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

Bandits (Explore-Exploit), UCB

Lecturer: Kris Kitani *Scribes: Ashwin Misra, Mansi Agarwal*

1 Review

In the previous lecture, we learned about Adaboost algorithm. We also saw the PAC Learning framework to achieve a good performance with a high probability. We completed the topic on online supervised learning and started understanding the Multi-Armed Bandit problem.

1.1 Adaboost

1.1.1 PAC Learning Model

PAC stands for "Probably Approximately Correct", and it is a generic framework to analyse the sample complexity of a learning algorithm in order to achieve good performance with high probability. The PAC learning model is a theoretical framework to answer two questions:

1. What is the optimal dataset size to obtain good generalization?
2. What is the computational cost of learning?

1.2 Adaboost

Algorithm 1 discusses the steps. The algorithm is run for multiple rounds and in each round, the algorithm maintains weights for each training data point. Weights signify the importance of that data point and the learner should avoid making mistakes on instances with higher weights. In each round, the algorithm learns a weak learner on the weighted dataset. Then, the algorithm adjust the weights of the data points based on the error of the current weak learner. In the end, the algorithm returns a combination of the weak learners whose weight is determined by the mistake that it makes.

Algorithm 1 Adaboost

Require: $\mathbf{D} = \{\mathbf{x}_n, y_n\}_{n=1}^N, \{\mathbf{w}_n^{(0)}\}_{n=1}^N, T$

1: **for** $t = 1, \cdots, T$ **do**
2: $\quad \mathbf{p}^{(t)} = \mathbf{w}^{t-1} / \sum_n \mathbf{w}_n^{t-1}$
3: $\quad h^t = \text{WEAKLEARNER}(\mathbf{D}, \mathbf{p}^{(t)})$
4: $\quad \epsilon^{(t)} = \sum_n p_n^t |h^t(\mathbf{x}_n) - y_n|$
5: $\quad \beta^{(t)} = \epsilon^{(t)}(1 - \epsilon^{(t)})$
6: $\quad \mathbf{w}_n^t = \mathbf{w}_n^{(t-1)} \beta^{1-|h^{(t)}(\mathbf{x}_n^{(t)}) - y_n^{(t)}|} \; \forall n$
7: **end for**
8: $h_F(\mathbf{x}) = 1\{\sum_{t=1}^T (\log(\frac{1}{\beta^{(t)}}) h^{(t)}(\mathbf{x})) \geq \frac{1}{2} \sum_{t=1}^T (\log(\frac{1}{\beta^{(t)}})\}$

1.2.1 Error bound of Adaboost

Theorem 1 (Mistake bound of Adaboost)**.** *Let ϵ be the error made by h_F. We have*

$$\epsilon \leq 2^T \prod_{t=1}^{T} \sqrt{\epsilon_t(1-\epsilon_t)}.$$

1.3 Multi-Armed Bandits

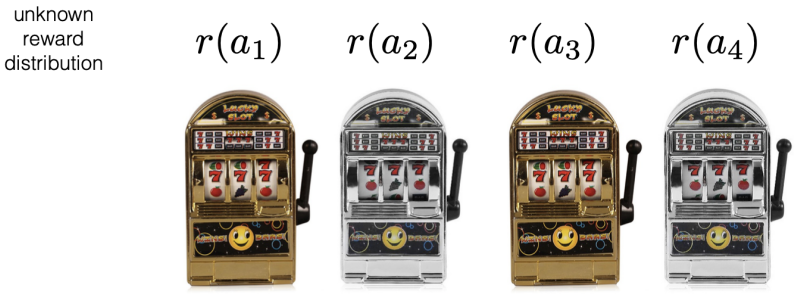

Figure 1: Multi-Armed Bandits.

Multi-armed bandits have the following characteristics:

1. **One-Shot Feedback**: One action from the player leads to one reward and selecting any particular action does not change the state at the next time-step.
2. **Exhaustive Feedback**: The player can pull all the arms for the duration of the game, therefore actions are finite. The state is static and finite.
3. **Evaluative Feedback**: The player receives a reward sampled from the underlying unknown reward distributions at each time-step.

The goal of multi-armed bandit algorithms is to maximize the total reward the player over a horizon. Some applications of the MAB problem include:

1. Advertisement Placement: Choosing which advertisement to display by giving reward if user clicks on the ad.
2. Robotic Grasps: Choosing how to pick object where robot is rewarded if it graps an object.

