Statistical Techniques in Robotics (16-831, S22) Lecture #12 (Monday, February 28)

# Bandit: Explore-Exploit

*Lecturer: Kris Kitani* *Scribes: Alex Pletta, Benjamin Younes*

## 1 Review

In the last lectures, we previously learned about multi-armed bandits and learned how they can be applied for a variety of applications. Bandits are used for partial feedback, as opposed to Prediction With Expert Advice (PWEA) which allows the learner to access all possible loss for each possible decision.

A bandit is commonly modeled as a slot machine where, if selected (the bandit's "arm" is pulled), it can provide a reward based on an underlying probability distribution. Multi-Armed Bandit (MAB) problems contain more than one source for getting reward (i.e. multiple bandits).

Previously, we discussed the problem formulation of multi-armed bandits, but did not address any solution methods. Commonly, multi-armed bandit problems seek to "pull" the sequence of arm(s) that deliver the highest expected reward. In some cases this simply involves finding the optimal bandit.

### 1.1 Multi-Armed Bandit

We recall the characteristics of the Multi-Armed Bandit problem [1]:

1. **One-Shot**: The MAB problem fundamentally has no concept of state. This means that any action taken by the learner at any time step has no effect on future probabilities. This means that each singular action leads to only one singular reward. This makes the problem One-Shot.
2. **Exhaustive**: The MAB problem allows us to pull all of the levers over the course of the time series. Pulling one lever does not disqualify us from pulling any future lever(s). This makes the MAB problem exhaustive.
3. **Evaluative**: In the MAB problem, we receive instant feedback on our sampled reward. Functionally, this allows us to close the loop on weight updates every time step. This makes the MAB problem evaluative.

### 1.2 Types of Bandits

There are many different types of bandits, depending on environment and context assumptions, as highlighted in Table 1. In the scope of this lecture, we will focus Context-free assumptions in a Stochastic environment. This means that each arm has a static reward distribution and that each pull gives a single sample from that distribution. We will be discussing Explore-Exploit and UCB.

Table 1: Different types of bandits covered in this course.

| | **Context-free** | **Contextual** |
|---|---|---|
| **Stochastic environment** | *Explore-Exploit, UCB* | linUCB |
| **Adversarial environment** | EXP3 | EXP4 |

### 1.3 Key Assumptions for Multi-Armed Bandit

1. There is an underlying probability distribution which the learner does not have access to. This distribution determines reward for a single "arm pull".
2. The learner does not have access to any losses for any bandit that was not selected.
3. The learner can only pull one arm at each step, and receives a single reward.
4. The learner knows how many total pulls (typically referred to as $T$) at the start of the problem.
5. The learner has total access to sampling each bandit, and knows how many bandits there are. The number of bandits remains constant over all time steps.

## 2 Summary

In today's lecture, we will discuss how multi-armed bandit problems can be solved using explore-exploit algorithms. A natural question for such a problem is how to choose which arm to pull, especially considering the potential mean and variance of a certain bandit?

Because this is a hidden information problem, we must first characterize the expected reward from each bandit in order to find which bandit(s) are worth pursuing.

### 2.1 Explore-Exploit Algorithm

#### 2.1.1 Notation Definition

| | |
|---|---|
| $a_k \in A$ | action (arm)$\lvert\mathcal{A}\rvert = K$ |
| $r^{(t)}$ | reward (receive for given time $t$) |
| $M$ | Exploration steps (for each pull of an arm) |
| $\hat{\mu}_k$ | Expected reward for the $k$th arm |

The Explore-Exploit Algorithm [2] is listed as Algorithm 1 below.

**Algorithm 1** Explore-Exploit Algorithm

**for** $k = 1 \rightarrow K$ ▷ Explore over arms **do**
  **for** $m = 1 \rightarrow M$ ▷ Iterate through all explore steps **do**
    $a = k$ ▷ Select the arm
    RECEIVE$(r)$ ▷ Get the reward.
    $\hat{\mu}_k = \hat{\mu}_k + r/M$ ▷ Update the estimated mean for that arm.
  **end for**
  **for** $t = KM \rightarrow T$ ▷ Exploit best arm **do**
    $a^{(t)} = \text{argmax}_{k'} \hat{\mu}_k$ ▷ Select arm with maximum estimated reward
  **end for**
**end for**

## 2.2 Regret Bound for Explore-Exploit

As we will show, the regret bound for Explore-Exploit is:

$$R_{\text{explore-exploit}} = O(K^{1/3}T^{2/3}) \tag{1}$$

where $K$ = the number of arms and $T$ = number of total time steps. This bound is no-regret, because the bound is sublinear with respect to $T$.

