribution-Dependent Bounds for UCB

For a proper choice of the exploration/exploitation parameter ε, the UCB algorithm achieves a logarithmic pseudo-regret in expectation.

Theorem 15.4 *For any $\varepsilon > 0$, UCB(ε) yields*

$$\mathbf{E}R_n \le \sum_{a\in\mathcal{A}} \frac{2\varepsilon \log n}{\Delta_a} + 2\sum_{a\in\mathcal{A}} \Delta_a \sum_{t=k}^{n-1} \frac{1}{t^{\varepsilon}}$$

For $\varepsilon \geq 1$ UCB(ε) yields a pseudo-regret that grows logarithmically in n, while for $\varepsilon < 1$ we are only guaranteed that the pseudo-regret grows at most polynomially in n:

$$\sum_{t=k}^{n-1} \frac{1}{t^{\varepsilon}} \leq \int_k^n (t-1)^{\varepsilon} \mathrm{d}t \leq \begin{cases} \frac{n^{1-\varepsilon}}{1-\varepsilon} & \text{if } \varepsilon < 1 \\ \log n & \text{if } \varepsilon = 1 \\ \frac{1}{\varepsilon - 1} & \text{if } \varepsilon > 1 \end{cases}$$

The bound given in Theorem 15.4 is distribution-dependent, as it depends on the sub-optimality gaps Δ_a's.

We break the proof of Theorem 15.4 into three steps: a general step that holds for *any* algorithm; a key step that holds for UCB; and a final step that simply combines the previous results together. We give proofs that make explicit the rationale behind the design of the upper confidence bound in (15.1).

Step 1: Proposition 15.1 tells us that in order to control $\mathbf{E}R_n$ it suffices to control $\mathbf{E}N_{n,a}$ for each sub-optimal arm $a \in \mathcal{A}$. We show that $\mathbf{E}N_{n,a}$ is bounded by the sum of the probabilities that arm a is played given that it has been played for a certain number of times in the past.

Proposition 15.5 *For any non-decreasing sequence $s_1 \leq \ldots \leq s_n$ in $\mathbb{R}_+$ and any $a \in \mathcal{A}$, we have*

$$\boxed{\mathbf{E}N_{n,a} \leq s_n + \sum_{t=k}^{n-1} \mathbf{P}(A_{t+1} = a | N_{t,a} \geq s_t)}$$

Proof: Fix a non-decreasing sequence $s_1 \leq \ldots \leq s_n$ in $\mathbb{R}_+$ and $a \in \mathcal{A}$. Using that for any events E, F we have $\mathbf{1}_E = \mathbf{1}_{E,F} + \mathbf{1}_{E,F^C}$, and that the algorithm plays each arm once at the beginning, we find

$$N_{n,a} = 1 + \sum_{t=k}^{n-1} \mathbf{1}_{A_{t+1}=a} = 1 + \sum_{t=k}^{n-1} \mathbf{1}_{A_{t+1}=a, N_{t,a} < s_t} + \sum_{t=k}^{n-1} \mathbf{1}_{A_{t+1}=a, N_{t,a} \geq s_t}.$$

As the sequence is non-decreasing, we have

$$1 + \sum_{t=k}^{n-1} \mathbf{1}_{A_{t+1}=a, N_{t,a} < s_t} \leq s_n.$$

We prove this by contradiction. Assume that the indicator $\mathbf{1}_{A_{t+1}=a, N_{t,a} < s_t}$ takes the value 1 for more than $s_n - 1$ times, for the parameter t in the range $\{k, \ldots, n-1\}$. Let $\tilde{t}$ be the parameter corresponding to the $\lfloor s_n \rfloor$-th time that the indicator function is equal to 1. Then the event $\{A_{\tilde{t}+1} = a, N_{\tilde{t},a} < s_{\tilde{t}}\}$ must be true, but this is impossible as arm a must have been played at least $1 + \lfloor s_n \rfloor$ times, namely, $N_{\tilde{t},a} \geq 1 + \lfloor s_n \rfloor$.

