Optimization for Machine Learning 50.579

Instructor: Ioannis Panageas Scribed by: Deepanway Ghosal, Wayne Lin

Lecture 9. Introduction to Multi-armed Bandits.

1 Introduction

Multi-armed bandit problems are examples of sequential decision problems where a fixed set of resources has to be allocated between alternative choices to maximize the expected gain. This is a classic framework with exploration-exploitation trade-off, where algorithms make decisions over time under uncertainty. At the beginning, the properties of the different alternative choices are unknown or partially known, and may become better understood as resources are allocated to those choices at the time of each sequential decision. Hence, there is always this balance between staying with the choice which gave the highest reward in the past and exploring new options which may give better rewards in the future. Therefore, the goal is to design algorithms with the expectation that they will be able to choose the optimal sequence of actions and consequently give us higher rewards or payoffs.

More formally, the multi-arm bandit problem is said to be a sequential allocation problem governed by a set of actions. At each time step an action is selected, and a reward is observed. The objective is to maximize the total reward observed for a given time horizon (total number of time steps). We first introduce the framework:

The player is given K different arms (actions) and a total time horizon of T. Both K & T are known. At each time step $t = 1, 2, .., T$:

1. The player chooses an arm a_t.
2. A reward $r_t \in [0, 1]$ is observed.

In this lecture we will analyze *stochastic bandits* and *adversarial bandits.* We have the following set of assumptions for *stochastic bandits*,

- The reward for each arm $a \in [K]$ is independent and identically distributed (IID). For each arm a there is a distribution D_a over reals, called the *reward distribution.* This distribution is unknown to the player. Each time arm a is chosen by the player, the reward is sampled independently from the distribution D_a.
- The player observes only the reward for the selected arm. Rewards for other arms, that were not selected, are not observed.

This assumption of having IID rewards is a feature of *stochastic bandits*. However, in *adversarial bandits*, instead of rewards, we have costs and the costs are not IIDs. In this case we have the following assumptions,

- The costs for all arms and all rounds are chosen in advance by the adversary. Each time arm a is chosen by the player in some round t, the cost $\in [0,1]$ for that arm a in that round t is revealed to the player.
- The player observes only the cost for the selected arm. Costs for other arms, that were not selected, are not observed.

Notation: We introduce the following set of notations first. Arms are denoted by a and the set of all arms is A. Rounds are denoted by t, and the arm chosen at round t is denoted by a_t. The mean reward of arm a is denoted by $\mu(a) = E[D_a]$. The arm with the best mean reward will then have a reward of $\mu^* = \max_{a \in A} \mu(a)$.

Now, the strategy of the player is to maximize the sum of all collected rewards. The best reward is obtained when the optimal arm (the arm with the best mean reward μ^*) is chosen at every round. However, any information about $\mu(a)$ or D_a is unknown to the player at the very beginning. The strategy of maximizing the sum of rewards, is therefore, equivalent to minimizing the regret, where the regret is defined as,

$$R(T) = \mu^* T - \sum_{t=1}^{T} \mu(a_t)$$

Intuitively, the regret describes how far off are we from the optimal reward at each round and sums up those differences over the whole time horizon T. Choosing a bad arm would mean we will end up with a comparatively higher regret, whereas, choosing the best arm at every round would mean we will have a regret of 0 and the highest possible reward. So, a strategy of minimizing the regret would accomplish our goal of maximizing the reward.

Note that, $R(T)$ is a random variable as the arm a_t chosen at round t is a random quantity and may depend on the randomness in the rewards and the algorithm. So, for analysis we will consider the expected value of the regret: $\mathbb{E}[R(T)]$. We will analyze the asymptotic dependence of regret $\mathbb{E}[R(T)]$ on the time horizon T.

2 Algorithms (Stochastic Setting)

In this section, we will present a number of different algorithms that the player can employ and we will analyze the regret in terms of time horizon T and number of arms K.

2.1 Explore First Algorithm

The most simple strategy could be to first estimate the expected reward for all the arms and then use the arm with the maximum estimated reward for the rest of the rounds. We do this by keeping the initial few rounds only for exploration. In this phase, we explore all the arms uniformly. Then in the next phase, we exploit the optimal arm for the rest of the rounds.

