Week 6 - Lecture 31

$$\bar{R}(ETC, n) = \underbrace{m\sum_{i=1}^{K}\Delta_i}_{\text{exploration}} + \underbrace{(n-mk)\sum_{i=1}^{k}\Delta_i \exp\left\{\frac{-m\Delta_i^2}{4}\right\}}_{\text{commit}} \quad \text{—— (1)}$$

If m is small then exploration will be less, thus estimate may not be good.
If m is large then the estimate will be good but we would have wasted lot of time exploring non-optimal arms.
Thus need to treat m as a variable and optimize over m.

Thus differentiating & equating with 0, we can find m.
Simplifying eq (1), ignoring mk from $(n-mK)$

$$\bar{R}(ETC, n) \leq m\sum_{i=1}^{k}\Delta_i + \underbrace{n\sum_{i=1}^{k}\Delta_i \exp\left\{\frac{-m\Delta_i^2}{4}\right\}}_{\text{convex}}$$

Differentiating w.r.t. m
and equating with zero

$$\sum_{i=1}^{K}\Delta_i + n\sum_{i=1}^{k}\Delta_i \exp\left\{\frac{-m\Delta_i^2}{4}\right\} \times \frac{(-\Delta_i^2)}{4} = 0$$

Let $K = 2$ $\quad$ [$\Delta_1 = 0$, $\Delta_2 = +ve$ as 1 is optimal]

$$\Delta_2 + n\Delta_2 \exp\left\{\frac{m\Delta_2}{4}\right\} \times \left(\frac{-\Delta_2^2}{4}\right) = 0$$

$$\therefore \quad m = \max\left\{1, \left\lceil \frac{4}{\Delta_2^2} \log\left(\frac{n\Delta_2^2}{4}\right)\right\rceil\right\} \quad \text{——} (1)$$

Since we wan m to be an integer ≥ 1

Putting value of m in eq (1), for $k=2$

$$\bar{R}(ETC, n) \leq \Delta_2 + \frac{4}{\Delta_2}\left(1 + \max\left\{0, \log\left(\frac{n\Delta_2^2}{4}\right)\right\}\right)$$

Simplifying:

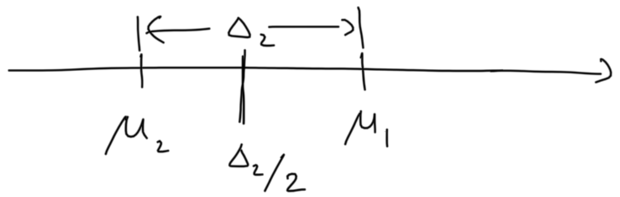
$$\bar{R}(ETC, n) \leq \Delta_2 + \frac{4}{\Delta_2}\left(1 + \log\left(\frac{n\Delta_2^2}{4}\right)\right)$$

Thus regret is $O(\log(n))$

But to set the m as (1) we need to know Δ_2.
It need not know the estimates but the gap between the estimates.

$\Delta_2 = \mu_1 - \mu_2$

Algo is estimating $\hat{\mu}_1$ & $\hat{\mu}_2$



As long as $\hat{\mu}_1$ and $\hat{\mu}_2$ are seperated by Δ_2 and as long as $\hat{\mu}_1 > \Delta_2/2$ and $\hat{\mu}_2 < \Delta_2/2$ then the true optimal arm will be estimated correctly.

But we may not know Δ_2 apriori.

Greedy Algorithm

Sample each arm certain number of times, get estimate and then play greedily henceforth.

If $m = 1$, then we sample each arm once & then select arm greedily henceforth.

ex: arm 1 = Ber (0.5)
arm 2 = Ber (0.8) Possibly,

	1	2	3	
arm 1	1	1	1	...
arm 2	0	–	–	

we will always end up playing arm 1 which is sub-optimal, thus regret will be linear.

Epsilon - Greedy Algorithm

How to choose epsilon and derive regret

bounds to be done by students.