Taking the expected value we find

$$\mathbf{E}N_{n,a} \leq s_n + \sum_{t=k}^{n-1} \mathbf{P}(A_{t+1} = a, N_{t,a} \geq s_t) \leq s_n + \sum_{t=k}^{n-1} \mathbf{P}(A_{t+1} = a | N_{t,a} \geq s_t)\, \mathbf{P}(N_{t-1,a} \geq s_t),$$

and the proof follows as $\mathbf{P}(N_{t-1,a} \geq s_t) \leq 1$ for any t. ■

Step 2: The previous step is general and holds for *any* algorithm (that plays each arm once in the initial phase). The next step is specific to UCB, and is the key result to prove Theorem 15.4. We show that if a sub-optimal arm has been played for a number of times that is inversely proportional to the square of the sub-optimality gap Δ_a, then the probability that it is played again using the UCB protocol is small. This is an expected property of a good algorithm, as arms with a small sub-optimality gap are harder to detect and so they should be played more times (as compared to arms with a large sub-optimality gap) in order for a good algorithm to be guaranteed to play them again with small probability.

Lemma 15.6 *Let* $A_{t+1} = \text{argmax}_{a\in\mathcal{A}} U_{t,a}$ *with* $U_{t,a} = M_{t,a} + \sqrt{\frac{\log(1/\delta)}{2N_{t,a}}}$. *For any* $a \in \mathcal{A}$ *such that* $\Delta_a > 0$ *we have*

$$\boxed{\mathbf{P}\left(A_{t+1} = a | N_{t,a} \geq 2\frac{\log(1/\delta)}{\Delta_a^2}\right) \leq 2\delta}$$

Proof: We give a general proof that is meant to illustrate where the choice of the factor $2\frac{\log(1/\delta)}{\Delta_a^2}$ in the conditional probability comes from. By the definition of UCB, we have

$$\{A_{t+1} = a\} = \{U_{t,b} \leq U_{t,a}\ \forall b \in \mathcal{A}\} \subseteq \{U_{t,a^\star} \leq U_{t,a}\} \subseteq \Big(\{U_{t,a^\star} \leq \mu_{a^\star}\} \cup \{\mu_{a^\star} \leq U_{t,a}\}\Big),$$

where the last set operation is immediately verified by looking at the complimentary, namely,

$$\{U_{t,a^\star} > U_{t,a}\} \supseteq \Big(\{U_{t,a^\star} > \mu_{a^\star}\} \cap \{\mu_{a^\star} > U_{t,a}\}\Big).$$

Let $s \in \mathbb{R}_+$. By the union bound,

$$\mathbf{P}(A_{t+1} = a | N_{t,a} \geq s) \leq \mathbf{P}(U_{t,a^\star} \leq \mu_{a^\star} | N_{t,a} \geq s) + \mathbf{P}(\mu_{a^\star} \leq U_{t,a} | N_{t,a} \geq s).$$

Recall that by the Hoeffding's inequality, if $X_1, \ldots, X_n$ are i.i.d. samples from $[0,1]$ with mean μ, then for any $x > 0$

$$\mathbf{P}\left(\frac{1}{n}\sum_{i=1}^n X_i - \mu \geq x\right) \leq e^{-2nx^2}, \qquad \mathbf{P}\left(\frac{1}{n}\sum_{i=1}^n X_i - \mu \leq -x\right) \geq e^{-2nx^2},$$

and with the choice $x = \sqrt{\frac{\log(1/\delta)}{2n}}$ we get, respectively,

$$\mathbf{P}\left(\frac{1}{n}\sum_{i=1}^n X_i - \mu \geq \sqrt{\frac{\log(1/\delta)}{2n}}\right) \leq \delta, \tag{15.2}$$

$$\mathbf{P}\left(\frac{1}{n}\sum_{i=1}^n X_i - \mu \leq -\sqrt{\frac{\log(1/\delta)}{2n}}\right) \leq \delta. \tag{15.3}$$