1. Exploration phase: try each arm N/K times.
2. Select the arm a^* with the highest average reward (break ties arbitrarily).
3. Exploitation phase: play a^* in all remaining $T - N$ rounds.

The parameter N is fixed in advance as a function of T and K. We will choose it in a way such that it minimizes the regret.

Let us denote the average reward for arm a after the exploration phase as $\hat{\mu_a}$. We can use Hoeffding inequality to bound the quantity $|\hat{\mu_a} - \mu_a|$ as following,

$$Pr\{|\hat{\mu_a} - \mu_a| \leq r(a)\} \geq 1 - \frac{2}{T^4} \tag{1}$$

Ideally we would want the average reward after the exploration phase to be a good estimate of the true expected reward. Hence the quantity $|\hat{\mu_a} - \mu_a|$ should be small, and using the Hoeffding inequality we can say that $|\hat{\mu_a} - \mu_a|$ can be made smaller than bounding radius $r(a) = \sqrt{\frac{2K \log T}{N}}$ with a large probability of at least $1 - \frac{2}{T^4}$

Now we define the *clean event* to be a event where inequality (1) holds for all arms. It follows that the probability of the *bad event*, which is the complement of the *clean event*, is going to be very small. So, for the rest of the analysis we would only consider the clean event because it happens with such a high probability.

We start the analysis in the case of a clean event. Suppose a^* is the arm with the best true expected reward. However, after the exploration phase, some other arm $a \neq a*$ is chosen because it has a higher average reward i.e. $\hat{\mu}(a) > \hat{\mu}(a^*)$. From (1), we can thus say,

$$\mu_a + r(a) \geq \hat{\mu_a} > \hat{\mu}(a^*) \geq \mu(a^*) - r(a^*)$$

or,

$$\hat{\mu}(a^*) - \mu_a \leq r(a) + r(a^*) = 2\sqrt{\frac{2K \log T}{N}} \tag{2}$$

Now, the N rounds in exploration phase can be considered to contribute at most N (regret of maximum 1 in each round) regret. For the exploitation phase, each round contributes $\hat{\mu}(a^*) - \mu_a$ which is bounded by $2\sqrt{\frac{2K \log T}{N}}$ from inequality (2). Hence the total regret after N rounds of exploration and $T - N$ rounds of exploitation,

$$\begin{aligned} R(T) &\leq N + 2\sqrt{\frac{2K \log T}{N}}(T - N) \\ &\leq N + \sqrt{\frac{8KT^2 \log T}{N}} \end{aligned} \tag{3}$$

The two summands in inequality (3) are monotonically increasing and decreasing with N. Hence, the sum can be minimized by making them approximately equal, which happens when we choose $N = 2T^{2/3}(K \log T)^{1/3}$. We can now put this value of N back in (3) to obtain

$$R(T) \leq 4T^{2/3}(K \log T)^{1/3} \tag{4}$$

To complete the result, we also need to analyze the case of the *bad events*. For *bad events* the regret can be at most T for T rounds. Moreover, the *bad event* happens when at least one arm doesn't satisfy (1) which happens with probability $\frac{K}{T^4}$ (from union bound of probability) or probability of $O(1/T^3)$. Hence, the overall regret,

$$\begin{aligned}\mathbb{E}[R(T)] &= \mathbb{E}[R(T)|\text{clean event}] * Pr[\text{clean event}] + \mathbb{E}[R(T)|\text{bad events}] * Pr[\text{bad events}] \\ &\leq 4T^{2/3}(K \log T)^{1/3} + T * O(T^{-3}) \\ &\leq O(T^{2/3}(K \log T)^{1/3})\end{aligned} \tag{5}$$

Theorem 2.1 *The explore first algorithm achieves regret* $O(T^{2/3}(K \log T)^{1/3})$

