Stochastic Bandits

January 12, 2025

1 Concentration Inequalities

Theorem 1 (Chebyshev's Inequality). *Let X be any random variable. Then for all $t > 0$,*

$$\Pr(|X - E[X]| \geq t) \leq \frac{Var(X)}{t^2}.$$

Theorem 2 (Markov inequality). *Let X be any random variable that takes only non negative values. Then for any $c > 0$,*

$$\Pr(X \geq cE[X]) \leq 1/c.$$

Theorem 3 (Hoffding's inequality). *Suppose $X_1, \ldots, X_n$ be independent bounded random variables such that for all $i \in [n]$, we have $a_i \leq X_i \leq b_i$ for all $i \in [n]$. Let $S_n = X_1 + \cdots + X_n$ and so $E[S_n] = \sum_i E[X_i]$.*

For any $t > 0$, we have

$$\Pr(|S_n - E[S_n]| \geq t) \leq 2e^{-\frac{2t^2}{\sum_i (b_i - a_i)^2}}$$

If $0 \leq X_i \leq 1$ for all $i \in [n]$ then we have,

$$\Pr(|S_n - E[S_n]| \geq t) \leq 2e^{-\frac{2t^2}{n}}$$

Let $\bar{X}_n = S_n/n = \frac{X_1 + \cdots + X_n}{n}$. So $E[\bar{X}_n] = \frac{\sum_i E[X_i]}{n}$ For any $t > 0$, we have

$$\Pr(|\bar{X}_n - E[\bar{X}_n]| \geq t) \leq 2e^{-\frac{2t^2 n^2}{\sum_i (b_i - a_i)^2}}$$

When for all i, $0 \leq X_i \leq 1$ then

$$\Pr(|\bar{X}_n - E[\bar{X}_n]| \geq t) \leq 2e^{-2t^2 n}$$

Remark. *Chebyshev's and Hoffding's inequalities works even for r.v. that can take negative values.*

2 Weak Law of Large Numbers

Let $X_1, \ldots, X_n$ be n iid (independent and identically distributed) random variables. Let $E[X] = \mu$ and $Var(X) = v$. Let $\bar{X}_n = \frac{\sum_i X_i}{n}$ be the empirical average. Note that $E[\bar{X}_n] = \mu$.

When $n \to \infty$ then the empirical average $\bar{X}_n$ converges to the true mean μ. We can see this from Chebyshev's inequality. As $X_1, \ldots, X_n$ are independent we have

$$Var(\bar{X}_n) = Var(\frac{\sum_i X_i}{n}) == \frac{nVar(X)}{n^2} = \frac{Var(X)}{n}.$$

So as $n \to \infty$, the variance of $\bar{X}_n$ tends to 0. So we have

$$\Pr(|\bar{X}_n - \mu| \geq \epsilon) \leq \frac{Var(X)}{n\epsilon^2} = \frac{v}{n\epsilon^2}$$

So for any fixed $\epsilon > 0$, for sufficiently large values of n (this value will depend on ϵ), the probability $\Pr(|\bar{X}_n - \mu| \geq \epsilon)$ will approach 0.

3 Stochastic Bandits

We start with the basic model of bandits, which is independent rewards. An algorithm has K possible arms to choose from, and there are T rounds, both K and T are known in advance (we will see later that one can relax the assumption that T should be known in advance). Each arm a is associated with a reward distribution D_a (pmf/pdf) which is unknown to the algorithm. The mean reward of the distribution D_a will be most relevant for us and is denoted by μ_a, i.e., $\mu_a = \underset{r \sim D_a}{E}[r]$ ($r \sim D_a$ denotes that r is sampled/drawn from the distribution D_a).

In each round $t \in T$:

1. algorithm picks an arm $a_t \in [K]$.
2. reward r_t is sampled from the distribution D_{a_t}.
3. algorithm receives the reward r_t.

The algorithm can be randomized, i.e., in any round it can fix a probability distribution over arms and can pick an arm from this distribution.

Again, it is important to note that $\mu_1, \ldots, \mu_K$ (and distributions $D_1, \ldots, D_K$) are unknown to the algorithm. The algorithm only sees the rewards (of the arms pulled by the algorithm).

We make the following assumptions:

1. rewards are bounded. For simplicty, we will assume rewards at any round will be in $[0, 1]$. So the means $\mu_1, \ldots, \mu_K$ all are in $[0, 1]$.
2. as we are in stochastic setting, we assume all drawn rewards are independent.

An important point to note that, regardless of the algorithm is deterministic or randomized, the arms pulled by the algorithm is a random variable. Because the algorithm's decision to pull any arm in some round will depend on past history of rewards observed by the algorithm. Since the received rewards are itself a random variable, the arms pulled are also random variable. This should become clear when we will see some algorithms.

Notations: Throughout the course, we will stick to the following notations. The set of arms will be denoted by $[K]$ and the total number of rounds will be denoted by T. The best arm is the arm a that has highest mean reward. We will use a^* for the best arm and μ^* for the mean reward of the best arm. That is $a^* = \arg\max_a \mu_a$ and $\mu^* = \mu_{a^*} = \max_a \mu_a$. For any suboptimal arm, we will use $\Delta_a := \mu^* - \mu_a$ for the gap of arm a. Finally, the arm pulled by the algorithm in tth round will be denoted by a_t.

Regret: How do we measure the performance of an algorithm? Recall that the goal of the algorithm is to maximize the sum of rewards received in all rounds.

One standard approach is to compare the algorithm performance with the best possible algorithm that knows the distributions D_a for all $a \in [K]$. Note that if means $\mu_1, \ldots, \mu_K$ are known then best strategy to maximize total expected rewards is to always pick the arm a^* to get total expected reward of $T\mu^*$. It makes sense to define the regret of an algorithm after T rounds as

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

where a_t is the arm pulled by the algorithm in the t th round. Thus $R(T)$ is the regret incurred by the algorithm after T rounds of not knowing the means $\mu_1, \mu_2, \ldots, \mu_K$. Note that the regret $R(T)$ is a random variable. We are interested in the expected regret $E[R(T)]$ of the algorithm.

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

The expectation in the above definition is taken over all the randomness, i.e., randomness over the draw of rewards from the distributions $D_1, \ldots, D_K$ and internal randomness of the algorithm (if the algorithm is randomized).

Remark. *1. By the definition only, $R(T)$ and $E[R(T)]$ of an algorithm depends on the problem-instance, i.e., means $\mu_1, \ldots, \mu_K$. Of course, we want an algorithm whose expected regret is small for all instances. In the previous line, 'for all problem instances' is important. For example, consider a stupid algorithm that always pulls arm 1. This algorithm will have regret 0 if the arm 1 has mean 1 and other arms have mean 0. But of course this algorithm will suffer for other instance such as if $\mu_1 = 0$ and $\mu_2 = 1$*

and will have regret of T. So we want an algorithm that performs well on all instances.

2. *If the regret is small then the algorithm's performance is close to best performance when the distributions are known. So we kind of learn the distributions.*

3. *If $r_1, \ldots, r_T$ are rewards received by the algorithm in rounds $1, 2, \ldots, T$ respectively then one can see that $E[\sum_{i=1}^{T} r_t] = E[\sum_{t=1}^{T} \mu_{a_t}]$. Hence the expected regret of an algorithm is the difference of expected total rewards of best strategy that knows $\mu_1, \ldots, \mu_K$ and the expected total rewards of the algorithm.*

Note: In some text books, $R(T)$ is called random-regret or realized regret and $E[R(T)]$ is called Regret. Its just a matter of convention, what names should we use. Often, from the context it will be clear whether we are talking about $R[T]$ or $E[R(T)]$.

Our goal is to design an algorithm whose expected regret $E[R(T)]$ is as small as possible for all problem-instances. Like a general discussion on algorithms, we will ignore constants and only consider Big-O dependence on T. Also, we assume $T \geq K$ as any reasonable algorithm will pull each arm atleast once. Note that since reward in any round is in $[0, 1]$, $R(T) \leq T$ always. We want an algorithms for which $R(T)$ grows sublinear in T, i.e., $\frac{R(T)}{T} \to 0$ as $T \to \infty$. In other words, average regret per round should be 0 when T is large (so we essentially we have learned the unknown distributions). Smaller the regret's dependence on T, faster the rate of convergence to 0 for the average regret per round.

3.1 Explore Then Commit Algorithm

This algorithm is very intuitive and perhaps we will first come up with this algorithm. If we know the means $\mu_1, \ldots, \mu_K$ then in each round the best algorithm will pick the arm a^* (recall notation, a^* has highest mean $\mu^* = \max_{a \in A}\{\mu_a\}$) to maximize total expected rewards. This algorithm first try each arm m times and finds the estimate $\hat{\mu}_a$ of the mean μ_a for every arm a. Thereafter, the algorithm will always pick an arm that maximizes the empirical mean $\hat{\mu}_a$ (we will call this arm a'). The value of m is set to $\frac{T^{2/3}(\log T)^{1/3}}{K^{2/3}}$ to optimze the expected regret.

We will now prove our first theorem of this course.

Theorem 4. *For any instance (i.e., for any values of unknowns $\mu_1, \ldots, \mu_K$), the expected regret of the algorithm ETC is $O(T^{2/3}(\log T)^{1/3}K^{1/3})$.*

Proof. Let $\epsilon = \sqrt{\frac{5 \ln T}{m}}$. For any arm a, let Bad_a be the event that $|\hat{\mu}_a - \mu_a| \geq \epsilon$. As $\hat{\mu}_a = \frac{\sum_{i=1}^{m} r_{ai}}{m}$ and all rewards are in $[0, 1]$, from Hoffding's inequality, we have

$$\Pr(Bad_a) = \Pr(|\hat{\mu}_a - \mu_a| \geq \epsilon) \leq \frac{2}{e^{2\epsilon^2 m}} = \frac{2}{T^{10}}.$$

Algorithm 1: Explore Then Commit

1. $m = \frac{T^{2/3}(\log T)^{1/3}}{K^{2/3}}$;
2. Try each arm m times. Let for arm a, $r_{a1}, r_{a2}, \ldots, r_{am}$ be the rewards received.;

```
/* Line 2 is exploration/investment phase.  In this phase,
   the algorithm tries each arm so even the worst arms.  But
   we hope that we will be able to find near-optimal arm for
   remaining rounds.                                       */
```

3. For each arm a, set $\hat{\mu}_a$ as the average received reward for the arm a, i.e., we have $\hat{\mu}_a = \frac{\sum_{i \in [m]} r_{ia}}{m}$. Let a' be the arm that maximizes $\hat{\mu}_a$, i.e., $a' = \arg\max_a \{\hat{\mu}_a\}$.;

/* We hope a' is the near-optimal arm, i.e., $\mu^* - \mu_{a'}$ is very small. */

4. From the round $N \cdot K + 1$, always pick the arm a'.

Let $Bad = \cup_a Bad_a$ be the event that for some arm a, $|\hat{\mu}_a - \mu_a| \geq \epsilon$ holds. So $Good = Bad^c$ is the event that, for all arms a, we have $|\hat{\mu}_a - \mu_a| < \epsilon$.