Before we can do the proof we need to understand Hoeffding's Inequality.

### 2.2.1 Hoeffding's Inequality

Hoeffding's Inequality comes from a paper written in 1963 [3]. This inequality fundamentally looks at the closeness of an empirical mean $\hat{\mu}$ to a true mean $\mu$. The difference between the two means is bounded by a confidence interval $\epsilon$ which decreases as a function of data received. Hoeffding's Inequality uses this concept to formulate a bounded probability.

**Theorem 1.** *Hoeffding's Inequality Consider a one-dimension distribution $\nu$ with expectation $\mu$, where any sample $r$ $\nu$ is bounded such that $r \in [0, 1]$. Given $T$ i.i.d samples $\{r^{(t)}\}_{t=1}^{T}$ we have that for any $\epsilon$:*

$$p(|\sum_{t=1}^{T} \frac{r^{(t)}}{T} - \mu| \geq \epsilon) \leq 2e^{-2T\epsilon^2} \tag{2}$$

*This inequality captures the simplex probability of the empirical mean falling **outside** of the confidence interval on the true mean.*

*If we want the probability to be less than a specific value, we can let the RHS be assigned to a single threshold variable $\delta$. Let:*

$$\delta = 2e^{-2T\epsilon^2} \tag{3}$$

*Now we want to consider what the optimal confidence interval $\epsilon$ is for a given threshold $\delta$. Solving for $\epsilon$:*

$$\delta = 2e^{-2T\epsilon^2} \tag{4}$$
$$\log(\delta) = \log(2e^{-2T\epsilon^2}) \tag{5}$$
$$\log(\delta) = \log(2) - 2T\epsilon^2 \tag{6}$$
$$\log(\delta) - \log(2) = -2T\epsilon^2 \tag{7}$$
$$-\log(\delta) + \log(2) = 2T\epsilon^2 \tag{8}$$
$$\log(2/\delta) = 2T\epsilon^2 \tag{9}$$
$$\frac{\log(2/\delta)}{2T} = \epsilon^2 \tag{10}$$
$$\sqrt{\frac{\log(2/\delta)}{2T}} = \epsilon \tag{11}$$

*We can use this formulation of $\epsilon$ to calculate desired confidence interval based on the desired probability threshold $\delta$. When this equality is true, then we can say the following.*

$$\sqrt{\frac{\log(2/\delta)}{2T}} = \epsilon \tag{12}$$
$$p(|\sum_{t=1}^{T} \frac{r^{(t)}}{T} - \mu| \geq \epsilon) \leq \delta \tag{13}$$

*We can re-formulate this probability to represent the probability of the empirical mean falling **inside** the confidence interval of the true mean. This is due to the fact that the entire confidence interval equals one over the entire probability distribution.*

$$p(|\sum_{t=1}^{T} \frac{r^{(t)}}{T} - \mu| \leq \epsilon) < 1 - \delta \tag{14}$$

*We can say the **event** $\{|\hat{\mu} - \mu\| < \epsilon\}$ holds with probability $1 - \delta$.*

### 2.2.2 Regret Bound Proof for Explore-Exploit Algorithm

We now have the tools to prove the regret bound for the explore-exploit algorithm.

This is broken down into the following steps:

1. Define regret bound on exploration phase
2. Define regret bound on exploitation phase
   (a) Define potential function

(b) Find upper bound for the potential function

(c) Find lower bound for the potential function

(d) Combine the upper and lower bounds

(e) Solve for the final bound

#### 2.2.2.1 Step 1: Define regret bound on exploration phase

By the algorithm, exploration is comprised of of $KM$ pulls. Therefore, a trivial upper bound is when all pulls result in zero reward. This upper bounds the regret as:

$$R_{\text{explore}} \leq O(KM) \tag{15}$$

#### 2.2.2.2 Step 2a: Define potential function for exploitation phase

Let the potential function be an inequality defined as:

$$\hat{\mu}_{\hat{k}} \geq \hat{\mu}_{k^\star} \tag{16}$$

where $\hat{\mu}_{\hat{k}}$ is equal to the highest estimated reward mean, and $\hat{\mu}_{k^\star}$ is the true highest reward mean.