2.2 Epsilon Greedy Algorithm

The *explore first* algorithm performs very poorly in the exploration phase. A simple alternative way to alleviate this would be to use a greedy method. In this case we use a greedy action selection method to maximize current reward by exploiting current knowledge. In particular, we behave greedily most of time by selecting the best arm from our knowledge of previous rounds, but once in a while with a small probability ϵ we perform exploration by randomly selecting any one of the possible arms. As a result, the exploration phase becomes more spread over time and we can analyze the regret bounds even for small t. This is performed in the *epsilon greedy* algorithm.

```
for round $t$ = 1, 2, .. $T$ do
    Toss a coin with success probability $\epsilon_t$;
    if success then
        explore: choose an arm uniformly at random;
    else
        exploit: choose the arm with the highest observed average reward so far;
    end
end
```

We analyze the regret bound of this algorithm by fixing round t. After round t, for each arm a, following Hoeffding inequality, we will have,

$$Pr\{|\hat{\mu_a} - \mu_a| \leq \epsilon\} \geq 1 - 2e^{-2\epsilon^2(t\epsilon_t/K)} \tag{6}$$

Among the first t rounds, the exploration happens only in $t\epsilon_t$ rounds and hence, each arm on average will be explored $t\epsilon_t/K$ times. Naturally, some arms will be chosen more in the exploitation phase. For those arms, we will have even tighter probability bounds and hence, for the asymptotic analysis we can proceed from (6).

Proceeding in a similar way as in (2), for arm a_t chosen at round t, we will have,

$$\hat{\mu}(a^*) - \mu_{a_t} \leq 2\sqrt{\frac{2K\log t}{t\epsilon_t}} \tag{7}$$

Now, the expected value of the regret only at the particular round of t,

$$\begin{aligned}\mathbb{E}[\tilde{R}(t)] &= Pr[\text{coin toss success}] * 1 + Pr[\text{coin toss failure}] * (\hat{\mu}(a^*) - \mu_{a_t}) \\ &= \epsilon_t + (1-\epsilon_t) * 2\sqrt{\frac{2K\log t}{t\epsilon_t}} \\ &\leq \epsilon_t + 2\sqrt{\frac{2K\log t}{t\epsilon_t}}\end{aligned} \tag{8}$$

In (8), $\mathbb{E}(\tilde{R}(t))$ can be minimized by making the two summands approximately equal, resulting in $\epsilon_t = t^{-1/3}(K\log t)^{1/3}$. Now, the overall regret can be bounded as,

$$\begin{aligned}\mathbb{E}[R(t)] &= \mathbb{E}\left[\sum_1^t \tilde{R}(t)\right] \\ &\leq t * \mathbb{E}[\tilde{R}(t)] \\ &\leq t^{2/3}(K\log t)^{1/3}\end{aligned} \tag{9}$$

Theorem 2.2 *The epsilon greedy algorithm with exploration probabilities $\epsilon_t = t^{-1/3}(K\log t)^{1/3}$ achieves regret bound $\mathbb{E}[R(t)] \leq O(t^{2/3}(K\log t)^{1/3})$ for each round t.*

2.3 Upper Confidence Bound (UCB) Elimination Algorithm

2.3.1 Two-arm UCB elimination algorithm

The UCB elimination algorithm for two arms involves alternating between the arms until we find out that one arm is, with very high probability, much better than the other, at which point we abandon the inferior arm and pick the superior one forever more. We do this by defining upper and lower confidence bounds (UCB/LCB) for for the mean $\mu(a)$ of each arm a for every $t \geq 2$:

$$UCB_t(a) := \hat{\mu}_t(a) + r_t(a), \quad LCB_t(a) := \hat{\mu}_t(a) + r_t(a),$$

where $\hat{\mu}_t(a)$ is the empirical mean reward for the arm observed thus far, and the confidence radius $r_t(a)$ is defined by

$$r_t(a) = \sqrt{\frac{2\log T}{n_t(a)}},$$

where $n_t(a) = \lceil \frac{t}{2} \rceil$ is the number of times the algorithm has tried arm a thus far. Observe that the confidence radius $r_t(a)$ only changes at times when the arm a is pulled.

The UCB elimination algorithm for two players is then as follows:

1. Alternate between the two arms a, a' until the two confidence intervals no longer intersect, i.e. $UCB_t(a) < LCB_t(a')$.
2. Eliminate the arm with the lower confidence interval (a), and use the arm (a') forever more.