By union bound, we have

$$\Pr(Bad) \leq \sum_a \Pr(Bad_a) \leq K \cdot \frac{2}{T^{10}} \leq \frac{2}{T^9}$$

(as we are assuming $T \geq K$ as any reasonable algorithm will try each arm at least once).

Also,

$$\Pr(Good) = 1 - \Pr(Bad) \geq 1 - \frac{2}{T^9}$$

As the $\Pr(Bad)$ is negligible (we have chosen $\epsilon = \sqrt{\frac{5 \ln T}{m}}$ to ensure that this happens), it will suffice for us to only bound the expeceted regret conditioned on event Good. Following calculation formally show this.

$$\begin{aligned} E[R(T)] &= \Pr(Bad)E[R(T)|Bad] + \Pr(Good)E[R(T)|Good] \\ &\leq \frac{2}{T^9} \cdot T + 1 \cdot E[R(T)|Good] \\ &\leq \frac{2}{T^8} + \cdot E[R(T)|Good] \end{aligned}$$

the second inequality comes from $E[R(T)|Bad] \leq T$ as all rewards are in $[0, 1]$.

Bounding $E[R(T)|Good]$. We will now assume event *Good*, i.e., for all arms a, we have $|\hat{\mu}_a - \mu_a| < \epsilon$. Recall that $R(T) = \sum_{t=1}^{T} \mu^* - \mu_{a_t}$.

The contribution to regret in the investment phase can be at most mK (assuming worst case contribution of 1 in each round). The contribution to the

regret from $(mK+1)$st round till end is $(T-mK-1)(\mu^*-\mu_{a'})$ (recall that a' is the arm that maximizes $\hat{\mu}_a$ and is always chosen from $(mK+1)$st round) . We now claim that $(\mu^*-\mu_{a'})<2\epsilon$. For now let us assume this claim and bound the regret. We will prove the claim later.

$$\begin{aligned} R(T)|Good &\leq mK+(T-mK-1)2\epsilon \\ &\leq mK+2T\epsilon \\ &= mK+2T\sqrt{\frac{5\ln T}{m}} \end{aligned}$$

As m inreases mK increases and $2T\sqrt{\frac{5\ln T}{m}}$ decreases so the above quantity will be maximized when $mK = 2T\sqrt{\frac{5\ln T}{m}}$, i.e., when $m=(2\sqrt{5})^{2/3}\frac{T^{2/3}(\log T)^{1/3}}{K^{2/3}}$. Substituting this value of m we get $R(T)$ (conditioned on *Good*) equal to $O(T^{2/3}K^{1/3}(\log T)^{1/3})$. Obviously, we also have then
$E[R(T)|Good]=O(T^{2/3}K^{1/3}(\log T)^{1/3})$.

From our earlier calculation, $E[R(T)]=\frac{2}{T^8}+\cdot E[R(T)|Good]=O(T^{2/3}K^{1/3}(\log T)^{1/3})$. Now all it remains to prove the claim that $(\mu^*-\mu_{a'})<2\epsilon$. Note that $\hat{\mu}_{a'}\geq\hat{\mu}_{a^*}>\mu^*-\epsilon$ and also $\mu_{a'}>\hat{\mu}_{a'}-\epsilon$. Thus we have $\mu^*-\mu_{a'}<2\epsilon$.

□

Stochastic Bandits

January 16, 2025

1 Concentration Inequalities

Theorem 1 (Chebyshev's Inequality)**.** *Let X be any random variable. Then for all $t > 0$,*

$$\Pr(|X - E[X]| \geq t) \leq \frac{Var(X)}{t^2}.$$

Theorem 2 (Markov inequality)**.** *Let X be any random variable that takes only non negative values. Then for any $c > 0$,*

$$\Pr(X \geq cE[X]) \leq 1/c.$$

Theorem 3 (Hoffding's inequality)**.** *Suppose $X_1, \dots, X_n$ be independent bounded random variables such that for all $i \in [n]$, we have $a_i \leq X_i \leq b_i$ for all $i \in [n]$. Let $S_n = X_1 + \cdots + X_n$ and so $E[S_n] = \sum_i E[X_i]$.*

For any $t > 0$, we have

$$\Pr(|S_n - E[S_n]| \geq t) \leq 2e^{-\frac{2t^2}{\sum_i (b_i - a_i)^2}}$$

If $0 \leq X_i \leq 1$ for all $i \in [n]$ then we have,

$$\Pr(|S_n - E[S_n]| \geq t) \leq 2e^{-\frac{2t^2}{n}}$$

Let $\bar{X}_n = S_n/n = \frac{X_1 + \cdots + X_n}{n}$. So $E[\bar{X}_n] = \frac{\sum_i E[X_i]}{n}$ For any $t > 0$, we have

$$\Pr(|\bar{X}_n - E[\bar{X}_n]| \geq t) \leq 2e^{-\frac{2t^2 n^2}{\sum_i (b_i - a_i)^2}}$$

When for all i, $0 \leq X_i \leq 1$ then

$$\Pr(|\bar{X}_n - E[\bar{X}_n]| \geq t) \leq 2e^{-2t^2 n}$$

Remark. *Chebyshev's and Hoffding's inequalities works even for r.v. that can take negative values.*

2 Weak Law of Large Numbers

Let $X_1, \dots, X_n$ be n iid (independent and identically distributed) random variables. Let $E[X] = \mu$ and $Var(X) = v$. Let $\bar{X}_n = \frac{\sum_i X_i}{n}$ be the empirical average. Note that $E[\bar{X}_n] = \mu$.

When $n \to \infty$ then the empirical average $\bar{X}_n$ converges to the true mean μ. We can see this from Chebyshev's inequality. As $X_1, \dots, X_n$ are independent we have

$$Var(\bar{X}_n) = Var(\frac{\sum_i X_i}{n}) == \frac{nVar(X)}{n^2} = \frac{Var(X)}{n}.$$

So as $n \to \infty$, the variance of $\bar{X}_n$ tends to 0. So we have

$$\Pr(|\bar{X}_n - \mu| \geq \epsilon) \leq \frac{Var(X)}{n\epsilon^2} = \frac{v}{n\epsilon^2}$$

So for any fixed $\epsilon > 0$, for sufficiently large values of n (this value will depend on ϵ), the probability $\Pr(|\bar{X}_n - \mu| \geq \epsilon)$ will approach 0.

3 Stochastic Bandits

We start with the basic model of bandits, which is independent rewards. An algorithm has K possible arms to choose from, and there are T rounds, both K and T are known in advance (we will see later that one can relax the assumption that T should be known in advance). Each arm a is associated with a reward distribution D_a (pmf/pdf) which is unknown to the algorithm. The mean reward of the distribution D_a will be most relevant for us and is denoted by μ_a, i.e., $\mu_a = \underset{r \sim D_a}{E}[r]$ ($r \sim D_a$ denotes that r is sampled/drawn from the distribution D_a).

In each round $t \in T$:

1. algorithm picks an arm $a_t \in [K]$.
2. reward r_t is sampled from the distribution D_{a_t}.
3. algorithm receives the reward r_t.

The algorithm can be randomized, i.e., in any round it can fix a probability distribution over arms and can pick an arm from this distribution.

Again, it is important to note that $\mu_1, \dots, \mu_K$ (and distributions $D_1, \dots, D_K$) are unknown to the algorithm. The algorithm only sees the rewards (of the arms pulled by the algorithm).

We make the following assumptions:

1. rewards are bounded. For simplicty, we will assume rewards at any round will be in $[0, 1]$. So the means $\mu_1, \dots, \mu_K$ all are in $[0, 1]$.
2. as we are in stochastic setting, we assume all drawn rewards are independent.

An important point to note that, regardless of the algorithm is deterministic or randomized, the arms pulled by the algorithm is a random variable. Because the algorithm's decision to pull any arm in some round will depend on past history of rewards observed by the algorithm. Since the received rewards are itself a random variable, the arms pulled are also random variable. This should become clear when we will see some algorithms.

Notations: Throughout the course, we will stick to the following notations. The set of arms will be denoted by $[K]$ and the total number of rounds will be denoted by T. The best arm is the arm a that has highest mean reward. We will use a^* for the best arm and μ^* for the mean reward of the best arm. That is $a^* = \arg\max_a \mu_a$ and $\mu^* = \mu_{a^*} = \max_a \mu_a$. For any suboptimal arm, we will use $\Delta_a := \mu^* - \mu_a$ for the gap of arm a. Finally, the arm pulled by the algorithm in tth round will be denoted by a_t.

Regret: How do we measure the performance of an algorithm? Recall that the goal of the algorithm is to maximize the sum of rewards received in all rounds.

One standard approach is to compare the algorithm performance with the best possible algorithm that knows the distributions D_a for all $a \in [K]$. Note that if means $\mu_1, \ldots, \mu_K$ are known then best strategy to maximize total expected rewards is to always pick the arm a^* to get total expected reward of $T\mu^*$. It makes sense to define the regret of an algorithm after T rounds as

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

where a_t is the arm pulled by the algorithm in the t th round. Thus $R(T)$ is the regret incurred by the algorithm after T rounds of not knowing the means $\mu_1, \mu_2, \ldots, \mu_K$. Note that the regret $R(T)$ is a random variable. We are interested in the expected regret $E[R(T)]$ of the algorithm.

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

The expectation in the above definition is taken over all the randomness, i.e., randomness over the draw of rewards from the distributions $D_1, \ldots, D_K$ and internal randomness of the algorithm (if the algorithm is randomized).

Remark. *1. By the definition only, $R(T)$ and $E[R(T)]$ of an algorithm depends on the problem-instance, i.e., means $\mu_1, \ldots, \mu_K$. Of course, we want an algorithm whose expected regret is small for all instances. In the previous line, 'for all problem instances' is important. For example, consider a stupid algorithm that always pulls arm 1. This algorithm will have regret 0 if the arm 1 has mean 1 and other arms have mean 0. But of course this algorithm will suffer for other instance such as if $\mu_1 = 0$ and $\mu_2 = 1$*

and will have regret of T. So we want an algorithm that performs well on all instances.

2. *If the regret is small then the algorithm's performance is close to best performance when the distributions are known. So we kind of learn the distributions.*

3. *If $r_1, \ldots, r_T$ are rewards received by the algorithm in rounds $1, 2, \ldots, T$ respectively then one can see that $E[\sum_{i=1}^{T} r_t] = E[\sum_{t=1}^{T} \mu_{a_t}]$. Hence the expected regret of an algorithm is the difference of expected total rewards of best strategy that knows $\mu_1, \ldots, \mu_K$ and the expected total rewards of the algorithm.*

Note: In some text books, $R(T)$ is called random-regret or realized regret and $E[R(T)]$ is called Regret. Its just a matter of convention, what names should we use. Often, from the context it will be clear whether we are talking about $R[T]$ or $E[R(T)]$.