There is a tradeoff between exploration and exploitation in MAB problems. We need to find a balance between exploiting arms that did well in the past vs exploring arms that might do well in the future.

2 Summary

In this lecture, we deal with stochastic (not adversarial) bandits. This essentially means that each bandit arm has a static reward distribution that does not change. Each arm pull essentially gives us a sample from the distribution. We saw in the previous lecture that given an estimated mean and the number of times an arm is pulled, it is not easy to say whether that arm is the most optimal arm for reward maximisation.

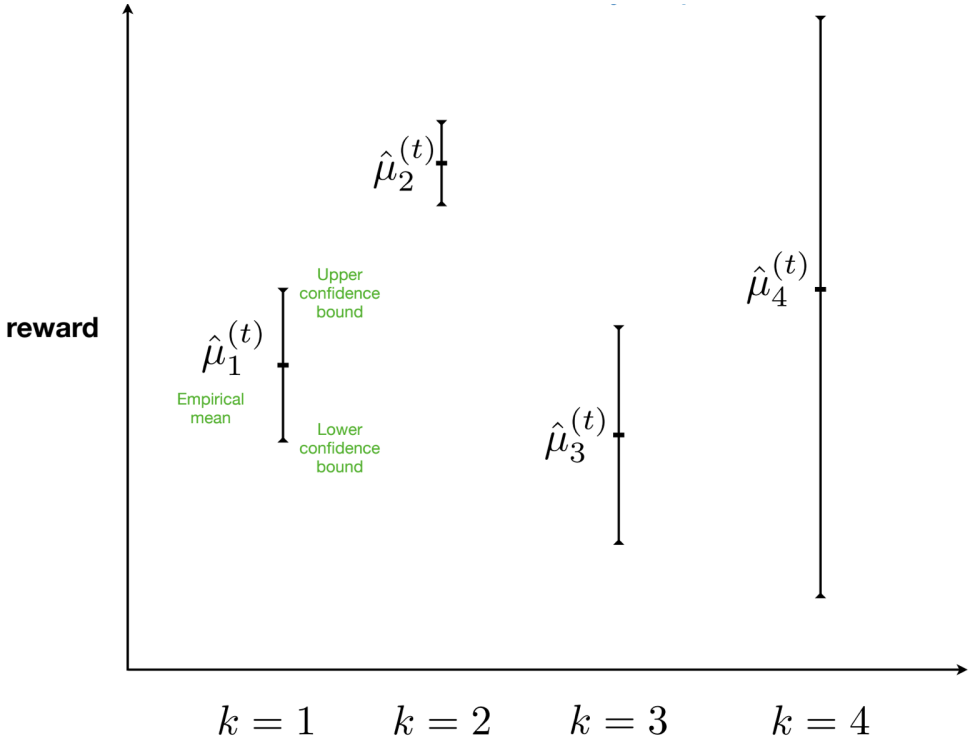

Figure 2: Exploration vs Exploitation

In Fig. 2, a more conservative player would pick arm $k = 2$ as $\hat{\mu}_2$ is the highest among other estimated mean values. Furthermore, the confidence bounds are also tight. On the other hand, a "gambler" would prefer high reward-high risk and choose the arm $k = 4$. The conservative player is analogous to *exploitation* and the gambler is analogous to *exploration*.

2.1 Explore-exploit algorithm

Each Multi-arm bandit aims at reward maximisation by exploring or/and exploiting at each timestep. This algorithm has two different stages.

1. **Explore Phase**: Pull each arm M times to estimate the mean reward.
2. **Exploit Phase**: Keep pulling the arm with the highest expected mean reward until T.

2.2 Multi-Armed Bandit: Explore-Exploit

Algorithm. 2 describes the Explore-Exploit algorithm.

Algorithm 2 Explore-Exploit

for $k = 1 \rightarrow K$ **do**
  for $m = 1 \rightarrow M$ **do**
    $a = k$
    `Receive`(r)
    $\hat{\mu}_k = \hat{\mu}_k + \frac{r}{M}$
  end for
end for
for $t = KM \rightarrow T$ **do**
  $a^{(t)} = \arg\max_{k'} \hat{\mu}_{k'}$
  `Receive`$(r^{(t)})$
end for

Here, K denotes the number of bandits, M is the number of exploration steps per arm, and T denotes the number of timesteps. Note that unlike online algorithms, T here do not go to ∞.