Let τ be the last round in which we had not invoked the elimination rule. To analyze the expected regret $\mathbb{E}[R(T)] = \mu^*(T) - \sum_{t=1}^{T} \mu(a_t)$, where a_t is the arm chosen at time t, we condition on the "clean" event ϵ that the true means for each arm fall within the confidence interval for every time step, i.e.

$$\begin{aligned}\epsilon &= \Big\{ \forall t \in [\tau],\ \text{arm } a: \quad \mu(a) \in [LCB_t(a), UCB_t(a)] \Big\} \\ &= \Big\{ \forall t \in [\tau],\ \text{arm } a: \quad |\hat{\mu}_t(a) - \mu(a)| \leq r_t(a) \Big\}.\end{aligned}$$

Then by the law of total expectation, we have

$$\mathbb{E}[R(T)] = \mathbb{E}[R(T) \mid \epsilon] \cdot \mathbb{P}[\epsilon] + \mathbb{E}[R(T) \mid \neg\epsilon] \cdot \mathbb{P}[\neg\epsilon].$$

For the first term (clean event), the eliminated arm cannot be the best arm, and so from time τ onwards we have zero expected regret. Now look at time τ, just before invoking the elimination rule. At time τ the confidence intervals of the two arms still intersect, and thus we have (in the clean event)

$$|\mu(a) - \mu(a')| \leq 2(r_\tau(a) + r_\tau(a')).$$

Then, since $n_\tau(a) = n_\tau(a') = \frac{\tau}{2}$, we have

$$r_\tau(a) = r_\tau(a') = \sqrt{\frac{4 \log T}{\tau}}.$$

Thus

$$\mathbb{E}[R(T) \mid \epsilon] = |\mu(a) - \mu(a')| \cdot \frac{\tau}{2} = \sqrt{\tau \log T} = O(\sqrt{T \log T}),$$

and since $\mathbb{P}[\epsilon] \leq 1$, we have that the first term,

$$\mathbb{E}[R(T) \mid \epsilon] \cdot \mathbb{P}[\epsilon] = O(\sqrt{T \log T}).$$

For the second term (unclean event), since the expected regret per timestep is ≤ 1, we have $\mathbb{E}[R(T) \mid \neg\epsilon] \leq T$. It then remains to bound $\mathbb{P}[\neg\epsilon]$. By Hoeffding's inequality, we have that, for any given arm a and time t,

$$\begin{aligned}\mathbb{P}\big(|\hat{\mu}_t(a) - \mu(a)| > r_j(a)\big) &\leq 2e^{-2n_t r_t(a)^2} \\ &= 2e^{-2n_t(a)\frac{2\log T}{n_t(a)}} \\ &= \frac{2}{T^4}.\end{aligned}$$

The only confidence radius and empirical mean changing at time t are those of arm a_t. (Thus, at time t, the only way that the event can be made unclean is for $\mu(a_t)$ to move outside of its confidence interval, since the confidence intervals for the other arms are unaffected.) Thus by union bound,

$$\begin{aligned}\mathbb{P}[\neg\epsilon] &\leq \bigcup_{t=1}^{\tau} \mathbb{P}\big(|\hat{\mu}_t(a_t) - \mu(a_t)| > r_t(a_t)\big) \\ &\leq 2\tau \cdot \frac{2}{T^4} \\ &= O(\frac{1}{T^3}).\end{aligned}$$

Thus the second term, $\mathbb{E}[R(T) \mid \neg\epsilon] \cdot \mathbb{P}[\neg\epsilon]$ is $O(\frac{1}{T^3})$.

Combining the two terms, we then get

$$\mathbb{E}[R(T)] = \mathbb{E}[R(T) \mid \epsilon] \cdot \mathbb{P}[\epsilon] + \mathbb{E}[R(T) \mid \neg\epsilon] \cdot \mathbb{P}[\neg\epsilon] = O(\sqrt{T \log T}).$$

Theorem 2.3 *The UCB elimination algorithm for two arms achieves an expected regret of*

$$O(\sqrt{T \log T}).$$

2.3.2 UCB elimination algorithm when there are more than two arms

When there are more than two arms, the UCB elimination algorithm is as follows:

1. Initially set all arms to “active”.
2. Try all active arms once.
3. Deaactivate all arms a for which there exists an arm a' with $UCB_t(a) < LCB_t(a')$.
4. Repeat Steps 2 and 3 until there is one arm left, then pick it for the rest of time.