Our goal is to design an algorithm whose expected regret $E[R(T)]$ is as small as possible for all problem-instances. Like a general discussion on algorithms, we will ignore constants and only consider Big-O dependence on T. Also, we assume $T \geq K$ as any reasonable algorithm will pull each arm atleast once. Note that since reward in any round is in $[0, 1]$, $R(T) \leq T$ always. We want an algorithms for which $R(T)$ grows sublinear in T, i.e., $\frac{R(T)}{T} \to 0$ as $T \to \infty$. In other words, average regret per round should be 0 when T is large (so we essentially we have learned the unknown distributions). Smaller the regret's dependence on T, faster the rate of convergence to 0 for the average regret per round.

3.1 Explore Then Commit Algorithm

This algorithm is very intuitive and perhaps we will first come up with this algorithm. If we know the means $\mu_1, \ldots, \mu_K$ then in each round the best algorithm will pick the arm a^* (recall notation, a^* has highest mean $\mu^* = \max_{a \in A}\{\mu_a\}$) to maximize total expected rewards. This algorithm first try each arm m times and finds the estimate $\hat{\mu}_a$ of the mean μ_a for every arm a. Thereafter, the algorithm will always pick an arm that maximizes the empirical mean $\hat{\mu}_a$ (we will call this arm a'). The value of m is set to $\frac{T^{2/3}(\log T)^{1/3}}{K^{2/3}}$ to optimze the expected regret.

We will now prove our first theorem of this course.

Theorem 4. *For any instance (i.e., for any values of unknowns $\mu_1, \ldots, \mu_K$), the expected regret of the algorithm ETC is $O(T^{2/3}(\log T)^{1/3}K^{1/3})$.*

Proof. Let $\epsilon = \sqrt{\frac{5 \ln T}{m}}$. For any arm a, let Bad_a be the event that $|\hat{\mu}_a - \mu_a| \geq \epsilon$. As $\hat{\mu}_a = \frac{\sum_{i=1}^{m} r_{ai}}{m}$ and all rewards are in $[0, 1]$, from Hoffding's inequality, we have

$$\Pr(Bad_a) = \Pr(|\hat{\mu}_a - \mu_a| \geq \epsilon) \leq \frac{2}{e^{2\epsilon^2 m}} = \frac{2}{T^{10}}.$$

Algorithm 1: Explore Then Commit

1. $m = \frac{T^{2/3}(\log T)^{1/3}}{K^{2/3}}$;
2. Try each arm m times. Let for arm a, $r_{a1}, r_{a2}, \ldots, r_{am}$ be the rewards received.;

`/* Line 2 is exploration/investment phase. In this phase, the algorithm tries each arm so even the worst arms. But we hope that we will be able to find near-optimal arm for remaining rounds. */`

3. For each arm a, set $\hat{\mu}_a$ as the average received reward for the arm a, i.e., we have $\hat{\mu}_a = \frac{\sum_{i\in[m]} r_{ia}}{m}$. Let a' be the arm that maximizes $\hat{\mu}_a$, i.e., $a' = \arg\max_a \{\hat{\mu}_a\}$.;

`/* We hope` a' `is the near-optimal arm, i.e.,` $\mu^* - \mu_{a'}$ `is very small. */`

4. From the round $N \cdot K + 1$, always pick the arm a'.

Let $Bad = \cup_a Bad_a$ be the event that for some arm a, $|\hat{\mu}_a - \mu_a| \geq \epsilon$ holds. So $Good = Bad^c$ is the event that, for all arms a, we have $|\hat{\mu}_a - \mu_a| < \epsilon$.

By union bound, we have

$$\Pr(Bad) \leq \sum_a \Pr(Bad_a) \leq K \cdot \frac{2}{T^{10}} \leq \frac{2}{T^9}$$

(as we are assuming $T \geq K$ as any reasonable algorithm will try each arm at least once).

Also,

$$\Pr(Good) = 1 - \Pr(Bad) \geq 1 - \frac{2}{T^9}$$

As the $\Pr(Bad)$ is negligible (we have chosen $\epsilon = \sqrt{\frac{5\ln T}{m}}$ to ensure that this happens), it will suffice for us to only bound the expeceted regret conditioned on event Good. Following calculation formally show this.

$$\begin{aligned} E[R(T)] &= \Pr(Bad)E[R(T)|Bad] + \Pr(Good)E[R(T)|Good] \\ &\leq \frac{2}{T^9} \cdot T + 1 \cdot E[R(T)|Good] \\ &\leq \frac{2}{T^8} + \cdot E[R(T)|Good] \end{aligned}$$

the second inequality comes from $E[R(T)|Bad] \leq T$ as all rewards are in $[0,1]$.

Bounding $E[R(T)|Good]$. We will now assume event $Good$, i.e., for all arms a, we have $|\hat{\mu}_a - \mu_a| < \epsilon$. Recall that $R(T) = \sum_{t=1}^{T} \mu^* - \mu_{a_t}$.

The contribution to regret in the investment phase can be at most mK (assuming worst case contribution of 1 in each round). The contribution to the

regret from $(mK+1)$st round till end is $(T-mK-1)(\mu^* - \mu_{a'})$ (recall that a' is the arm that maximizes $\hat{\mu}_a$ and is always chosen from $(mK+1)$st round) . We now claim that $(\mu^* - \mu_{a'}) < 2\epsilon$. For now let us assume this claim and bound the regret. We will prove the claim later.

$$\begin{aligned} R(T)|Good &\leq mK + (T-mK-1)2\epsilon \\ &\leq mK + 2T\epsilon \\ &= mK + 2T\sqrt{\frac{5\ln T}{m}} \end{aligned}$$

As m inreases mK increases and $2T\sqrt{\frac{5\ln T}{m}}$ decreases so the above quantity will be maximized when $mK = 2T\sqrt{\frac{5\ln T}{m}}$, i.e., when $m = (2\sqrt{5})^{2/3}\frac{T^{2/3}(\log T)^{1/3}}{K^{2/3}}$. Substituting this value of m we get $R(T)$ (conditioned on *Good*) equal to $O(T^{2/3}K^{1/3}(\log T)^{1/3})$. Obviously, we also have then
$E[R(T)|Good] = O(T^{2/3}K^{1/3}(\log T)^{1/3})$.

From our earlier calculation, $E[R(T)] = \frac{2}{T^8} + \cdot E[R(T)|Good] = O(T^{2/3}K^{1/3}(\log T)^{1/3})$. Now all it remains to prove the claim that $(\mu^* - \mu_{a'}) < 2\epsilon$. Note that $\hat{\mu}_{a'} \geq \hat{\mu}_{a^*} > \mu^* - \epsilon$ and also $\mu_{a'} > \hat{\mu}_{a'} - \epsilon$. Thus we have $\mu^* - \mu_{a'} < 2\epsilon$.

$\square$

4 Successive Elimination Algorithm

One drawback of ETC algorithm is that it will continue to explore an arm large number of times (m times) even if an arm's reward history might suggest to not pull this arm further. In Successive Elimination algorithm, we discontinue the arm forever once we have belief that the arm is not good. Below is the high level description of this algorithm.

Algorithm 2: Successive Elimination - High Level Description

---

1) Pull every arm once;
2) If there is 'sufficient evidence' that some arm a is not a good arm then remove this arm;

Repeat the above steps over the remaining arms

Now to describe Successive Elimination fully, we just need to specify what 'sufficient evidence' is. For the same, we introduce some notations. For any arm a and round t, let $n_a(t)$ be the number of times the arm a is pulled till the round t. Obviously, we have $\sum_a n_a(t) = t$. Further, let $\hat{\mu}_a(t)$ be the empirical mean of received rewards from the arm a till round t. Formally, let r_{ai} be the reward received from the arm a on the ith pull. Then we have $\hat{\mu}_a(t) = \frac{\sum_{i=1}^{n_a(t)} r_{ai}}{n_a(t)}$. Let $\epsilon_a(t) = \sqrt{\frac{5 \log T}{n_a(t)}}$. Finally, let $UCB_a(t) = \hat{\mu}_a(t) + \epsilon_a(t)$ and $LCB_a(t) = \hat{\mu}_a(t) - \epsilon_a(t)$.

Now we describe the 'sufficent evidence' which Succeesive Elimination employs. Recall the analysis of ETC algorithm. There we define an event *Good* (and show that it holds with high probability) and show that conditioned on *Good*, $\hat{\mu}_a - \epsilon \leq \mu \leq \hat{\mu}_a + \epsilon$ where $\epsilon = \sqrt{\frac{5 \log T}{m}}$. Here also, we will define an event *Good* (and show it will hold with high probability) conditioned on which for all arm a and round t, we will have $LCB_a(t) \leq \mu \leq UCB_a(t)$. Now if at any time t, we have $UCB_a(t) < LCB_{a'}(t)$ for some arms a and a' then we know that $\mu_a < \mu_{a'}$. Hence, it is not a good strategy to pull arm a in any subsequent rounds because we will be better off pulling arm a'. In other words, we can eliminate the arm a for future.

Algorithm 3: Successive Elimination

Activate all the arms;
while $\#$-*rounds* $< T$ **do**
  Pull each active arm once (and receive rewards);
  Deactivate all arms a such that there exits some another arm a' with $LCB_a < UCB_{a'}$;
end

Theorem 5. *For all instances, i.e, for all values of $\mu_1, \ldots, \mu_K$,*

$$E[R(T)] = O(\sqrt{KT \log T})$$

Proof. Let *Good* be the following event: for all arms a and for all rounds t, we have $|\hat{\mu}_a(t) - \mu_a| < \epsilon_a(t)$ (that is, $LCB_a(t) \leq \mu_a \leq UCB_a(t)$). We will prove that

$$\Pr(Good) \geq 1 - O(\frac{1}{T^8})$$

For now, let us assume the above. Like ETC, it will suffice to bound $E[R(T)|Good]$.

$$\begin{aligned}
E[R(T)] &= \Pr(Bad)E[R(T)|Bad] + \Pr(Good)E[R(T)|Good] \\
&\leq O(\frac{1}{T^8}) \cdot T + 1 \cdot E[R(T)|Good] \\
&\leq O(\frac{1}{T^7}) + \cdot E[R(T)|Good]
\end{aligned}$$

the second inequality comes from $E[R(T)|Bad] \leq T$ as all rewards are in $[0, 1]$.