As we can see, this algorithm has two different stages. The first loop corresponds to the exploration stage where each arm is pulled M times to estimate the mean $\hat{\mu}$ iteratively. The second loop employs exploitation. The algorithm picks the arm with the "best" estimated mean for the rest of the timesteps.

2.2.1 Regret Analysis

We will now derive the regret bound of the explore-exploit algorithm. However, before that, we need to discuss **Hoeffding's Inequality** as it'll be used to derive the bounds.

Theorem 2. Hoeffding's Inequality Consider a one-dimension distribution ν with expectation μ, where any sample $r \sim \nu$ is bounded such that $r \in [0, 1]$. Given T i.i.d samples, $\{r^{(t)}\}_{t=1}^{T}$, we have that for any ϵ:

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

From the equation, we see that as we increase T, the probability that the absolute difference between estimated mean and actual mean would be greater than a given threshold decreases. In other words, this inequality means that the estimate of the mean gets better with the more samples are available.

Say, we want the probability to be less than a given specified value δ. In that case, we can then plug in δ into the inequality and solve for ϵ:

$$
\begin{aligned}
\delta &= 2e^{-2T\epsilon^2} \\
log(\delta) &= log(2) - 2T\epsilon^2 \\
2T\epsilon^2 &= log(2) - log(\delta) \\
2T\epsilon^2 &= log(\frac{2}{\delta}) \\
\epsilon^2 &= \frac{log(\frac{2}{\delta})}{2T} \\
\epsilon &= \sqrt{\frac{log(\frac{2}{\delta})}{2T}}
\end{aligned}
$$

This threshold, ϵ, is called "confidence interval" or "confidence bound", as illustrated in Fig. 3.

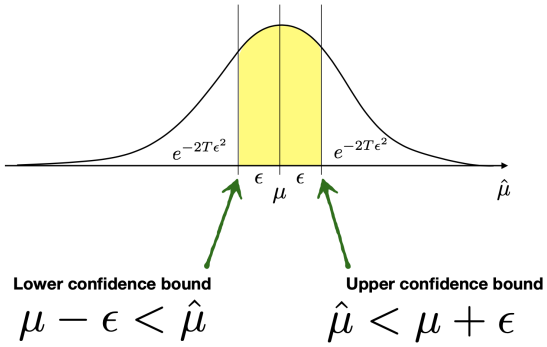

Figure 3: An illustration of confidence interval.

Now, we can derive the regret bound for the Explore-Exploit algorithm. We divide the derivation in two steps: first for the exploration phase and then for the exploitation phase.

Exploration Phase For the exploration phase, the worst case is that the algorithm gets no reward at any timestep. Say, the kth arm is the true best arm. For $M(K-1)$ steps the algorithm does not pick the best arm and therefore, there is a positive regret. In the exploration stage, when the algorithm pulls the kth arm, there could be a scenario where the rewards for those timesteps is not the best since rewards come from a stochastic distribution. Therefore, in worse case scenario, the regret bound for the exploration phase is $\mathcal{O}(KM)$.

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

Exploitation Phase For the exploitation phase, we follow the five step procedure for finding bounds which is:

1. Define the potential function.
2. Upper bound the potential.
3. Lower bound the potential.
4. Combine the bounds.
5. Use algebra to compute the regret bound.

We first define the potential function using the fact that if we pulled the wrong arm (i.e. not the true best arm), that means our estimate of the best arm was wrong. Formally, our potential is defined as

$$\hat{\mu}_{\hat{k}} \geq \hat{\mu}_{k^*}$$

where $\widehat{k} = \arg\max_k \widehat{\mu}_k$ and $k^* = \arg\max_k \mu_k$.

Now, we derive the upper and lower bounds of this potential.