#### 2.2.2.3 Step 2b: Upper bound for exploitation phase

To upper bound the potential function, we must bound the highest estimated reward mean $\hat{\mu}_{\hat{k}}$. From Hoeffding's Inequality, we can use the confidence bound from the difference between the estimated and true mean for the $\hat{k}$ arm, and then substitute the optimal value for $\epsilon$. Note that in case the equation for $\epsilon$ uses $M$ instead of $T$ because that is the number of iterations performed in the exploration phase.

$$\hat{\mu}_{\hat{k}} \leq \mu_{\hat{k}} + \epsilon \tag{17}$$

$$\leq \mu_{\hat{k}} + \sqrt{\frac{\log(2/\delta)}{2M}} \tag{18}$$

#### 2.2.2.4 Step 2c: Lower bound for exploitation phase

To bound the lower bound the potential function, we must now bound the true highest reward mean $\hat{\mu}_{k^\star}$. We can again use Hoeffding's Inequality, but this time with the lower confidence interval.

$$\hat{\mu}_{k^\star} \geq \mu_{k^\star} - \epsilon \tag{19}$$

$$\geq \mu_{k^\star} - \sqrt{\frac{\log(2/\delta)}{2M}} \tag{20}$$

#### 2.2.2.5 Step 2d: Combine the upper and lower bounds for exploitation phase

We start with the original potential function and then substitute the upper and lower bounds:

$$\hat{\mu}_{\hat{k}} \geq \hat{\mu}_{k^\star} \tag{21}$$

$$\mu_{\hat{k}} + \sqrt{\frac{\log(2/\delta)}{2M}} \geq \hat{\mu}_{\hat{k}} \geq \hat{\mu}_{k^\star} \geq \mu_{k^\star} - \sqrt{\frac{\log(2/\delta)}{2M}} \tag{22}$$

$$\mu_{\hat{k}} + \sqrt{\frac{\log(2/\delta)}{2M}} \geq \mu_{k^\star} - \sqrt{\frac{\log(2/\delta)}{2M}} \tag{23}$$

#### 2.2.2.6 Step 2e: Solve for the final bound

Now we will rearrange to solve for the final bound on the regret.

$$\mu_{k^\star} - \mu_{\hat{k}} \leq 2\sqrt{\frac{\log(2/\delta)}{2M}} \tag{24}$$

The regret for all time steps of exploitation is the summation of difference in means over all the time steps. Since exploitation occurs from step $KM$, when exploration finishes, until $T$, then the RHS of the equation will be multiplied by that many steps (i.e. $T - KM$). This is an upper bound on the exploitation regret.

$$R_{\text{exploit}} = \sum_{t=KM+1}^{T} (\mu_{k^\star}^{(t)} - \mu_{\hat{k}}^{(t)}) \leq (T - KM)2\sqrt{\frac{\log(2/\delta)}{2M}} \tag{25}$$

Using Big-$O$ notation, this inequality is on the order of:

$$R_{\text{exploit}} \leq O(2(T - KM)\sqrt{\frac{1}{M}}) \tag{26}$$

Finally, to get the full regret bound on explore-exploit we simply combine the respective explore and exploit bounds:

$$R_{\text{explore-exploit}} = R_{\text{explore}} + R_{\text{exploit}} \tag{27}$$

$$R_{\text{explore-exploit}} = O(KM) + O(2(T - KM)\sqrt{\frac{1}{M}}) \tag{28}$$

$$R_{\text{explore-exploit}} \leq KM + 2T\sqrt{\frac{1}{M}} \tag{29}$$

This is an elegant bound, but it is not yet no regret. In order to make the algorithm no-regret, we must solve for an optimal $M$.