Bounding $E[R(T)|Good]$ *Each calculation in this paragraph is conditioned on Good.* Note that $R(T) = \sum_t (\mu^* - \mu_{a_t}) = \sum_a n_a(T)(\mu^* - \mu_a)$. If we show for each arm a, $(\mu^* - \mu_a) \leq (8\sqrt{\frac{5 \log T}{n_a(T)}})$ then we are done. This is because $R(T) = \sum_a n_a(T)(\mu^* - \mu_a) \leq \sum_a n_a(T)(8\sqrt{\frac{5 \log T}{n_a(T)}}) \leq 8\sqrt{5 n_a(T) \log T}$. As $\sqrt{x}$ is a concave function so we have $\frac{\sum_a \sqrt{n_a(T)}}{K} \leq \sqrt{\frac{\sum_a n_a(T)}{K}} = \sqrt{T/K}$. Thus $R(T) \leq 8\sqrt{5KT \log T}$. So it remains to show that for any arm a, we have $(\mu^* - \mu_a) \leq 8\sqrt{\frac{5 \log T}{n_a(T)}}$. For the sake of analysis, we will refer to the ith iteration of while loop as phase i (for any i). Our first easy observation is that the arm a^* with will never be deactivated. Let t be the last round (corresponding to the end of some phase) where the arm a remained active (in other words, arm a was played exactly once after t). Note that $n_a(t) = n_{a^*}(t)$ (because both arms a and a^* are active till t so both of them are played equal number of times, which is the number of phases completed till t). As the arm a is not deactivated at t, we must have $UCB_a(t) \geq LCB_{a^*}(t)$. Further, we have $\epsilon_a(t) = \epsilon_{a^*}(t) = \sqrt{\frac{5 \log T}{n_a(t)}} = \sqrt{\frac{5 \log T}{n_a(T)-1}}$. It is easy to now see that we have $\mu^* - \mu_a \leq 4\epsilon_a(t) = 4\sqrt{\frac{5 \log T}{n_a(T)-1}} \leq 8\sqrt{\frac{5 \log T}{n_a(T)}}$.

Bounding $\Pr(Good)$ It remains to show :

$$\Pr(Good) \geq 1 - O(\frac{1}{T^8})$$

Let $r_{a1}, r_{a2}, \ldots, r_{aT}$ be T samples from the distribution D_a. Let $Bad_a(t)$ be the event that $|\frac{r_{a1}+\cdots+r_{at}}{t} - \mu_a| \geq \sqrt{\frac{5 \log T}{t}}$. By Hoffding's inequality, we have

$\Pr(Bad_a(t)) \leq O(1/e^{10\log T}) = O(\frac{1}{T^{10}})$. Let Bad_a be the event that for some $1 \leq t \leq T$, we have $|\frac{r_{a1}+\cdots+r_{at}}{t} - \mu_a| \geq \sqrt{\frac{5\log T}{t}}$. By union bound, $\Pr(Bad_a) \leq T \cdot O(\frac{1}{T^{10}}) = O(\frac{1}{T^9})$ Let $Bad = \cup_a Bad_a$. Again by union bound, $\Pr(Bad) \leq K \cdot O(\frac{1}{T^9}) = O(\frac{1}{T^8})$.

□

The regret in the above theorem is worst-case regret, i.e, for any problem-instance (that is, for any values of K, T and $\mu_1, \ldots, \mu_K$, the expected regret $E[R(T)] \leq O(\sqrt{KT\log T})$. Now we will show another type of bounds on expected regret, which will be instance -dependent bound.

Theorem 6. *The expected regret of Successive Elimination satisfies*

$$E[R(T)] \leq O(\log T) \sum_{a:\triangle_a>0} \frac{1}{\triangle_a}$$

where $\triangle_a = \mu^* - \mu_a$ *is the gap of arm* a.

Proof. We define the events *Bad* and *Good* as in the above theorem. Again it will suffice to bound $E[R(T)|Good]$. All calculations now are conditioned on *Good*. We claim that for any suboptimal arm a, we have $n_a(T) \leq 50000\frac{\log T}{\triangle_a^2}$. This implies that $R(T) = \sum_a n_a(T)\triangle_a \leq O(\log T)\sum_{a:\triangle_a>0}\frac{1}{\triangle_a}$. Suppose $n_a(T) > 50000\frac{\log T}{\triangle_a^2}$. Consider the time t when $n_a(t) = 50000\frac{\log T}{\triangle_a^2}$. As the arm a^* is always active we also have $n_{a^*}(t) = 50000\frac{\log T}{\triangle_a^2}$. So now we have $\epsilon_a(t) = \epsilon_{a^*}(t) = \sqrt{\frac{5\log T}{n_a(t)}} = \triangle_a/100$. As we have assumed *Good*, we have $LCB_{a^*}(t) \geq \mu^* - \triangle_a/100$ and $UCB_a(t) \leq \mu_a + \triangle_a/100$. So we have $UCB_a(t) < LCB_{a^*}(t)$ which implies that arm a will be eliminated in round t which contradicts that $n_a(T) > 50000\frac{\log T}{\triangle_a^2}$.

□

Stochastic Bandits

January 19, 2025

1 Concentration Inequalities

Theorem 1 (Chebyshev's Inequality)**.** *Let X be any random variable. Then for all $t > 0$,*

$$\Pr(|X - E[X]| \geq t) \leq \frac{Var(X)}{t^2}.$$

Theorem 2 (Markov inequality)**.** *Let X be any random variable that takes only non negative values. Then for any $c > 0$,*

$$\Pr(X \geq cE[X]) \leq 1/c.$$

Theorem 3 (Hoffding's inequality)**.** *Suppose $X_1, \ldots, X_n$ be independent bounded random variables such that for all $i \in [n]$, we have $a_i \leq X_i \leq b_i$ for all $i \in [n]$. Let $S_n = X_1 + \cdots + X_n$ and so $E[S_n] = \sum_i E[X_i]$.*

For any $t > 0$, we have

$$\Pr(|S_n - E[S_n]| \geq t) \leq 2e^{-\frac{2t^2}{\sum_i (b_i - a_i)^2}}$$

If $0 \leq X_i \leq 1$ for all $i \in [n]$ then we have,

$$\Pr(|S_n - E[S_n]| \geq t) \leq 2e^{-\frac{2t^2}{n}}$$

Let $\bar{X}_n = S_n/n = \frac{X_1 + \cdots + X_n}{n}$. So $E[\bar{X}_n] = \frac{\sum_i E[X_i]}{n}$ For any $t > 0$, we have

$$\Pr(|\bar{X}_n - E[\bar{X}_n]| \geq t) \leq 2e^{-\frac{2t^2 n^2}{\sum_i (b_i - a_i)^2}}$$

When for all i, $0 \leq X_i \leq 1$ then

$$\Pr(|\bar{X}_n - E[\bar{X}_n]| \geq t) \leq 2e^{-2t^2 n}$$

Remark. *Chebyshev's and Hoffding's inequalities works even for r.v. that can take negative values.*

2 Weak Law of Large Numbers

Let $X_1, \ldots, X_n$ be n iid (independent and identically distributed) random variables. Let $E[X] = \mu$ and $Var(X) = v$. Let $\bar{X}_n = \frac{\sum_i X_i}{n}$ be the empirical average. Note that $E[\bar{X}_n] = \mu$.

When $n \to \infty$ then the empirical average $\bar{X}_n$ converges to the true mean μ. We can see this from Chebyshev's inequality. As $X_1, \ldots, X_n$ are independent we have

$$Var(\bar{X}_n) = Var(\frac{\sum_i X_i}{n}) == \frac{nVar(X)}{n^2} = \frac{Var(X)}{n}.$$

So as $n \to \infty$, the variance of $\bar{X}_n$ tends to 0. So we have

$$\Pr(|\bar{X}_n - \mu| \geq \epsilon) \leq \frac{Var(X)}{n\epsilon^2} = \frac{v}{n\epsilon^2}$$

So for any fixed $\epsilon > 0$, for sufficiently large values of n (this value will depend on ϵ), the probability $\Pr(|\bar{X}_n - \mu| \geq \epsilon)$ will approach 0.

3 Stochastic Bandits

We start with the basic model of bandits, which is independent rewards. An algorithm has K possible arms to choose from, and there are T rounds, both K and T are known in advance (we will see later that one can relax the assumption that T should be known in advance). Each arm a is associated with a reward distribution D_a (pmf/pdf) which is unknown to the algorithm. The mean reward of the distribution D_a will be most relevant for us and is denoted by μ_a, i.e., $\mu_a = \underset{r \sim D_a}{E}[r]$ ($r \sim D_a$ denotes that r is sampled/drawn from the distribution D_a).

In each round $t \in T$:

1. algorithm picks an arm $a_t \in [K]$.
2. reward r_t is sampled from the distribution D_{a_t}.
3. algorithm receives the reward r_t.

The algorithm can be randomized, i.e., in any round it can fix a probability distribution over arms and can pick an arm from this distribution.

Again, it is important to note that $\mu_1, \ldots, \mu_K$ (and distributions $D_1, \ldots, D_K$) are unknown to the algorithm. The algorithm only sees the rewards (of the arms pulled by the algorithm).

We make the following assumptions:

1. rewards are bounded. For simplicty, we will assume rewards at any round will be in $[0, 1]$. So the means $\mu_1, \ldots, \mu_K$ all are in $[0, 1]$.
2. as we are in stochastic setting, we assume all drawn rewards are independent.

An important point to note that, regardless of the algorithm is deterministic or randomized, the arms pulled by the algorithm is a random variable. Because the algorithm's decision to pull any arm in some round will depend on past history of rewards observed by the algorithm. Since the received rewards are itself a random variable, the arms pulled are also random variable. This should become clear when we will see some algorithms.

Notations: Throughout the course, we will stick to the following notations. The set of arms will be denoted by $[K]$ and the total number of rounds will be denoted by T. The best arm is the arm a that has highest mean reward. We will use a^* for the best arm and μ^* for the mean reward of the best arm. That is $a^* = \arg\max_a \mu_a$ and $\mu^* = \mu_{a^*} = \max_a \mu_a$. For any suboptimal arm, we will use $\Delta_a := \mu^* - \mu_a$ for the gap of arm a. Finally, the arm pulled by the algorithm in tth round will be denoted by a_t.

Regret: How do we measure the performance of an algorithm? Recall that the goal of the algorithm is to maximize the sum of rewards received in all rounds.

One standard approach is to compare the algorithm performance with the best possible algorithm that knows the distributions D_a for all $a \in [K]$. Note that if means $\mu_1, \ldots, \mu_K$ are known then best strategy to maximize total expected rewards is to always pick the arm a^* to get total expected reward of $T\mu^*$. It makes sense to define the regret of an algorithm after T rounds as

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

where a_t is the arm pulled by the algorithm in the t th round. Thus $R(T)$ is the regret incurred by the algorithm after T rounds of not knowing the means $\mu_1, \mu_2, \ldots, \mu_K$. Note that the regret $R(T)$ is a random variable. We are interested in the expected regret $E[R(T)]$ of the algorithm.

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

The expectation in the above definition is taken over all the randomness, i.e., randomness over the draw of rewards from the distributions $D_1, \ldots, D_K$ and internal randomness of the algorithm (if the algorithm is randomized).

Remark. *1. By the definition only, $R(T)$ and $E[R(T)]$ of an algorithm depends on the problem-instance, i.e., means $\mu_1, \ldots, \mu_K$. Of course, we want an algorithm whose expected regret is small for all instances. In the previous line, 'for all problem instances' is important. For example, consider a stupid algorithm that always pulls arm 1. This algorithm will have regret 0 if the arm 1 has mean 1 and other arms have mean 0. But of course this algorithm will suffer for other instance such as if $\mu_1 = 0$ and $\mu_2 = 1$*

and will have regret of T. So we want an algorithm that performs well on all instances.