Upper Bound For the upper bound, we can use the Upper Confidence Bound (UCB) (derived using Hoeffding's Inequality) (illustrated in Fig. 3):

$$\begin{aligned}\hat{\mu}_{\hat{k}} &\leq \mu_{\hat{k}} + \epsilon \\ &\leq \mu_{\hat{k}} + \sqrt{\frac{\log 2/\delta}{2M}}\end{aligned}$$

Lower Bound For the lower bound, we can use the Lower Confidence Bound (LCB) (derived using Hoeffding's Inequality) (illustrated in Fig. 3):

$$\begin{aligned}\hat{\mu}_{\hat{k}} &\geq \mu_{k^*} - \epsilon \\ &\geq \mu_{k^*} - \sqrt{\frac{\log 2/\delta}{2M}}\end{aligned}$$

Combining Lower and Upper Bounds We can combine these bounds

$$\begin{aligned}\mu_{\hat{k}} + \sqrt{\frac{\log(2/\delta)}{2M}} &\geq \mu_{k^*} - \sqrt{\frac{\log(2/\delta)}{2M}} \\ \mu_{k^*} - \mu_{\hat{k}} &\leq 2\sqrt{\frac{\log(2/\delta)}{2M}}\end{aligned}$$

The above equation shows the regret bound for one time step. We need to estimate the regret bound for all the timesteps (in the exploitation phase).

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

Combining Exploration and Exploitation Bounds We now have expressions for the regret bounds for the explore and exploit phases. We sum them to find the final expression for the regret of the explore-exploit algorithm.

$$\begin{aligned} R_{\text{exploit}} &= (T - MK)(\mu_{k^*} - \mu_{\widehat{k}}) \\ &\leq (T - MK)2\sqrt{\frac{\log(2/\delta)}{2M}} \\ &\leq \mathcal{O}\left(2(T - MK)\sqrt{\frac{1}{M}}\right). \end{aligned}$$

$$R_{\text{explore}} = \mathcal{O}(KM).$$

$$\begin{aligned} R &= R_{\text{explore}} + R_{\text{exploit}} \\ &\leq \mathcal{O}(KM) + \mathcal{O}\left(2(T - MK)\sqrt{\frac{1}{M}}\right) \\ &\leq \mathcal{O}\left(KM + 2T \cdot \sqrt{\frac{1}{M}}\right) \end{aligned}$$

We see that the regret is linear in T, which would mean that the algorithm does not have a no-regret bound. If we want our algorithm to be no-regret, we want the regret to grow sub-linearly. So, we can choose M in a way to make the regret sub-linear. We take the derivative of RHS and set it to zero.

$$\begin{aligned} 0 &= \frac{d}{dM}\left\{KM + 2T \cdot \sqrt{\frac{1}{M}}\right\} \\ 0 &= K - TM^{-3/2} \\ M &= \left(\frac{T}{K}\right)^{2/3} \end{aligned}$$

Note that, we need to know the total number of pulls T in advance to get the optimal value of M.

Plugging this optimal M into our regret bound expression, we get:

$$\begin{aligned}
R &= R_{\text{explore}} + R_{\text{exploit}} \\
&\leq KM + 2T \cdot \sqrt{\frac{1}{M}} \\
&= K\left(\frac{T}{K}\right)^{2/3} + 2T\sqrt{\left(\frac{T}{K}\right)^{-2/3}} \\
&= K\left(\frac{T}{K}\right)^{2/3} + 2T\left(\frac{T}{K}\right)^{-1/3} \\
&= K^{1/3}T^{2/3} + 2T^{2/3}K^{1/3} \\
&= 3K^{1/3}T^{2/3} \\
&= \mathcal{O}(K^{1/3}T^{2/3})
\end{aligned}$$

Thus, the regret bound for the Explore-Exploit algorithm is $R = \mathcal{O}(K^{1/3}T^{2/3})$. Note that this is a sub-linear regret bound and that the total number of pulls T is assumed to be known.

2.3 Multi-Armed Bandit: Upper Confidence Bound

Even though the explore-exploit algorithm is sublinear in time, it is not the best method and we can obtain lower regret bounds. Now, we discuss another algorithm called the Upper Confidence Bound (UCB) to solve a context-free bandit problem. UCB is an optimistic learner where the arm $k(t)$ with the largest mean reward ($\hat{\mu}_k$) plus a confidence term is picked. The algorithm is detailed in Algorithm. 3.