#### 2.2.2.7 Solving for optimal $M$ to have no-regret upper bound

We can solve for the optimal $M$ by setting the derivative of regret with respect to $M$, to zero.

$$0 = \frac{\partial}{\partial M}\{KM + 2T\sqrt{\frac{1}{M}}\} \tag{30}$$

$$= K - 2T\frac{1}{2}M^{-3/2} \tag{31}$$

$$TM^{-3/2} = K \tag{32}$$

$$M^{-3/2} = \frac{K}{T} \tag{33}$$

$$M = (\frac{T}{K})^{2/3} \tag{34}$$

Note that this solution for $M$ relies on the assumption that we know how many $T$ time steps will be performed.

We can now plug the optimal $M$ back into the regret bound to get:

$$R_{\text{explore-exploit}} = R_{\text{explore}} + R_{\text{exploit}} \tag{35}$$

$$R_{\text{explore-exploit}} \leq KM + 2T\sqrt{\frac{1}{M}} \tag{36}$$

$$= K(\frac{T}{K})^{2/3} + 2T\sqrt{(\frac{T}{K})^{-2/3}} \tag{37}$$

$$= K(\frac{T}{K})^{2/3} + 2T(\frac{T}{K})^{-1/3} \tag{38}$$

$$= K^{1/3}T^{2/3} + 2T^{2/3}K^{1/3} \tag{39}$$

$$= 3K^{1/3}T^{2/3} \tag{40}$$

In Big-$O$ notation, the final no-regret bound is then:

$$R_{\text{explore-exploit}} \leq O(K^{1/3}T^{2/3}) \tag{41}$$

We can see that this is sublinear with respect to $T$ and thus no-regret.

## 2.3 Upper Confidence Bound

We saw in the previous section that solving the MAB problem balances a tradeoff between exploration and exploitation. The Upper Confidence Bound algorithm provides a different of making this balance. The algorithm does so by starting to exploit earlier and continuing to explore later than the more naive Explore-Exploit algorithm introduced earlier.

### 2.3.1 Notation Definition

| | | |
|---|---|---|
| $k^{(t)}$ | bandit arm pulled at time step $t$ | (42) |
| $r^{(t)}$ | reward feedback at time step $t$ | (43) |
| $T_k^{(t)}$ | number of times that a specific arm $k$ was pulled up to time step $t\mu_k$ | the underlying mean reward functio (44) |
| $\hat{\mu}_k^{(t)}$ | the estimated mean reward function of arm $k$ at time step $t$ | (45) |
| $T$ | Total number of time steps | (46) |

We will now take a look at the algorithm for the Upper Confidence Bound, as Algorithm 2:

**Algorithm 2** UCB Algorithm

**for** $t = 1 \rightarrow T$ ▷ Loop through all time history **do**
  **if** $t \leq K$ **then**
    $k = t$ ▷ Pull every arm at least once.
  **else**
    $k = \text{argmax}_{k'}(\hat{\mu}_{k'} + \sqrt{\frac{\log(2T/\delta')}{2T_{k'}^{(t)}}})$ ▷ Select the arm based on the summation of current mean estimate and current upper confidence bound.
  **end if**
  RECEIVE($r^{(t)}$) ▷ Get reward from nature.
  $T_k^{(t)} = T_k^{(t')} + 1$ ▷ Update pull counter for that arm.
  $\hat{\mu}_k = \frac{1}{T_k^{(t)}}(\hat{\mu}_k(T_k^{(t)} - 1) + r^{(t)})$ ▷ Update estimated mean for the arm $k$.
**end for**

Before going on, let's take a closer look at the arm selection step.

$$k = \text{argmax}_{k'}(\hat{\mu}_{k'} + \sqrt{\frac{\log(2T/\delta')}{2T_{k'}^{(t)}}}) \tag{47}$$

This selection step tries to select the mean with highest estimated reward while introducing the upper confidence bound "bonus term". The upper confidence bound takes into account the number of pulls up to now $T_{k'}^{(t)}$ and the desired probability constant $\delta'$. This formulation is a result from Hoeffding's inequality and Union Bound. We will now further explain what this Union Bound means.

### 2.3.2 Union Bound

The Union Bound comes is derived from Boole's inequality. It is the probability of two separate events happening which may share underlying event probability. Functionally, this allows for the removal of double counting during a probability union.

We know the union of two probabilities is:

$$p(x_1 \cup x_2) = p(x_1) + p(x_2) - p(x_1 \cap x_2) \tag{48}$$

Boole's inequality then creates the bound:

$$p(x_1 \cup x_2) \leq p(x_1) + p(x_2) \tag{49}$$

The probability of one event or the other occurring is less than or equal to the sum of both events happening independently.