2. *If the regret is small then the algorithm's performance is close to best performance when the distributions are known. So we kind of learn the distributions.*

3. *If $r_1, \ldots, r_T$ are rewards received by the algorithm in rounds $1, 2, \ldots, T$ respectively then one can see that $E[\sum_{i=1}^{T} r_t] = E[\sum_{t=1}^{T} \mu_{a_t}]$. Hence the expected regret of an algorithm is the difference of expected total rewards of best strategy that knows $\mu_1, \ldots, \mu_K$ and the expected total rewards of the algorithm.*

Note: In some text books, $R(T)$ is called random-regret or realized regret and $E[R(T)]$ is called Regret. Its just a matter of convention, what names should we use. Often, from the context it will be clear whether we are talking about $R[T]$ or $E[R(T)]$.

Our goal is to design an algorithm whose expected regret $E[R(T)]$ is as small as possible for all problem-instances. Like a general discussion on algorithms, we will ignore constants and only consider Big-O dependence on T. Also, we assume $T \geq K$ as any reasonable algorithm will pull each arm atleast once. Note that since reward in any round is in $[0, 1]$, $R(T) \leq T$ always. We want an algorithms for which $R(T)$ grows sublinear in T, i.e., $\frac{R(T)}{T} \to 0$ as $T \to \infty$. In other words, average regret per round should be 0 when T is large (so we essentially we have learned the unknown distributions). Smaller the regret's dependence on T, faster the rate of convergence to 0 for the average regret per round.

3.1 Explore Then Commit Algorithm

This algorithm is very intuitive and perhaps we will first come up with this algorithm. If we know the means $\mu_1, \ldots, \mu_K$ then in each round the best algorithm will pick the arm a^* (recall notation, a^* has highest mean $\mu^* = \max_{a \in A}\{\mu_a\}$) to maximize total expected rewards. This algorithm first try each arm m times and finds the estimate $\hat{\mu}_a$ of the mean μ_a for every arm a. Thereafter, the algorithm will always pick an arm that maximizes the empirical mean $\hat{\mu}_a$ (we will call this arm a'). The value of m is set to $\frac{T^{2/3}(\log T)^{1/3}}{K^{2/3}}$ to optimze the expected regret.

We will now prove our first theorem of this course.

Theorem 4. *For any instance (i.e., for any values of unknowns $\mu_1, \ldots, \mu_K$), the expected regret of the algorithm ETC is $O(T^{2/3}(\log T)^{1/3}K^{1/3})$.*

Proof. Let $\epsilon = \sqrt{\frac{5 \ln T}{m}}$. For any arm a, let Bad_a be the event that $|\hat{\mu}_a - \mu_a| \geq \epsilon$. As $\hat{\mu}_a = \frac{\sum_{i=1}^{m} r_{ai}}{m}$ and all rewards are in $[0, 1]$, from Hoffding's inequality, we have

$$\Pr(Bad_a) = \Pr(|\hat{\mu}_a - \mu_a| \geq \epsilon) \leq \frac{2}{e^{2\epsilon^2 m}} = \frac{2}{T^{10}}.$$

Algorithm 1: Explore Then Commit

1. $m = \frac{T^{2/3}(\log T)^{1/3}}{K^{2/3}}$;
2. Try each arm m times. Let for arm a, $r_{a1}, r_{a2}, \ldots, r_{am}$ be the rewards received.;

```
/* Line 2 is exploration/investment phase.  In this phase,
   the algorithm tries each arm so even the worst arms.  But
   we hope that we will be able to find near-optimal arm for
   remaining rounds.                                        */
```

3. For each arm a, set $\hat{\mu}_a$ as the average received reward for the arm a, i.e., we have $\hat{\mu}_a = \frac{\sum_{i \in [m]} r_{ia}}{m}$. Let a' be the arm that maximizes $\hat{\mu}_a$, i.e., $a' = \arg\max_a \{\hat{\mu}_a\}$.;

```
/* We hope a' is the near-optimal arm, i.e., μ* − μ_a' is very
   small.                                                   */
```

4. From the round $N \cdot K + 1$, always pick the arm a'.

Let $Bad = \cup_a Bad_a$ be the event that for some arm a, $|\hat{\mu}_a - \mu_a| \geq \epsilon$ holds. So $Good = Bad^c$ is the event that, for all arms a, we have $|\hat{\mu}_a - \mu_a| < \epsilon$.

By union bound, we have

$$\Pr(Bad) \leq \sum_a \Pr(Bad_a) \leq K \cdot \frac{2}{T^{10}} \leq \frac{2}{T^9}$$

(as we are assuming $T \geq K$ as any reasonable algorithm will try each arm at least once).

Also,

$$\Pr(Good) = 1 - \Pr(Bad) \geq 1 - \frac{2}{T^9}$$

As the $\Pr(Bad)$ is negligible (we have chosen $\epsilon = \sqrt{\frac{5 \ln T}{m}}$ to ensure that this happens), it will suffice for us to only bound the expeceted regret conditioned on event Good. Following calculation formally show this.

$$\begin{aligned}
E[R(T)] &= \Pr(Bad)E[R(T)|Bad] + \Pr(Good)E[R(T)|Good] \\
&\leq \frac{2}{T^9} \cdot T + 1 \cdot E[R(T)|Good] \\
&\leq \frac{2}{T^8} + \cdot E[R(T)|Good]
\end{aligned}$$

the second inequality comes from $E[R(T)|Bad] \leq T$ as all rewards are in $[0, 1]$.

Bounding $E[R(T)|Good]$**.** We will now assume event $Good$, i.e., for all arms a, we have $|\hat{\mu}_a - \mu_a| < \epsilon$. Recall that $R(T) = \sum_{t=1}^{T} \mu^* - \mu_{a_t}$.

The contribution to regret in the investment phase can be at most mK (assuming worst case contribution of 1 in each round). The contribution to the

regret from $(mK+1)$st round till end is $(T-mK-1)(\mu^*-\mu_{a'})$ (recall that a' is the arm that maximizes $\hat{\mu}_a$ and is always chosen from $(mK+1)$st round) . We now claim that $(\mu^*-\mu_{a'})<2\epsilon$. For now let us assume this claim and bound the regret. We will prove the claim later.

$$\begin{aligned}
R(T)|Good &\leq mK+(T-mK-1)2\epsilon \\
&\leq mK+2T\epsilon \\
&= mK+2T\sqrt{\frac{5\ln T}{m}}
\end{aligned}$$

As m inreases mK increases and $2T\sqrt{\frac{5\ln T}{m}}$ decreases so the above quantity will be maximized when $mK=2T\sqrt{\frac{5\ln T}{m}}$, i.e., when $m=(2\sqrt{5})^{2/3}\frac{T^{2/3}(\log T)^{1/3}}{K^{2/3}}$. Substituting this value of m we get $R(T)$ (conditioned on *Good*) equal to $O(T^{2/3}K^{1/3}(\log T)^{1/3})$. Obviously, we also have then
$E[R(T)|Good]=O(T^{2/3}K^{1/3}(\log T)^{1/3})$.

From our earlier calculation, $E[R(T)]=\frac{2}{T^8}+\cdot E[R(T)|Good]=O(T^{2/3}K^{1/3}(\log T)^{1/3})$. Now all it remains to prove the claim that $(\mu^*-\mu_{a'})<2\epsilon$. Note that $\hat{\mu}_{a'}\geq\hat{\mu}_{a^*}>\mu^*-\epsilon$ and also $\mu_{a'}>\hat{\mu}_{a'}-\epsilon$. Thus we have $\mu^*-\mu_{a'}<2\epsilon$.

□

4 Successive Elimination Algorithm

One drawback of ETC algorithm is that it will continue to explore an arm large number of times (m times) even if an arm's reward history might suggest to not pull this arm further. In Successive Elimination algorithm, we discontinue the arm forever once we have belief that the arm is not good. Below is the high level description of this algorithm.

Algorithm 2: Successive Elimination - High Level Description

---

1) Pull every arm once;
2) If there is 'sufficient evidence' that some arm a is not a good arm then remove this arm;
Repeat the above steps over the remaining arms

Now to describe Successive Elimination fully, we just need to specify what 'sufficient evidence' is. For the same, we introduce some notations. For any arm a and round t, let $n_a(t)$ be the number of times the arm a is pulled till the round t. Obviously, we have $\sum_a n_a(t) = t$. Further, let $\hat{\mu}_a(t)$ be the empirical mean of received rewards from the arm a till round t. Formally, let r_{ai} be the reward received from the arm a on the ith pull. Then we have $\hat{\mu}_a(t) = \frac{\sum_{i=1}^{n_a(t)} r_{ai}}{n_a(t)}$. Let $\epsilon_a(t) = \sqrt{\frac{5 \log T}{n_a(t)}}$. Finally, let $UCB_a(t) = \hat{\mu}_a(t) + \epsilon_a(t)$ and $LCB_a(t) = \hat{\mu}_a(t) - \epsilon_a(t)$.

Now we describe the 'sufficent evidence' which Succeesive Elimination employs. Recall the analysis of ETC algorithm. There we define an event *Good* (and show that it holds with high probability) and show that conditioned on *Good*, $\hat{\mu}_a - \epsilon \leq \mu \leq \hat{\mu}_a + \epsilon$ where $\epsilon = \sqrt{\frac{5 \log T}{m}}$. Here also, we will define an event *Good* (and show it will hold with high probability) conditioned on which for all arm a and round t, we will have $LCB_a(t) \leq \mu \leq UCB_a(t)$. Now if at any time t, we have $UCB_a(t) < LCB_{a'}(t)$ for some arms a and a' then we know that $\mu_a < \mu_{a'}$. Hence, it is not a good strategy to pull arm a in any subsequent rounds because we will be better off pulling arm a'. In other words, we can eliminate the arm a for future.

Algorithm 3: Successive Elimination

Activate all the arms;
while $\#$-*rounds* $< T$ **do**
  Pull each active arm once (and receive rewards);
  Deactivate all arms a such that there exits some another arm a' with $LCB_a < UCB_{a'}$;
end

Theorem 5. *For all instances, i.e, for all values of $\mu_1, \dots, \mu_K$,*

$$E[R(T)] = O(\sqrt{KT\log T})$$

Proof. Let *Good* be the following event: for all arms a and for all rounds t, we have $|\hat{\mu}_a(t) - \mu_a| < \epsilon_a(t)$ (that is, $LCB_a(t) \le \mu_a \le UCB_a(t)$). We will prove that

$$\Pr(Good) \ge 1 - O(\frac{1}{T^8})$$

For now, let us assume the above. Like ETC, it will suffice to bound $E[R(T)|Good]$.

$$\begin{aligned} E[R(T)] &= \Pr(Bad)E[R(T)|Bad] + \Pr(Good)E[R(T)|Good] \\ &\le O(\frac{1}{T^8}) \cdot T + 1 \cdot E[R(T)|Good] \\ &\le O(\frac{1}{T^7}) + \cdot E[R(T)|Good] \end{aligned}$$

the second inequality comes from $E[R(T)|Bad] \le T$ as all rewards are in $[0, 1]$.