Algorithm 3 Upper Confidence Bound (UCB)

Input: Time horizon T, Threshold δ'
for $t = 1 \to T$ **do**
  if $t \leq K$ **then**
    $k = t$ ▷ Initially pull each arm once (exploration)
  else
    $k = \arg\max_{k'} \left(\hat{\mu}_{k'} + \sqrt{\frac{\log(2T/\delta')}{2T_{k'}^{(t)}}}\right)$ ▷ mean reward + upper confidence
  end if
  `RECEIVE`$(r^{(t)})$
  $T_k^{(t)} = T_k^{(t')} + 1$ ▷ update pull counter for kth arm
  $\hat{\mu}_k = \frac{1}{T_k^{(t)}}\left(\hat{\mu}_k(T_k^{(t)} - 1) + r^{(t)}\right)$ ▷ update mean reward for k
end for

2.3.1 Understanding confidence term

The confidence term is obtained using Hoeffding's inequality and depends on the number of pulls of a particular arm $T_{k'}^{(t)}$, total pulls T and δ. So, as the game progresses and the number of pulls increase, the learner becomes more confident and the confidence term reduces.

Theorem 3. Boole's inequality (Union Bound) The probability that at least one of the events happens is no greater than the sum of the probabilities of the individual events. Mathematically, $p(\bigcup_i x_i) \leq \sum_i p(x_i)$, where x_i's are events.

Proof Sketch For two events x_1 and x_2,

$$p(x_1 \cup x_2) = p(x_1) + p(x_2) - p(x_1 \cap x_2)$$
$$p(x_1 \cup x_2) \leq p(x_1) + p(x_2)$$

The same logic can be extended to arbitary n events. Using Hoeffding inequality (Theorem 1) and Union bound (Theorem 2), we get the confidence interval as:

$$\mathbb{P}\left(\bigcup_{t=1}^{T} |\widehat{\mu}_k^{(t)} - \mu_k| > \sqrt{\frac{\log(2/\delta)}{2T_k^{(t)}}}\right) \leq \sum_{t=1}^{T} \mathbb{P}\left(|\widehat{\mu}_k^{(t)} - \mu_k| > \sqrt{\frac{\log(2/\delta)}{2T_k^{(t)}}}\right)$$
$$\leq T\delta.$$

Then let $\delta' \triangleq T\delta$ we get

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

We will use the above theorem to derive the regret bound for UCB algorithm:.

2.3.2 Regret Bound

We will use the first five steps to calculate the regret bound starting from the potential function.

The basic inequality that is defined is as follows, the learner may or may not select a non-optimal arm, until the confidence bound of the optimal arm is higher than that of the non-optimal arm. Mathematically, k^* refers to the actual optimal arm, and $k^{(t)}$ refers to the estimated optimal arm at time step t:

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

We see the confidence bound in the inequality as it is also used to estimate the "best" arm at t time step. s

Upper Bound We wish to obtain a upper bound on the predicted arm. From Hoeffding's inequality, we know that

$$|\hat{\mu}^{(t)} - \mu^{(t)}| \leq \epsilon'$$

Using this, we have:

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

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

Lower Bound We wish to obtain a upper bound on the arm from the base inequality that we derived. From Hoeffding's inequality, we know that

$$|\hat{\mu}^{(t)} - \mu| \leq \epsilon'$$

And we know that at time t,

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

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

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

Combining the Upper and Lower Bounds The lower bound is-

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

And the upper bound as we defined is-

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

Combining both the upper and lower bound, we get:

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

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

This being for one instance, if we are pulling an arm k for each time instant t, then we have-

$$\mathcal{R}_{UCB} = \sum_{t=1}^{T} (\mu_{k^*} - \mu_{k^{(t)}})$$

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

$$= 1/2\sqrt{\log(2T/\delta')} \sum_{t=1}^{T} \sqrt{\frac{1}{T_{k^{(t)}}^{(t)}}}$$

We do this process to factor out constants. Next, we change the notation using a sum over arms.

$$= 1/2\sqrt{\log(2T/\delta')}\sum_{t=1}^{T}\sum_{j=1}^{K} 1[k^{(t)} = j]\sqrt{\frac{1}{T_j^{(t)}}}$$

$$= 1/2\sqrt{\log(2T/\delta')}\sum_{j=1}^{K}\sum_{t=1}^{T} 1[k^{(t)} = j]\sqrt{\frac{1}{T_j^{(t)}}}$$

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

There are two mathematical inequalities required to derive the regret bound as discussed in the lecture, namely Jensens inequality and the summation bound-

Theorem 4. Jensen's Inequality Iff f is a convex function, $f(\sum_n P_n x_n) \leq \sum_n P_n f(x_n)$.