By the independence between the rewards and the arms' pulls, using (15.3) we get

$$\begin{aligned}
\mathbf{P}(U_{t,a^\star} \leq \mu_{a^\star} | N_{t,a} \geq s) &= \mathbf{P}\left(M_{t,a^\star} - \mu_{a^\star} \leq -\sqrt{\frac{\log(1/\delta)}{2N_{t,a^\star}}}\,\middle|\, N_{t,a} \geq s\right) \\
&= \mathbf{E}\left[\mathbf{P}\left(M_{t,a^\star} - \mu_{a^\star} \leq -\sqrt{\frac{\log(1/\delta)}{2N_{t,a^\star}}}\,\middle|\, N_{t,a} \geq s, N_{t,a^\star}\right)\middle| N_{t,a} \geq s\right] \\
&= \mathbf{E}\left[\mathbf{P}\left(M_{t,a^\star} - \mu_{a^\star} \leq -\sqrt{\frac{\log(1/\delta)}{2N_{t,a^\star}}}\,\middle|\, N_{t,a^\star}\right)\middle| N_{t,a} \geq s\right] \\
&\leq \mathbf{E}[\delta | N_{t,a} \geq s] = \delta.
\end{aligned}$$

Note that we have to condition on $N_{t,a^\star}$, as the concentration inequality (15.3) holds for a *deterministic* n. In the last equality we used the fact that given $N_{t,a^\star}$, the arms' pulls and the event $\{N_{t,a} \geq s\}$ are independent. Using that $\mu_{a^\star} = \mu_a + \Delta_a$ by the definition of the sub-optimality gaps, we find

$$\mathbf{P}(\mu_{a^\star} \leq U_{t,a} | N_{t,a} \geq s) = \mathbf{P}\left(M_{t,a} - \mu_a \geq \Delta_a - \sqrt{\frac{\log(1/\delta)}{2N_{t,a}}}\,\middle|\, N_{t,a} \geq s\right).$$

Proceeding as above, we would know how to bound this probability using (15.2) if $\Delta_a = 2\sqrt{\frac{\log(1/\delta)}{2N_{t,a}}}$ or, equivalently, $N_{t,a} = 2\frac{\log(1/\delta)}{\Delta_a^2}$. This immediately suggest to take $s = \tilde{s} := 2\frac{\log(1/\delta)}{\Delta_a^2}$ so that the event $\{N_{t,a} \geq \tilde{s}\}$ is equivalent to the event $\{\Delta_a \geq 2\sqrt{\frac{\log(1/\delta)}{2N_{t,a}}}\}$ and, proceeding as above,

$$\begin{aligned}
\mathbf{P}(\mu_{a^\star} \leq U_{t,a} | N_{t,a} \geq \tilde{s}) &\leq \mathbf{P}\left(M_{t,a} - \mu_a \geq \sqrt{\frac{\log(1/\delta)}{2N_{t,a}}} \middle| N_{t,a} \geq \tilde{s} \right) \\
&= \mathbf{E}\left[\mathbf{P}\left(M_{t,a} - \mu_a \geq \sqrt{\frac{\log(1/\delta)}{2N_{t,a}}} \middle| N_{t,a} \geq \tilde{s}, N_{t,a} \right) \middle| N_{t,a} \geq \tilde{s} \right] \\
&= \mathbf{E}\left[\mathbf{P}\left(M_{t,a} - \mu_a \geq \sqrt{\frac{\log(1/\delta)}{2N_{t,a}}} \middle| N_{t,a} \right) \middle| N_{t,a} \geq \tilde{s} \right] \\
&\leq \mathbf{E}[\delta | N_{t,a} \geq \tilde{s}] = \delta.
\end{aligned}$$

Putting everything together we find

$$\mathbf{P}\left(A_{t+1} = a | N_{t,a} \geq 2\frac{\log(1/\delta)}{\Delta_a^2} \right) \leq 2\delta.$$

■

Step 3: Combining the previous two steps together, we obtain the following result that immediately yields the proof of Theorem 15.4.

Proposition 15.7 *For any $a \in \mathcal{A}$ we have*

$$\boxed{\mathbf{E}N_{n,a} \leq 2\frac{\varepsilon \log n}{\Delta_a^2} + 2\sum_{t=k}^{n-1} \frac{1}{t^\varepsilon}}$$

Proof: With the choice $\delta = 1/t^\varepsilon$, Lemma 15.6 yields

$$\mathbf{P}\left(A_{t+1} = a | N_{t,a} \geq 2\frac{\varepsilon \log t}{\Delta_a^2} \right) \leq \frac{2}{t^\varepsilon}.$$

Choosing $s_t = 2\frac{\varepsilon \log t}{\Delta_a^2}$, Proposition 15.5 yields

$$\mathbf{E}N_{n,a} \leq s_n + \sum_{t=k}^{n-1} \mathbf{P}(A_{t+1} = a | N_{t,a} \geq s_t) \leq 2\frac{\varepsilon \log n}{\Delta_a^2} + 2\sum_{t=k}^{n-1} \frac{1}{t^\varepsilon}.$$

■

The proof of Theorem 15.4 follows by Proposition 15.7 and Proposition 15.1.