The following theorem can be shown, with proof similar to the two-arm case:

Theorem 2.4 *The UCB elimination algorithm for K arms achieves an expected regret of*

$$O(\sqrt{KT \log T}).$$

2.4 Upper Confidence Bound (UCB) algorithm

The UCB algorithm involves keeping track of (empirical) confidence bounds on the true means of each arm, and always picking the arm with the highest upper confidence bound (UCB). To do this, we first try each arm once to get an empirical mean $\hat{\mu}(a)$ for each arm a. Then, for any subsequent time t, we, similar to the UCB elimination algorithm, define the upper/lower confidence bounds for every arm a:

$$UCB_t(a) := \hat{\mu}_t(a) + r_t(a), \quad LCB_t(a) := \hat{\mu}_t(a) + r_t(a),$$

where $\hat{\mu}_t(a)$ is the empirical mean reward for the arm observed thus far, and the confidence radius $r_t(a)$ is defined by

$$r_t(a) = \sqrt{\frac{2\log T}{n_t(a)}},$$

where $n_t(a)$ is the number of times the algorithm has tried arm a thus far. Observe that the confidence radius $r_t(a)$ only changes at times when the arm a is pulled.

Where the UCB algorithm differs from the UCB elimination algorithm in the previous section is that, after going one round trying each arm so that we obtain a confidence interval for each arm, the UCB algorithm just continues picking the arm that has the largest UCB at that time. The motivation for this is that there are two possible reasons why an arm might have a high UCB: either the empirical reward is large, which makes it likely that the true reward is large, or the confidence radius is large, which means that the arm has not been explored much yet. Either reason makes this arm worth trying, until its UCB drops below that of another arm.

The UCB algorithm is summarized as follows:

1. Try each arm once.
2. In each round t, pick the arm with the highest upper confidence bound, i.e. $\arg\max_a UCB_t(a)$.

The analysis for the UCB algorithm is similar to that for the UCB elimination algorithm in the last section: we condition on the "clean" event ϵ that the true means always lie within the confidence intervals, i.e.

$$\begin{aligned}\epsilon &= \Big\{\forall t \in [\tau],\ \text{arm } a: \quad \mu(a) \in [LCB_t(a), UCB_t(a)]\Big\} \\ &= \Big\{\forall t \in [\tau],\ \text{arm } a: \quad |\hat{\mu}_t(a) - \mu(a)| \leq r_t(a)\Big\},\end{aligned}$$

and again, by the law of total expectation we have

$$\mathbb{E}[R(T)] = \mathbb{E}[R(T) \mid \epsilon] \cdot \mathbb{P}[\epsilon] + \mathbb{E}[R(T) \mid \neg\epsilon] \cdot \mathbb{P}[\neg\epsilon].$$

We now bound the second term (contribution by unclean event) first, because it is very similar to the analysis for the UCB elimination algorithm. Again, the expected regret per timestep is ≤ 1, so $\mathbb{E}[R(T) \mid \neg\epsilon] \leq T$. It then remains to bound $\mathbb{P}[\neg\epsilon]$. By Hoeffding's inequality, we have again that, for any given arm a and time t,

$$\mathbb{P}\big(|\hat{\mu}_t(a) - \mu(a)| > r_j(a)\big) \leq \frac{2}{T^4}.$$

Now denote by a_t the arm chosen at time t. The only confidence radius and empirical mean changing at time t are those of arm a_t. Thus, by union bound, we have

$$\begin{aligned}\mathbb{P}[\neg\epsilon] &\leq \bigcup_{t=1}^{T} \mathbb{P}\big(|\hat{\mu}_t(a_t) - \mu(a_t)| > r_t(a_t)\big)\\ &\leq 2T \cdot \frac{2}{T^4}\\ &= O(\frac{1}{T^3}).\end{aligned}$$