Bounding $E[R(T)|Good]$ *Each calculation in this paragraph is conditioned on Good.* Note that $R(T) = \sum_t(\mu^* - \mu_{a_t}) = \sum_a n_a(T)(\mu^* - \mu_a)$. If we show for each arm a, $(\mu^* - \mu_a) \le (8\sqrt{\frac{5\log T}{n_a(T)}})$ then we are done. This is because $R(T) = \sum_a n_a(T)(\mu^* - \mu_a) \le \sum_a n_a(T)(8\sqrt{\frac{5\log T}{n_a(T)}}) \le 8\sqrt{5n_a(T)\log T}$. As $\sqrt{x}$ is a concave function so we have $\frac{\sum_a \sqrt{n_a(T)}}{K} \le \sqrt{\frac{\sum_a n_a(T)}{K}} = \sqrt{T/K}$. Thus $R(T) \le 8\sqrt{5KT\log T}$. So it remains to show that for any arm a, we have $(\mu^* - \mu_a) \le 8\sqrt{\frac{5\log T}{n_a(T)}}$. For the sake of analysis, we will refer to the ith iteration of while loop as phase i (for any i). Our first easy observation is that the arm a^* with will never be deactivated. Let t be the last round (corresponding to the end of some phase) where the arm a remained active (in other words, arm a was played exactly once after t). Note that $n_a(t) = n_{a^*}(t)$ (because both arms a and a^* are active till t so both of them are played equal number of times, which is the number of phases completed till t). As the arm a is not deactivated at t, we must have $UCB_a(t) \ge LCB_{a^*}(t)$. Further, we have $\epsilon_a(t) = \epsilon_{a^*}(t) = \sqrt{\frac{5\log T}{n_a(t)}} = \sqrt{\frac{5\log T}{n_a(T)-1}}$. It is easy to now see that we have $\mu^* - \mu_a \le 4\epsilon_a(t) = 4\sqrt{\frac{5\log T}{n_a(T)-1}} \le 8\sqrt{\frac{5\log T}{n_a(T)}}$.

Bounding $\Pr(Good)$ It remains to show :

$$\Pr(Good) \ge 1 - O(\frac{1}{T^8})$$

Let $r_{a1}, r_{a2}, \dots, r_{aT}$ be T samples from the distribution D_a. Let $Bad_a(t)$ be the event that $|\frac{r_{a1}+\cdots+r_{at}}{t} - \mu_a| \ge \sqrt{\frac{5\log T}{t}}$. By Hoffding's inequality, we have

$\Pr(Bad_a(t)) \leq O(1/e^{10\log T}) = O(\frac{1}{T^{10}})$. Let Bad_a be the event that for some $1 \leq t \leq T$, we have $|\frac{r_{a1}+\cdots+r_{at}}{t} - \mu_a| \geq \sqrt{\frac{5\log T}{t}}$. By union bound, $\Pr(Bad_a) \leq T \cdot O(\frac{1}{T^{10}}) = O(\frac{1}{T^9})$ Let $Bad = \cup_a Bad_a$. Again by union bound, $\Pr(Bad) \leq K \cdot O(\frac{1}{T^9}) = O(\frac{1}{T^8})$.

□

The regret in the above theorem is worst-case regret, i.e, for any problem-instance (that is, for any values of K, T and $\mu_1, \ldots, \mu_K$, the expected regret $E[R(T)] \leq O(\sqrt{KT\log T})$. Now we will show another type of bounds on expected regret, which will be instance -dependent bound.

Theorem 6. *The expected regret of Successive Elimination satisfies*

$$E[R(T)] \leq O(\log T) \sum_{a:\triangle_a>0} \frac{1}{\triangle_a}$$

where $\triangle_a = \mu^* - \mu_a$ is the gap of arm a.

Proof. We define the events Bad and $Good$ as in the above theorem. Again it will suffice to bound $E[R(T)|Good]$. All calculations now are conditioned on $Good$. We claim that for any suboptimal arm a, we have $n_a(T) \leq 50000\frac{\log T}{\triangle_a^2}$. This implies that $R(T) = \sum_a n_a(T)\triangle_a \leq O(\log T)\sum_{a:\triangle_a>0}\frac{1}{\triangle_a}$. Suppose $n_a(T) > 50000\frac{\log T}{\triangle_a^2}$. Consider the time t when $n_a(t) = 50000\frac{\log T}{\triangle_a^2}$. As the arm a^* is always active we also have $n_{a^*}(t) = 50000\frac{\log T}{\triangle_a^2}$. So now we have $\epsilon_a(t) = \epsilon_{a^*}(t) = \sqrt{\frac{5\log T}{n_a(t)}} = \triangle_a/100$. As we have assumed $Good$, we have $LCB_{a^*}(t) \geq \mu^* - \triangle_a/100$ and $UCB_a(t) \leq \mu_a + \triangle_a/100$. So we have $UCB_a(t) < LCB_{a^*}(t)$ which implies that arm a will be eliminated in round t which contradicts that $n_a(T) > 50000\frac{\log T}{\triangle_a^2}$.

□

UCB Algorithm

Let $n_a(t), \epsilon_a(t), \hat{\mu}_a, UCB_a(t)$ be as defined before (in Successive Elimination algorithm).

The idea of UCB is to add bonus to the empirical mean and then pick an arm that has highest value of empirical mean plus bonus. The bonus is chosen to be $\epsilon_a(t)$ which is equal to $\sqrt{\frac{5\log T}{n_a(t)}}$ (since $\hat{\mu}_a(t) + \epsilon_a(t) = UCB_a(t)$, the algorithm, at any time t, pulls an arm that has highest value of $UCB_a(t)$).

Let us now go into the intuition behind the UCB algorithm in detail. During the initial rounds, the difference between the empirical mean and the actual mean of an arm can be significant. Consequently, selecting an arm solely based on the maximum empirical mean value is not a good strategy. To address this issue, the algorithm incorporates a bonus term. If an arm a is underexplored, the bonus is large, which encourages the algorithm to explore that arm (even if its empirical mean is small at this time). One concern might be whether the bonus term could lead the algorithm to pull bad arms excessively. However, this scenario will not happen because the bonus term diminishes as the number of pulls for the arm increases.

Algorithm 4: UCB Algorithm

$UCB_a = \infty$ for all arms a;
`/* Initialization */`
while *#-rounds* $< T$ **do**
  Pull an arm that has the highest value of UCB_a;
  `/* Recall that` $UCB_a(t) = \hat{\mu}_a(t) + \epsilon_a(t)$ `*/`
end

UCB acheives the same guarantee on expected regret as that of Successive Elimination. The proof is almost same so here we do not give a complete proof.

Theorem 7. *For all instances, i.e, for all values of* $\mu_1, \ldots, \mu_K$*,*

$$E[R(T)] = O(\sqrt{KT\log T})$$

and

$$E[R(T)] = O(\log T) \sum_{a:\triangle_a > 0} \frac{1}{\triangle_a}$$

where $\triangle_a = \mu^* - \mu_a$ *is the gap of arm* a*.*

Proof. The proof is almost same as that of Successive Elimination. To prove $E[R(T)] = O(\sqrt{KT\log T})$, we now show that for any arm a, $\mu^* - \mu_a \leq 2\sqrt{\frac{5\log T}{n_a(T)}}$ (assuming *Good*) (and then the rest of the proof is exactly same). Let t_a be the last round when the arm a was pulled. Thus $n_a(T) = n_a(t)$. Note $\mu_a \geq UCB_a(t) - 2\epsilon_a(t)$ (as we are assuming *good*), $UCB_{a^*} \geq \mu^*$ (as we are assuming

Good) and $UCB_{a^*}(t_a) \leq UCB_a(t_a)$ (as arm a was pulled in round t_a). Hence, $\mu^* - \mu_a \leq 2\sqrt{\frac{5\log T}{n_a(t_a)}} = 2\sqrt{\frac{5\log T}{n_a(T)}}$. Now the proof goes the same was as in Successive Elimination algorithm.

The proof of instance dependent bound is also similar. We here prove that (assuming *Good*) for any suboptimal arm a, we have $n_a(T) \leq 50000\frac{\log T}{\triangle_a^2}$ (and then the proof is exactly same). Suppose not. Consider a time t when $n_a(t) = 50000\frac{\log T}{\triangle_a^2}$. We have $\epsilon_a(t) = \triangle_a/100$. Note that for any time $t' > t$, we have $\epsilon_a(t') \leq \epsilon_a(t)$. For any time $t' > t$, we have $UCB_a(t') \leq \mu_a + 2\epsilon_a(t') \leq \mu_a + 2\epsilon_a(t) < \mu^* \leq UCB_{a^*}(t')$. This means that for any $t' > t$ we will have $UCB_{a^*}(t') > UCB_a(t')$ which means that arm a will never be pulled after t. This contradicts that $n_a(t) > 50000\frac{\log T}{\triangle_a^2}$.

$\square$

MOSS algorithm (UCB2)

MOSS is a variant of UCB and it has the expected regret of $O(\sqrt{KT})$ and hence it beats both Successive Elimination and UCB in theory. In the next lecture, we will see that no algorithm can have smaller expected regret than $O(\sqrt{KT})$ and hence MOSS is the best possible algorithm.

MOSS is same as UCB expect how bonus is calculated. Now the bonus is set to $\sqrt{\frac{\max(\log \frac{T}{Kn_a(t)},0)}{n_a(t)}}$. Let

$$I_a(t) = \hat{\mu}_a(t) + \sqrt{\frac{\max(\log \frac{T}{Kn_a(t)}, 0)}{n_a(t)}}.$$

At any time t, MOSS pulls an arm that has a maximum value of I_a. One reason MOSS has better expected regret bound than UCB is that the estimation precision when $n_a(t)$ is large is more accurate in MOSS than UCB. In other words, the bonus more quickly goes to 0 in MOSS than UCB as $n_a(t)$ increases.

Algorithm 5: MOSS Algorithm

$I_a = \infty$ for all arms a;
`/* Initialization */`
while $\#$-*rounds* $< T$ **do**
  Pull an arm that has the highest value of I_a;
end

Theorem 8. *The expected regret $E[R(T)]$ of MOSS satisfies*

$$E[R(T)] = O(\sqrt{KT})$$

Proof. We will use a trick that is frequently employed in analysis of randomized algorithms. Instead of sampling from the distribution D_a at the time when the arm a is pulled, we assume that T independent samples from every distribution D_a has already been sampled before the start of the algorithm. For each arm a, let $r_{a1}, \ldots, r_{aT}$ be the T independent samples drawn from the distribution D_a. Now when the algorithm pulls arm a, we provide sample (to the algorithm) from $r_{a1}, \ldots, r_{aT}$. In particular, the sample provided to the algorithm for ith pull of arm a is r_{ai}.

Let us define new notations. Let for any $1 \leq x \leq T$, $\hat{\mu}_{a\,x} = \frac{\sum_{j=1}^{x} r_{aj}}{x}$ be the average of first x rewards (of T samples drawn beforehand).With respect to this new notation, note that $\hat{\mu}_a(t) = \hat{\mu}_{a\,n_a(t)}$. Further, for any $1 \leq x \leq T$, let $I_{a\,x} = \hat{\mu}_{a\,x} + \sqrt{\frac{\max(\log \frac{T}{Kx},0)}{x}}$. Again note that $I_a(t) = I_{a\,n_a(t)}$. We will call $I_a(t)$ as the index of arm a after time t.