Fact 5. *Summation Bound in the lecture slides:*

$$\sum_{t=1}^{T}\sqrt{\frac{1}{t}} \leq 2\sqrt{T}$$

The proof is as follows

$$\begin{aligned}\sum_{t=1}^{T}\sqrt{\frac{1}{t}} &= 1 + \sum_{t=2}^{T}\sqrt{\frac{1}{t}} \\ &\leq 1 + \int_{t=1}^{T}\sqrt{\frac{1}{t}}dt \\ &= 1 + 2\sqrt{(T)} - 2 = 2\sqrt{T} - 1 \\ &\leq 2\sqrt{T}\end{aligned}$$

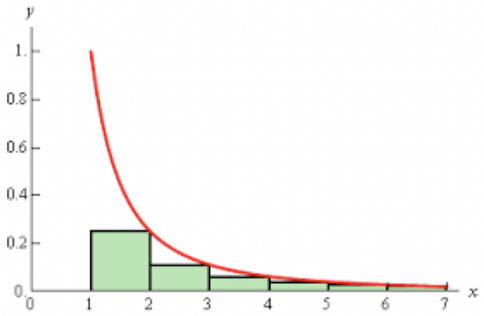

Figure 4: Graphical form of the summation bound

Plugging in our equation

$$\mathcal{R}_{UCB} \leq \sqrt{\log(2T/\delta')} \sum_{j=1}^{K} \sum_{t=1}^{T_j^{(T)}} \sqrt{\frac{1}{t}}$$

$$\leq 1/2\sqrt{\log(2T/\delta')} \sum_{j=1}^{K} 2\sqrt{T_j^{(T)}} \qquad \text{(Using Summation bound)}$$

$$\leq \sqrt{8\log(2T/\delta')} K \left(\frac{1}{K} \sum_{j=1}^{K} \sqrt{T_j^{(T)}} \right)$$

$$\leq \sqrt{\log(2T/\delta')} K \left(\sqrt{\frac{1}{K} \sum_{j=1}^{K} T_j^{(T)}} \right) \qquad \text{(Using Jensen's inequality)}$$

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

$$= \sqrt{\log(2T/\delta')} \sqrt{TK} \approx \mathcal{O}(\sqrt{TK})$$

If we compare, the regret bound for UCB ($\mathcal{O}(\sqrt{T})$ is "more" sub-linear with respect to T than Explore-Exploit algorithm ($\mathcal{O}(T^{\frac{2}{3}})$) and hence, is better. This is perhaps because UCB starts the exploitation at an earlier stage and then continues exploring.

3 Appendix

The proof of the Hoeffdings inequality is as follows- Let $Z_1, Z_2, ...Z_n$ be random bounded variables such that $Z_i \in [0, 1]$ with probability 1. Hence Z is a random variable with $E[Z] = 0$ and $0 \leq Z \leq 1$, then by Chernoff's bound[1], we have

$$E[e^{sZ}] \leq e^{\frac{s^2}{8}}$$

By the convexity of the exponential function, we have

$$e^{sZ} \leq ze^s + (1-z)e^0, 0 \leq z \leq 1$$
$$E[e^{sZ}] \leq E[Z]e^s + E[1-z]e^0$$
$$E[Z] = 0$$
Hence
$$E[e^{sZ}] \leq 1$$

If we say that ϕ is a function of s then, we can write this as

$$E[e^{sZ}] \leq e^{\phi(s)}$$

To minimize the upper bound, let us express $\phi(s)$ as a taylor's series with a remainder-

$$\phi(s) = \phi(0) + s\phi'(0) + \frac{s^2}{2}\phi''(v), v \in [0, s]$$
$$\phi'(s) = 0, \phi''(s) \leq 1/4$$
From this
$$\phi(s) \leq s^2/8$$
$$E[e^{sZ}] \leq e^{s^2/8}$$

Applying this as an upper bound to derive Hoeffdings inequality-

$$P(S_n - E[S_n] \geq t) \leq 2e^{-2T\epsilon^2}$$
In terms of the slide-
$$p\left(\left|\sum_{t=1}^{T} \frac{r^{(t)}}{T} - \mu\right| \geq \epsilon\right) \leq 2e^{-2T\epsilon^2}$$

4 References

[1] https://nowak.ece.wisc.edu/SLT07/lecture7.pdf