Thus again, the second term (the contribution from the unclean event),

$$\mathbb{E}[R(T) \mid \neg\epsilon] \cdot \mathbb{P}[\neg\epsilon] = O(\frac{1}{T^3}). \tag{10}$$

Now we analyze the first term. (The contribution from the clean event.) Let a^* be the optimal arm. At time t, if we chose arm a_t, we have:

$$UCB_t(a_t) \geq UCB_t(a^*),$$

by virtue of the fact that we chose a_t over a^* at time t;

$$\mu(a_t) \in [LCB_t(a_t), UCB_t(a_t)] \implies \mu(a_t) + 2r_t(a_t) \geq UCB_t(a_t);$$

and

$$UCB_t(a^*) \geq \mu(a^*),$$

which always holds in the clean event. Combining the last three inequalities, we have $\mu(a_t) + 2r_t(a_t) \geq \mu(a^*)$, which can be rearranged to give

$$\mu(a^*) - \mu(a_t) \leq 2r_t(a_t).$$

But we have that

$$r_t(a_t) = \sqrt{\frac{2\log T}{n_t(a_t)}} \leq \sqrt{\frac{2\log T}{n_T(a_t)}} = r_T(a_t),$$

so combining the last two inequalities we obtain

$$\mu(a^*) - \mu(a_t) \leq 2r_T(a_t). \tag{11}$$

Thus the contribution of any arm a to the regret is

$$\big[\mu(a^*) - \mu(a)\big] n_T(a) \leq 2r_T(a) \cdot n_T(a) = 2\sqrt{2\log T n_T(a)},$$

and so we can bound the total regret in the clean event, $\mathbb{E}[R(T) \,|\, \epsilon]$, by

$$2\sqrt{2\log T}\sum_a \sqrt{n_T(a)}.$$

Since $\sqrt{\cdot}$ is concave, we can use Jensen's inequality to bound the sum

$$\sum_a \sqrt{n_T(a)} \le K\sqrt{\frac{\sum_a n_T(a)}{K}} = \sqrt{TK}.$$

Thus the total regret in the clean event, $\mathbb{E}[R(T) \,|\, \epsilon]$, is bounded by

$$2\sqrt{2\log T}\sqrt{TK} = O(\sqrt{KT\log T}).$$

It can be seen that this term far outweighs the unclean contribution in (10), and so we have the following theorem:

Theorem 2.5 *The UCB algorithm achieves regret*

$$\mathbb{E}\big[R(T)\big] \le O(\sqrt{KT\log T}).$$

The bound on the regret in Theorem 2.5 is good when arms perform close to each other, because there is no dependence on the differences in means for the different arms. But what if we are guaranteed that the second-best performing arm is significantly worse than the best performing arm a^*? It would seem that we should then be able to eliminate the bad-performing arms earlier. Can we then obtain a better bound for the regret, with reduced dependence on T?

It turns out that we can. We can rearrange (11) to get, for every arm a with $\mu(a) < \mu(a^*)$ (i.e., every arm whose picking at any time contributes to the expected regret), an upper bound on the number of times we would try arm a in the clean event:

$$n_T(a) \le \frac{8\log T}{[\mu(a^*) - \mu(a)]^2}.$$

We can then rearrange this to get the contribution of arm a to the total regret (in the clean event):

$$n_T(a)\cdot[\mu(a^*) - \mu(a)] \le \frac{8\log T}{\mu(a^*) - \mu(a)}.$$

We can then bound the total regret in the clean event,

$$\sum_{a\colon \mu(a)<\mu(a^*} n_T(a)[\mu(a^*) - \mu(a)] \le O(\log T) \sum_{a\colon \mu(a)<\mu(a^*} \frac{1}{\mu(a^*) - \mu(a)}.$$

Again, the contribution from the unclean event is negligible compared to this, and so we have the following theorem:

Theorem 2.6 *The UCB algorithm achieves regret*

$$\mathbb{E}\big[R(T)\big] \leq O(\log T)\left(\sum_{a:\mu(a)<\mu(a^*)} \frac{1}{\mu(a^*)-\mu(a)}\right).$$

It should be noted that the bounds in Theorems 2.5 and 2.6 both hold, in any instance, for the UCB algorithm. In a given scenario, the UCB algorithm will have a fixed performance (in expected regret), but one bound may be tighter than the other (and hence more useful) depending on the situation.