Let $\delta = \max\{\mu^* - \min_{1 \leq x \leq T} I_{a^*\,x}, 0\}$. Note that δ is a random variable. By definition only, the index of the best arm will never be less than $\mu^* - \delta$, i.e.,

$I_{a^*}(t) \geq \mu^* - \delta$ for all t. We will prove later that $E[\delta] \leq 10\sqrt{\frac{K}{T}}$. It will be helpful to keep this fact in mind.

Let us call an arm a as *Good* if $\triangle_a \leq 5\sqrt{\frac{K}{T}}$ (note that this is different - in previous algorithms, *Good* and *Bad* were events). An arm a is called *Bad* if $\triangle_a > 5\sqrt{\frac{K}{T}}$. Now it will be clear why we have defined *Good* and *Bad* arms in this way. Recall that $R(T) = \sum_a R_a(T)$ where $R_a(T) = n_a(T)\triangle_a$.

$$\begin{aligned}
R(T) &= \sum_{a:a \text{ is } Good} R_a(T) + \sum_{a:a \text{ is } Bad} R_a(T) \\
&\leq \sum_{a:a \text{ is } Good} n_a(T) 5\sqrt{\frac{K}{T}} + \sum_{a:a \text{ is } Bad} R_a(T) \\
&\leq 5\sqrt{\frac{K}{T}} \sum_{a:a \text{ is } Good} n_a(T) + \sum_{a:a \text{ is } Bad} R_a(T) \\
&\leq 5\sqrt{\frac{K}{T}} \cdot T + \sum_{a:a \text{ is } Bad} R_a(T) \\
&\leq 5\sqrt{KT} + \sum_{a:a \text{ is } Bad} R_a(T)
\end{aligned}$$

□

Thus it suffices to show $\sum_{a:a \text{ is } Bad} R_a(T) = O(\sqrt{KT})$. Let us introduce few more notations. For any bad arm a, we define a value k_a (which is a random variable) as follows:

$$k_a = |\{1 \leq x \leq T | I_{a\,x} > \mu_a + \frac{\triangle_a}{2}\}|$$

We also define J which is is a random subset of bad arms defined as below:

$$J = \{a \in [K] | a \text{ is } Bad \text{ and } \triangle_a > 2\delta\}$$

A very important observation is that for any arm in J, we have $n_a(T) \leq k_a$. This is the main crux of the analysis. As directly showing bounds for $E[n_a(t)]$ is difficult but later we will be able to show bounds for $E[k_a]$ for bad arms.

Now

$$\begin{aligned}
\sum_{a:a \text{ is } Bad} R_a(T) &= \sum_{a \in J} n_a(T)\triangle_a + \sum_{a \notin J: a \text{ is } Bad} n_a(T)\triangle_a \\
&\leq \sum_{a:a \text{ is } Bad} k_a\triangle_a + 2\delta T
\end{aligned}$$

Now

$$\begin{aligned}
E[\sum_{a:a \text{ is } Bad} R_a(T)] &\leq \sum_{a:a \text{ is } Bad} E[k_a]\triangle_a + 2E[\delta]T \\
&\leq \sum_{a:a \text{ is } Bad} E[k_a]\triangle_a + 20\sqrt{KT}
\end{aligned}$$

as we earlier claimed (without proof) that $E[\delta] \le 10\sqrt{\frac{K}{T}}$. Thus it suffices to show that $\sum_{a:a \text{ is } Bad} E[k_a]\triangle_a = O(\sqrt{KT})$. Recall that for any event A, we use 1_A for indicator r.v. that takes the value 1 if A happens and 0 otherwise.

For any bad arm a, we have

$$\begin{aligned}
E[k_a] &= E[|\{1 \le x \le T | I_{a\,x} > \mu_a + \frac{\triangle_a}{2}\}|] \\
&= E[\sum_{x=1}^{T} 1_{I_{a\,x} > \mu_a + \frac{\triangle_a}{2}}] \\
&= \sum_{x=1}^{T} E[1_{I_{a\,x} > \mu_a + \frac{\triangle_a}{2}}] \\
&= \sum_{x=1}^{T} \Pr(I_{a\,x} > \mu_a + \frac{\triangle_a}{2}) \\
&= \sum_{x=1}^{T} \Pr(\hat{\mu}_{a\,x} + \sqrt{\frac{\max(\log\frac{T}{Kx}, 0)}{x}} > \mu_a + \frac{\triangle_a}{2}) \\
&= \sum_{x=1}^{T} \Pr(\hat{\mu}_{a\,x} - \mu_a > \frac{\triangle_a}{2} - \sqrt{\frac{\max(\log\frac{T}{Kx}, 0)}{x}}) \\
&\le \sum_{x=1}^{8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a^2}} 1 + \sum_{x=8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a^2}}^{T} \Pr(\hat{\mu}_{a\,x} - \mu_a > \frac{\triangle_a}{2} - \sqrt{\frac{\max(\log\frac{T}{Kx}, 0)}{x}}) \\
&= 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a^2} + \sum_{x=\frac{8\log\frac{T\triangle_a^2}{K}}{\triangle_a^2}}^{T} \Pr(\hat{\mu}_{a\,x} - \mu_a > \frac{\triangle_a}{2} - \sqrt{\frac{\max(\log\frac{T}{Kx}, 0)}{x}})
\end{aligned}$$

As the arm a in the above calculations is bad, we have $\triangle_a > 5\sqrt{\frac{K}{T}}$. This implies that for $x \ge 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a^2}$, we have

$$\begin{aligned}
\frac{\max(\log\frac{T}{Kx}, 0)}{x} &\le \frac{\max(\log\frac{T}{K8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a^2}}, 0)}{8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a^2}} \\
&= \frac{\triangle_a^2}{8} \cdot \frac{\log\frac{T\triangle_a^2}{8K\log(T\triangle_a^2/K)}}{\log(T\triangle_a^2/K)} \\
&\le \frac{\triangle_a^2}{8}
\end{aligned}$$

The last inequality holds because $\log(T\triangle_a^2/K) > 1$ as $\triangle_a > 5\sqrt{\frac{K}{T}}$. Therefore

$$\begin{aligned}\Pr(\hat{\mu}_{a\,x} - \mu_a > \frac{\triangle_a}{2} - \sqrt{\frac{\max(\log\frac{T}{Kx}, 0)}{x}}) &\leq \Pr(\hat{\mu}_{a\,x} - \mu_a > \frac{\triangle_a}{2} - \frac{\triangle_a}{2\sqrt{2}}) \\ &\leq 2exp(-2c^2\triangle_a^2 x)\end{aligned}$$

where $c = \frac{1}{2}(1 - \frac{1}{\sqrt{2}})$.

Now

$$\begin{aligned}E[k_a] &\leq 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a^2} + \sum_{x=\frac{8\log\frac{T\triangle_a^2}{K}}{\triangle_a^2}}^{T} 2exp(-2c^2\triangle_a^2 x) \\ &\leq 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a^2} + \sum_{x=\frac{8\log\frac{T\triangle_a^2}{K}}{\triangle_a^2}}^{\infty} 2exp(-2c^2\triangle_a^2 x) \\ &= 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a^2} + \frac{exp(-2c^2\triangle_a^2 x_0)}{1 - exp(-2c^2\triangle_a^2)}\end{aligned}$$

where $x_0 = 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a^2}$ and last equality comes from the geometric series summation.

As $exp(-2c^2\triangle_a^2 x_0) < 1$ we have

$$E[k_a] \leq 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a^2} + \frac{1}{1 - exp(-2c^2\triangle_a^2)}$$

And hence

$$\triangle_a E[k_a] \leq 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a} + \frac{\triangle_a}{1 - exp(-2c^2\triangle_a^2)}$$

Using $1 - e^{-y} \geq y - y^2/2$ we have

$$\begin{aligned}\triangle_a E[k_a] &\leq 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a} + \frac{\triangle_a}{2c^2\triangle_a^2 - 2c^4\triangle_a^4} \\ &\leq 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a} + \frac{1}{2c^2\triangle_a(1 - c^2\triangle_a^2)} \\ &\leq 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a} + \frac{1}{2c^2\triangle_a(1 - c^2)} \\ &\leq 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a} + \frac{1}{2c^2(1 - c^2)}\frac{\sqrt{T}}{5\sqrt{K}}\end{aligned}$$

One can check that the maximum value of $f(y) = \frac{\log(Ty^2/K)}{y}$ is $O(\sqrt{\frac{T}{K}})$. Thus

$$\sum_{a:a \text{ is } bad} \triangle_a E[k_a] \leq \sum_{a:a \text{ is } bad} O(\frac{\sqrt{T}}{\sqrt{K}}) = O(\sqrt{KT})$$

All remains to show: $E[\delta] \leq 10\sqrt{\frac{K}{T}}$.

Stochastic Bandits

January 22, 2025

1 Concentration Inequalities

Theorem 1 (Chebyshev's Inequality)**.** *Let X be any random variable. Then for all $t > 0$,*

$$\Pr(|X - E[X]| \geq t) \leq \frac{Var(X)}{t^2}.$$

Theorem 2 (Markov inequality)**.** *Let X be any random variable that takes only non negative values. Then for any $c > 0$,*

$$\Pr(X \geq cE[X]) \leq 1/c.$$

Theorem 3 (Hoffding's inequality)**.** *Suppose $X_1, \ldots, X_n$ be independent bounded random variables such that for all $i \in [n]$, we have $a_i \leq X_i \leq b_i$ for all $i \in [n]$. Let $S_n = X_1 + \cdots + X_n$ and so $E[S_n] = \sum_i E[X_i]$.*

For any $t > 0$, we have

$$\Pr(|S_n - E[S_n]| \geq t) \leq 2e^{-\frac{2t^2}{\sum_i (b_i - a_i)^2}}$$

If $0 \leq X_i \leq 1$ for all $i \in [n]$ then we have,

$$\Pr(|S_n - E[S_n]| \geq t) \leq 2e^{-\frac{2t^2}{n}}$$

Let $\bar{X}_n = S_n/n = \frac{X_1 + \cdots + X_n}{n}$. So $E[\bar{X}_n] = \frac{\sum_i E[X_i]}{n}$ For any $t > 0$, we have

$$\Pr(|\bar{X}_n - E[\bar{X}_n]| \geq t) \leq 2e^{-\frac{2t^2 n^2}{\sum_i (b_i - a_i)^2}}$$

When for all i, $0 \leq X_i \leq 1$ then

$$\Pr(|\bar{X}_n - E[\bar{X}_n]| \geq t) \leq 2e^{-2t^2 n}$$

Remark. *Chebyshev's and Hoffding's inequalities works even for r.v. that can take negative values.*

2 Weak Law of Large Numbers

Let $X_1, \ldots, X_n$ be n iid (independent and identically distributed) random variables. Let $E[X] = \mu$ and $Var(X) = v$. Let $\bar{X}_n = \frac{\sum_i X_i}{n}$ be the empirical average. Note that $E[\bar{X}_n] = \mu$.