15.7 Distribution-Independent Bounds for UCB

Theorem 15.4 yields distribution-dependent upper bounds for UCB that only grow logarithmically with n and depend on the sub-optimality gaps Δ_a's. It is possible to show that this result is optimal in the minimax

sense: using the minimax notion of optimality, which we will introduce in the next lecture, no other algorithm can do better!

However, if for some $a \in \mathcal{A}$ we let $\Delta_a = \log n/n$, then a direct application of Theorem 15.4 yields upper bounds for $\mathbf{E}R_n$ that increase linearly with n, which per se does not tell us anything about the learning capabilities of UCB in worst case scenarios (recall that R_n is always upper-bounded by a linear function of n). Things are not as bad as they might seem though. A more refined application of Proposition 15.7 allows to prove a distribution-independent (i.e., worst-case) bound for UCB that for $\varepsilon \geq 1$ grows as $\sqrt{kn \log n}$ (recall that we think of k as fixed by the problem formulation, and we are only interested in the dependence of the bounds with the length of the game, i.e., with n). As we will see next time, this result is quasi-optimal in the minimax sense, as we will prove a *minimax lower bound* that grows like $\sqrt{kn}$. To recap, in the minimax sense, UCB yields optimal distribution-dependent bounds and quasi-optimal distribution-independent bounds.

Theorem 15.8 *For any $n \geq k$ we have*

$$\boxed{\mathbf{E}R_n \leq 2\sqrt{2\varepsilon kn \log n} + 2k \sum_{t=k}^{n-1} \frac{1}{t^{\varepsilon}}}$$

Proof: Fix $c > 0$. We can divide the arms into two groups: those such that $\Delta_a < c$, and those such that $\Delta_a \geq c$. Using the decomposition of the pseudo-regret given in Proposition 15.1, applying Proposition 15.7 only to the arms in the second group, we find

$$\begin{aligned}
\mathbf{E}R_n &= \sum_{a \in \mathcal{A}} \Delta_a \mathbf{E}N_{n,a} = \sum_{a \in \mathcal{A}: \Delta_a \leq c} \Delta_a \mathbf{E}N_{n,a} + \sum_{a \in \mathcal{A}: \Delta_a > c} \Delta_a \mathbf{E}N_{n,a} \\
&\leq c \sum_{a \in \mathcal{A}: \Delta_a \leq c} \mathbf{E}N_{n,a} + \sum_{a \in \mathcal{A}: \Delta_a > c} \Delta_a \left(2\frac{\varepsilon \log n}{\Delta_a^2} + 2\sum_{t=k}^{n-1} \frac{1}{t^{\varepsilon}} \right) \\
&\leq cn + \sum_{a \in \mathcal{A}: \Delta_a > c} \frac{2\varepsilon \log n}{\Delta_a} + 2 \sum_{a \in \mathcal{A}: \Delta_a > c} \Delta_a \sum_{t=k}^{n-1} \frac{1}{t^{\varepsilon}},
\end{aligned}$$

where for the last inequality we used $\sum_{a \in \mathcal{A}: \Delta_a \leq c} N_{n,a} \leq \sum_{a \in \mathcal{A}} N_{n,a} = n$. Using that $\Delta_a \leq 1$ (recall from Section 15.2 that we assume the unknown distributions to be supported in $[0, 1]$), we finally obtain

$$\mathbf{E}R_n \leq cn + \frac{2\varepsilon k \log n}{c} + 2k \sum_{t=k}^{n-1} \frac{1}{t^{\varepsilon}}.$$

Optimizing this bound over $c > 0$ we find the optimal value $c = \sqrt{\frac{2\varepsilon k \log n}{n}}$ which yields the result in the statement of the theorem. ∎

References

[1] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2):235–256, May 2002.