We now apply this concept to our scenario. Recall that for a single time step, the probability that a estimated mean will be outside of the confidence interval $\delta$ is bounded by the following equation:

$$p(|\hat{\mu}_k^{(t)} - \mu_k| > \sqrt{\frac{\log(2/\delta)}{2T_k^{(t)}}}) \leq \delta \tag{50}$$

Over a sequence of time steps, we can express this probability as a union of probabilities:

$$p(\cup_{t=1}^{T}\{|\hat{\mu}_k^{(t)} - \mu_k| > \sqrt{\frac{\log(2/\delta)}{2T_k^{(t)}}}\}) \tag{51}$$

Following from Boole's inequality, the probability of either the estimate falling outside of the confidence interval or not is bounded by the probability of all estimates falling outside of that confidence interval. This can be written as the following:

$$p(\cup_{t=1}^{T}\{|\hat{\mu}_k^{(t)} - \mu_k| > \sqrt{\frac{\log(2/\delta)}{2T_k^{(t)}}}\}) \leq \sum_{t=1}^{T} p(|\hat{\mu}_k^{(t)} - \mu_k| > \sqrt{\frac{\log(2/\delta)}{2T_k^{(t)}}}) \tag{52}$$

Recall that the probability of a single time step is bounded by $\delta$, so the sum over all time steps $T$ is bounded by $T\delta$:

$$\sum_{t=1}^{T} p(|\hat{\mu}_k^{(t)} - \mu_k| > \sqrt{\frac{\log(2/\delta)}{2T_k^{(t)}}}) \leq T\delta = \delta' \tag{53}$$

With this new $\delta'$, we can rewrite the confidence interval, as derived earlier, for the union event as:

$$\epsilon' = \sqrt{\frac{\log(2T/\delta')}{2T_k^{(t)}}} \tag{54}$$

We now have the tools we need to prove the regret bound on the UCB algorithm!

### 2.3.3 Regret Bound of UCB

We will show that the UCB algorithm has a no-regret bound because the bound is sublinear in $T$. The regret bound is:

$$R_{\text{UCB}} = O(2\sqrt{\log(2T/\delta)}\sqrt{KT}) \tag{55}$$
$$= O(\sqrt{KT}) \tag{56}$$

Note that in this formulation $K$ is still the number of arms and $T$ is the selected time horizon.

The proof is broken down into the following steps:

1. Define base inequality for the potential function
2. Establish the upper bound using the confidence bound
3. Establish the lower bound using the confidence bound
4. Combine the confidence bounds
5. Solve for the final regret bound

#### 2.3.3.1 Step 1: Define base inequality for the potential function

The base inequality is derived from the observation that when the UCB algorithm makes a mistake, it will select a non-optimal arm. When this mistake is made, then it was because the estimated mean and respective upper confidence bound was greater than the estimated mean of the optimal arm with respective upper confidence bound. Recall that the estimated non-optimal mean and estimated optimal mean expressions are derived from Hoeffding's inequality and the Union Bound in the previous steps.

$$\hat{\mu}_k^{(t)} + \sqrt{\frac{\log(2T/\delta')}{2T_k^{(t)}}} \geq \hat{\mu}_{k^\star}^{(t)} + \sqrt{\frac{\log(2T/\delta')}{2T_{k^\star}^{(t)}}} \tag{57}$$

This base inequality serves as the potential function for the remaining steps of the proof.

#### 2.3.3.2 Step 2: Establish the upper bound using the confidence bound

To establish an upper bound on the potential function, we now need to establish an upper bound on the estimated non-optimal mean and respective confidence interval.