3 Algorithms (Adversarial Setting)

For the adversarial setting (given in the introduction), the adversary picks costs c_a^t for each arm a and each time t. Thus where a_t denotes the arm picked at time t, the regret is given by

$$R(T) = \sum_{t\in[T]} c_{a_t}^t - \min_a \sum_{t\in[T]} c_a^t,$$

and the objective of an algorithm is to minimize the expected regret, $\mathbb{E}[R(T)]$.

The only algorithm we shall analyze for the adversarial setting shall be the Exp3 algorithm.

3.1 Exp3 Algorithm

The Exp3 algorithm follows the Multiplicative Weights Update algorithm (MWUA):

```
Initialize $w_a^0 = 1$ for all $a \in [K]$.
**For** $t = 1, 2, \ldots, T$ **do**
    **Choose** arm $a$ with probability $p_a^t$ proportional to $w_a^{t-1}$.
    **Only for** the chosen arm $a$, **do**
        $w_a^t = e^{-\epsilon c_a^t / p_a^t} w_a^{t-1}$.
    **End For**
**End For**
```

For our analysis, define, for every time t, the K-dimensional vector

$$\hat{\mathbf{c}}^t = \frac{c_{a_t}^t}{p_{a_t}^t}\mathbf{e}_{a_t}.$$

Note that the a-th component of $\hat{\mathbf{c}}^t$ is given by

$$\hat{c}_a^t = \begin{cases} \frac{c_a^t}{p_a^t} & \text{if } a_t = a, \text{ i.e. arm } a \text{ is chosen at time } t, \\ 0 & \text{otherwise.} \end{cases}$$

We can then rewrite the update step in the algorithm (Step 5) as

$$w_a^t = e^{-\epsilon \hat{c}_a^t} w_a^{t-1} \quad \forall\, a \in [K].$$

Note also that

$$\begin{aligned} \mathbb{E}_{a \sim \mathbf{p}^t}\left[\hat{\mathbf{c}}^t\right] &= \sum_{a \in [K]} p_a^t \cdot \left(\frac{c_a^t}{p_a^t}\mathbf{e}_a\right) \\ &= \sum_{a \in [K]} c_a^t \mathbf{e}_a \\ &= \mathbf{c}^t, \end{aligned}$$

and so at any time t, $\hat{\mathbf{c}}^t$ is an unbiased estimator for the true cost vector $\mathbf{c}^t$.

Now define

$$L_a^t := \sum_{\tau=1}^{t} \hat{c}_a^\tau,$$

so that

$$w_a^t = e^{-\epsilon L_a^t}.$$

Define a potential function Φ_t by

$$\Phi_t := -\frac{1}{\epsilon} \log \sum_{a \in [K]} w_a^{t-1}.$$

We have

$$\begin{aligned} \Phi_{t+1} - \Phi_t &= -\frac{1}{\epsilon} \log \frac{w_a^t}{w_a^{t-1}} \\ &= -\frac{1}{\epsilon} \log \frac{w_a^{t-1} \cdot e^{-\epsilon \hat{c}_a^t}}{w_a^{t-1}} \\ &= -\frac{1}{\epsilon} \log \mathbb{E}_{a \sim \mathbf{p}^t}\left[e^{-\epsilon \hat{c}_a^t}\right]. \end{aligned}$$

Then, because $e^{-x} \leq 1 - x + \frac{1}{2}x^2$, we have

$$\begin{aligned} \Phi_{t+1} - \Phi_t &= -\frac{1}{\epsilon} \log \mathbb{E}_{a \sim \mathbf{p}^t}\left[e^{-\epsilon \hat{c}_a^t}\right] \\ &\geq -\frac{1}{\epsilon} \mathbb{E}_{a \sim \mathbf{p}^t}\left[1 - \epsilon \hat{c}_a^t + \frac{1}{2}\epsilon^2 (\hat{c}_a^t)^2\right] \\ &= -\frac{1}{\epsilon} \log \left(1 - \mathbb{E}_{a \sim \mathbf{p}^t}\left[\epsilon \hat{c}_a^t - \frac{1}{2}\epsilon^2 (\hat{c}_a^t)^2\right]\right), \end{aligned}$$

and since $-x \geq \log(1-x)$ we have

$$\begin{aligned}\Phi_{t+1} - \Phi_t &\leq -\frac{1}{\epsilon} \log \left(1 - \mathbb{E}_{a \sim \mathbf{p}^t} \left[\epsilon \hat{c}_a^t - \frac{1}{2} \epsilon^2 (\hat{c}_a^t)^2 \right] \right) \\ &\geq \frac{1}{\epsilon} \mathbb{E}_{a \sim \mathbf{p}^t} \left[\epsilon \hat{c}_a^t - \frac{1}{2} \epsilon^2 (\hat{c}_a^t)^2 \right] \\ &= \mathbb{E}_{a \sim \mathbf{p}^t} \left[\hat{c}_a^t \right] - \frac{1}{2} \epsilon \mathbb{E}_{a \sim \mathbf{p}^t} \left[(\hat{c}_a^t)^2 \right] \\ &= \sum_{a \in [K]} p_a^t \hat{c}_a^t - \frac{1}{2} \epsilon \sum_{a \in [K]} p_a^t (\hat{c}_a^t)^2 \\ &= \sum_{a \in [K]} p_a^t \hat{c}_a^t - \frac{1}{2} \epsilon \sum_{a \in [K]} c_a^t \hat{c}_a^t .\end{aligned}$$

Taking expectation, we have

$$\mathbb{E}[\Phi_{t+1} - \Phi_t] \geq \sum_{a \in [K]} p_a^t c_a^t - \frac{1}{2} \epsilon \sum_{a \in [K]} (c_a^t)^2 \geq \sum_{a \in [K]} p_a^t c_a^t - \frac{K\epsilon}{2}.$$

Then, taking the telescopic sum, we have

$$\Phi_{T+1} - \Phi_1 \geq \sum_{t=1}^{T} \sum_{a \in [K]} p_a^t c_a^t - \frac{KT\epsilon}{2}. \tag{12}$$

On the other hand, we have $\Phi_1 = -\frac{1}{\epsilon} \log K$, so, where a^* is the optimal arm, we have

$$\mathbb{E}[\Phi_{T+1} - \Phi_1] \leq \mathbb{E}[L_{a^*}^T - (-\frac{1}{\epsilon} \log K)] = \sum_{t=1}^{T} c_{a^*}^t + \frac{1}{\epsilon} \log K. \tag{13}$$

Combining (12) and (13), we then have that

$$\mathbb{E}[R(T)] = \sum_{t=1}^{T} \sum_{a \in [K]} p_a^t c_a^t - \sum_{t=1}^{T} c_{a^*}^t \leq \frac{KT\epsilon}{2} + \frac{1}{\epsilon} \log K.$$

We can then choose $\epsilon = \sqrt{\frac{2 \log K}{TK}}$, from which it would follow that the expected regret $\mathbb{E}[R(T)]$ is $O(\sqrt{TK \log K})$.

Theorem 3.1 *The Exp3 algorithm with* $\epsilon = \sqrt{\frac{2 \log K}{TK}}$ *achieves an expected regret of*

$$O(\sqrt{TK \log K}).$$

4 Conclusion

In these two lectures we were introduced to framework of multi-armed bandits, both the stochastic setting, where each arm gives a reward according to a fixed distribution, and the adversarial setting, where each arm at each time comes with a different cost, determined by an adversary. In both settings, the aim was to design an algorithm that minimizes the expected regret.

For the stochastic setting, we analyzed the Explore First algorithm, the Epsilon Greedy algorithm, the UCB Elimination algorithm, and the UCB algorithm. For large times T, it was found that the UCB elimination algorithm and the UCB algorithm had a favorable bound of $O(\sqrt{KT\log T})$ on the expected regret.

For the adversarial setting, the Exp3 algorithm was analyzed. The expected regret was found to be $O(\sqrt{TK\log K})$.