Recall from Hoeffding's inequality that $\{|\hat{\mu}^{(t)} - \mu| < \epsilon\}$ holds with probability $1 - \delta$. In other words, the estimated mean for the non-optimal bound is upper bounded by the true mean for the non-optimal bound and the respective upper confidence bound. This theorem allows us to bound the mean, and then we can add the upper confidence to both sides to arrive at the final upper bound on the potential function:

$$\mu_k + \sqrt{\frac{\log(2T/\delta')}{2T_k^{(t)}}} \geq \hat{\mu}_k^{(t)} \tag{58}$$

$$\mu_k + 2\sqrt{\frac{\log(2T/\delta')}{2T_k^{(t)}}} \geq \hat{\mu}_k^{(t)} + \sqrt{\frac{\log(2T/\delta')}{2T_k^{(t)}}} \tag{59}$$

#### 2.3.3.3 Step 3: Establish the lower bound using the confidence bound

Similar to the upper bound derivation, we can apply the same knowledge of Hoeffding's to the lower bound derivation. In this case, the true mean of the optimal arm *subtracted* by the upper confidence bound is the lower bound on the estimated mean of the optimal arm. We can then again add the confidence bound to both sides to form the lower bound on the overall potential function. Note that on the RHS that the summation of negative and positive confidence bounds cancel out leaving only the true mean of the optimal arm.

$$\hat{\mu}_{k^\star}^{(t)} \geq \mu_{k^\star} - \sqrt{\frac{\log(2T/\delta')}{2T_{k^\star}^{(t)}}} \tag{60}$$

$$\hat{\mu}_{k^\star}^{(t)} + \sqrt{\frac{\log(2T/\delta')}{2T_{k^\star}^{(t)}}} \geq \mu_{k^\star} - \sqrt{\frac{\log(2T/\delta')}{2T_{k^\star}^{(t)}}} + \sqrt{\frac{\log(2T/\delta')}{2T_{k^\star}^{(t)}}} \tag{61}$$

$$\hat{\mu}_{k^\star}^{(t)} + \sqrt{\frac{\log(2T/\delta')}{2T_{k^\star}^{(t)}}} \geq \mu_{k^\star} \tag{62}$$

#### 2.3.3.4 Step 4: Combine the confidence bounds

We can now combine the upper and lower bounds together by relating them through the potential function:

$$\hat{\mu}_k^{(t)} + \sqrt{\frac{\log(2T/\delta')}{2T_k^{(t)}}} \geq \hat{\mu}_{k^\star}^{(t)} + \sqrt{\frac{\log(2T/\delta')}{2T_{k^\star}^{(t)}}} \tag{63}$$

$$\mu_k + 2\sqrt{\frac{\log(2T/\delta')}{2T_k^{(t)}}} \geq \hat{\mu}_k^{(t)} + \sqrt{\frac{\log(2T/\delta')}{2T_k^{(t)}}} \geq \hat{\mu}_{k^\star}^{(t)} + \sqrt{\frac{\log(2T/\delta')}{2T_{k^\star}^{(t)}}} \geq \mu_{k^\star} \tag{64}$$

$$\mu_k + 2\sqrt{\frac{\log(2T/\delta')}{2T_k^{(t)}}} \geq \mu_{k^\star} \tag{65}$$

#### 2.3.3.5 Step 5: Solve for the final regret bound

We can rearrange the final inequality as:

$$\mu_k + 2\sqrt{\frac{\log(2T/\delta')}{2T_k^{(t)}}} \geq \mu_{k^\star} \tag{66}$$

$$\mu_{k^\star} - \mu_k \leq 2\sqrt{\frac{\log(2T/\delta')}{2T_k^{(t)}}} \tag{67}$$

To get the final UCB regret bound, we can use the sum the difference between the optimal and selected mean distributions. Note that this step involves considerable algebraic steps. These notes will walk through and explain the large steps; we leave any intermediary steps to the concerned reader. In other words:

$$R_{\text{UCB}} = \sum_{t=1}^{T}(\mu_{k^\star} - \mu_{k^{(t)}}) \tag{68}$$

$$\leq \sum_{t=1}^{T} 2\sqrt{\frac{\log(2T/\delta')}{2T_k^{(t)}}} \tag{69}$$

We now factor the constants out of the summation.

$$R_{\text{UCB}} \leq \frac{1}{2}\sqrt{\log(2T/\delta')}\sum_{t=1}^{T}\sqrt{\frac{1}{T_k^{(t)}}} \tag{70}$$

We then sum over all arms:

$$R_{\text{UCB}} \leq \frac{1}{2}\sqrt{\log(2T/\delta')}\sum_{t=1}^{T}\sum_{j=1}^{K}\mathbb{1}[k^{(t)} == j]\sqrt{\frac{1}{T_j^{(t)}}} \tag{71}$$

Next we simply swap the order of the summation:

$$R_{\text{UCB}} \leq \frac{1}{2}\sqrt{\log(2T/\delta')}\sum_{j=1}^{K}\sum_{t=1}^{T}\mathbb{1}[k^{(t)} == j]\sqrt{\frac{1}{T_j^{(t)}}} \tag{72}$$

In this step we now sum $t$ over the time decay factor $T_j^{(t)}$, in place of summing the time decay factor $T_j^{(t)}$ over $t$. This reformulation condenses the indicator function into the sum:

$$R_{\text{UCB}} \leq \frac{1}{2}\sqrt{\log(2T/\delta')}\sum_{j=1}^{K}\sum_{t=1}^{T_j^{(t)}}\sqrt{\frac{1}{t}} \tag{73}$$

Note that here we can apply a useful inequality for integral bounds on summations:

$$\sum_{t=1}^{T} \sqrt{\frac{1}{t}} = 1 + \sum_{t=2}^{T} \sqrt{\frac{1}{t}} \tag{74}$$

$$\leq 1 + \int_{t=1}^{T} \sqrt{\frac{1}{t^2}}\, dt \tag{75}$$

$$= 1 + 2\sqrt{T} - 2 \tag{76}$$

$$= 2\sqrt{T} - 1 \tag{77}$$

$$\leq 2\sqrt{T} \tag{78}$$

Using this inequality, we then rewrite our our inequality as:

$$R_{\text{UCB}} \leq \frac{1}{2}\sqrt{\log(2T/\delta')} \sum_{j=1}^{K} 2\sqrt{T_j^{(T)}} \tag{79}$$

$$R_{\text{UCB}} \leq \sqrt{\log(2T/\delta')} \sum_{j=1}^{K} \sqrt{T_j^{(T)}} \tag{80}$$

$$\tag{81}$$

We will now use Jensen's inequality.

**Theorem 2.** *Jensen's Inequality*

*Jensen's inequality upper bounds a sum of a function by a function of the sum. This is true when $f(x)$ is a concave function.*

$$\sum_n p_n f(x_n) \leq f(\sum_n p_n x_n) \tag{82}$$

*In the context of our scenario:*

$$\sum_k \frac{1}{K}\sqrt{T_k} \leq \sqrt{\sum_k \frac{1}{K} T_k} \tag{83}$$

Applying Jensen's inequality to our scenario:

$$R_{\text{UCB}} \leq \sqrt{\log(2T/\delta')} \sum_{j=1}^{K} \sqrt{T_j^{(T)}} \tag{84}$$

$$\leq \sqrt{\log(2T/\delta')} K \frac{1}{K} \sum_{j=1}^{K} \sqrt{T_j^{(T)}} \tag{85}$$

$$\leq \sqrt{\log(2T/\delta')} K \sqrt{\sum_{j=1}^{K} \frac{1}{K} T_j^{(T)}} \tag{86}$$

$$= \sqrt{\log(2T/\delta')} K \sqrt{\frac{T}{K}} \tag{87}$$

$$= \sqrt{\log(2T/\delta')} \sqrt{TK} \tag{88}$$

$$\sim O(\sqrt{TK}) \tag{89}$$

We see that in Big-$O$ notation, the regret bound on the UCB algorithm is sub-linear in $T$. This is a no-regret algorithm!

### 2.4 Comparing Explore-Exploit and UCB algorithms

We saw that the bounded regret for Explore-Exploit and UCB algorithms were:

$$R_{\text{Explore-Exploit}} \leq O(K^{1/3} T^{2/3}) \tag{90}$$

$$R_{\text{UCB}} \leq O(\sqrt{TK}) \tag{91}$$

The regret bound for the UCB algorithm is better than the regret bound for the Explore-Exploit algorithm in terms of reliance on $T$. This is because the UCB algorithm better balances exploration and exploitation than the Explore-Exploit algorithm.

We hope you enjoyed the lecture!

## References


[1] H. Robbins, "Some aspects of the sequential design of experiments," *Bulletin of the American Mathematical Society*, vol. 58, no. 5, pp. 527 – 535, 1952.

[2] T. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in Applied Mathematics*, vol. 6, no. 1, pp. 4–22, 1985.

[3] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 13–30, 1963.