When $n \to \infty$ then the empirical average $\bar{X}_n$ converges to the true mean μ. We can see this from Chebyshev's inequality. As $X_1, \ldots, X_n$ are independent we have

$$Var(\bar{X}_n) = Var(\frac{\sum_i X_i}{n}) == \frac{nVar(X)}{n^2} = \frac{Var(X)}{n}.$$

So as $n \to \infty$, the variance of $\bar{X}_n$ tends to 0. So we have

$$\Pr(|\bar{X}_n - \mu| \geq \epsilon) \leq \frac{Var(X)}{n\epsilon^2} = \frac{v}{n\epsilon^2}$$

So for any fixed $\epsilon > 0$, for sufficiently large values of n (this value will depend on ϵ), the probability $\Pr(|\bar{X}_n - \mu| \geq \epsilon)$ will approach 0.

3 Stochastic Bandits

We start with the basic model of bandits, which is independent rewards. An algorithm has K possible arms to choose from, and there are T rounds, both K and T are known in advance (we will see later that one can relax the assumption that T should be known in advance). Each arm a is associated with a reward distribution D_a (pmf/pdf) which is unknown to the algorithm. The mean reward of the distribution D_a will be most relevant for us and is denoted by μ_a, i.e., $\mu_a = \underset{r \sim D_a}{E}[r]$ ($r \sim D_a$ denotes that r is sampled/drawn from the distribution D_a).

In each round $t \in T$:

1. algorithm picks an arm $a_t \in [K]$.
2. reward r_t is sampled from the distribution D_{a_t}.
3. algorithm receives the reward r_t.

The algorithm can be randomized, i.e., in any round it can fix a probability distribution over arms and can pick an arm from this distribution.

Again, it is important to note that $\mu_1, \ldots, \mu_K$ (and distributions $D_1, \ldots, D_K$) are unknown to the algorithm. The algorithm only sees the rewards (of the arms pulled by the algorithm).

We make the following assumptions:

1. rewards are bounded. For simplicty, we will assume rewards at any round will be in $[0, 1]$. So the means $\mu_1, \ldots, \mu_K$ all are in $[0, 1]$.
2. as we are in stochastic setting, we assume all drawn rewards are independent.

An important point to note that, regardless of the algorithm is deterministic or randomized, the arms pulled by the algorithm is a random variable. Because the algorithm's decision to pull any arm in some round will depend on past history of rewards observed by the algorithm. Since the received rewards are itself a random variable, the arms pulled are also random variable. This should become clear when we will see some algorithms.

Notations: Throughout the course, we will stick to the following notations. The set of arms will be denoted by $[K]$ and the total number of rounds will be denoted by T. The best arm is the arm a that has highest mean reward. We will use a^* for the best arm and μ^* for the mean reward of the best arm. That is $a^* = \arg\max_a \mu_a$ and $\mu^* = \mu_{a^*} = \max_a \mu_a$. For any suboptimal arm, we will use $\Delta_a := \mu^* - \mu_a$ for the gap of arm a. Finally, the arm pulled by the algorithm in tth round will be denoted by a_t.

Regret: How do we measure the performance of an algorithm? Recall that the goal of the algorithm is to maximize the sum of rewards received in all rounds.

One standard approach is to compare the algorithm performance with the best possible algorithm that knows the distributions D_a for all $a \in [K]$. Note that if means $\mu_1, \ldots, \mu_K$ are known then best strategy to maximize total expected rewards is to always pick the arm a^* to get total expected reward of $T\mu^*$. It makes sense to define the regret of an algorithm after T rounds as

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

where a_t is the arm pulled by the algorithm in the t th round. Thus $R(T)$ is the regret incurred by the algorithm after T rounds of not knowing the means $\mu_1, \mu_2, \ldots, \mu_K$. Note that the regret $R(T)$ is a random variable. We are interested in the expected regret $E[R(T)]$ of the algorithm.

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

The expectation in the above definition is taken over all the randomness, i.e., randomness over the draw of rewards from the distributions $D_1, \ldots, D_K$ and internal randomness of the algorithm (if the algorithm is randomized).

Remark. *1. By the definition only, $R(T)$ and $E[R(T)]$ of an algorithm depends on the problem-instance, i.e., means $\mu_1, \ldots, \mu_K$. Of course, we want an algorithm whose expected regret is small for all instances. In the previous line, 'for all problem instances' is important. For example, consider a stupid algorithm that always pulls arm 1. This algorithm will have regret 0 if the arm 1 has mean 1 and other arms have mean 0. But of course this algorithm will suffer for other instance such as if $\mu_1 = 0$ and $\mu_2 = 1$*

and will have regret of T. So we want an algorithm that performs well on all instances.

2. *If the regret is small then the algorithm's performance is close to best performance when the distributions are known. So we kind of learn the distributions.*

3. *If $r_1, \ldots, r_T$ are rewards received by the algorithm in rounds $1, 2, \ldots, T$ respectively then one can see that $E[\sum_{i=1}^{T} r_t] = E[\sum_{t=1}^{T} \mu_{a_t}]$. Hence the expected regret of an algorithm is the difference of expected total rewards of best strategy that knows $\mu_1, \ldots, \mu_K$ and the expected total rewards of the algorithm.*

Note: In some text books, $R(T)$ is called random-regret or realized regret and $E[R(T)]$ is called Regret. Its just a matter of convention, what names should we use. Often, from the context it will be clear whether we are talking about $R[T]$ or $E[R(T)]$.

Our goal is to design an algorithm whose expected regret $E[R(T)]$ is as small as possible for all problem-instances. Like a general discussion on algorithms, we will ignore constants and only consider Big-O dependence on T. Also, we assume $T \geq K$ as any reasonable algorithm will pull each arm atleast once. Note that since reward in any round is in $[0, 1]$, $R(T) \leq T$ always. We want an algorithms for which $R(T)$ grows sublinear in T, i.e., $\frac{R(T)}{T} \to 0$ as $T \to \infty$. In other words, average regret per round should be 0 when T is large (so we essentially we have learned the unknown distributions). Smaller the regret's dependence on T, faster the rate of convergence to 0 for the average regret per round.

3.1 Explore Then Commit Algorithm

This algorithm is very intuitive and perhaps we will first come up with this algorithm. If we know the means $\mu_1, \ldots, \mu_K$ then in each round the best algorithm will pick the arm a^* (recall notation, a^* has highest mean $\mu^* = \max_{a \in A}\{\mu_a\}$) to maximize total expected rewards. This algorithm first try each arm m times and finds the estimate $\hat{\mu}_a$ of the mean μ_a for every arm a. Thereafter, the algorithm will always pick an arm that maximizes the empirical mean $\hat{\mu}_a$ (we will call this arm a'). The value of m is set to $\frac{T^{2/3}(\log T)^{1/3}}{K^{2/3}}$ to optimze the expected regret.

We will now prove our first theorem of this course.

Theorem 4. *For any instance (i.e., for any values of unknowns $\mu_1, \ldots, \mu_K$), the expected regret of the algorithm ETC is $O(T^{2/3}(\log T)^{1/3}K^{1/3})$.*

Proof. Let $\epsilon = \sqrt{\frac{5 \ln T}{m}}$. For any arm a, let Bad_a be the event that $|\hat{\mu}_a - \mu_a| \geq \epsilon$. As $\hat{\mu}_a = \frac{\sum_{i=1}^{m} r_{ai}}{m}$ and all rewards are in $[0, 1]$, from Hoffding's inequality, we have

$$\Pr(Bad_a) = \Pr(|\hat{\mu}_a - \mu_a| \geq \epsilon) \leq \frac{2}{e^{2\epsilon^2 m}} = \frac{2}{T^{10}}.$$

Algorithm 1: Explore Then Commit

1. $m = \frac{T^{2/3}(\log T)^{1/3}}{K^{2/3}}$;
2. Try each arm m times. Let for arm a, $r_{a1}, r_{a2}, \ldots, r_{am}$ be the rewards received.;

```
/* Line 2 is exploration/investment phase.  In this phase,
   the algorithm tries each arm so even the worst arms.  But
   we hope that we will be able to find near-optimal arm for
   remaining rounds.                                        */
```

3. For each arm a, set $\hat{\mu}_a$ as the average received reward for the arm a, i.e., we have $\hat{\mu}_a = \frac{\sum_{i \in [m]} r_{ia}}{m}$. Let a' be the arm that maximizes $\hat{\mu}_a$, i.e., $a' = \arg\max_a \{\hat{\mu}_a\}$.;

/* We hope a' is the near-optimal arm, i.e., $\mu^* - \mu_{a'}$ is very small. */

4. From the round $m \cdot K + 1$, always pick the arm a'.

Let $Bad = \cup_a Bad_a$ be the event that for some arm a, $|\hat{\mu}_a - \mu_a| \geq \epsilon$ holds. So $Good = Bad^c$ is the event that, for all arms a, we have $|\hat{\mu}_a - \mu_a| < \epsilon$.

By union bound, we have

$$\Pr(Bad) \leq \sum_a \Pr(Bad_a) \leq K \cdot \frac{2}{T^{10}} \leq \frac{2}{T^9}$$

(as we are assuming $T \geq K$ as any reasonable algorithm will try each arm at least once).

Also,

$$\Pr(Good) = 1 - \Pr(Bad) \geq 1 - \frac{2}{T^9}$$

As the $\Pr(Bad)$ is negligible (we have chosen $\epsilon = \sqrt{\frac{5 \ln T}{m}}$ to ensure that this happens), it will suffice for us to only bound the expeceted regret conditioned on event Good. Following calculation formally show this.

$$\begin{aligned} E[R(T)] &= \Pr(Bad)E[R(T)|Bad] + \Pr(Good)E[R(T)|Good] \\ &\leq \frac{2}{T^9} \cdot T + 1 \cdot E[R(T)|Good] \\ &\leq \frac{2}{T^8} + \cdot E[R(T)|Good] \end{aligned}$$

the second inequality comes from $E[R(T)|Bad] \leq T$ as all rewards are in $[0, 1]$.

Bounding $E[R(T)|Good]$. We will now assume event $Good$, i.e., for all arms a, we have $|\hat{\mu}_a - \mu_a| < \epsilon$. Recall that $R(T) = \sum_{t=1}^{T} \mu^* - \mu_{a_t}$.

The contribution to regret in the investment phase can be at most mK (assuming worst case contribution of 1 in each round). The contribution to the

regret from $(mK+1)$st round till end is $(T-mK)(\mu^* - \mu_{a'})$ (recall that a' is the arm that maximizes $\hat{\mu}_a$ and is always chosen from $(mK+1)$st round) . We now claim that $(\mu^* - \mu_{a'}) < 2\epsilon$. For now let us assume this claim and bound the regret. We will prove the claim later.

$$\begin{aligned} R(T)|Good &\leq mK + (T-mK)2\epsilon \\ &\leq mK + 2T\epsilon \\ &= mK + 2T\sqrt{\frac{5\ln T}{m}} \end{aligned}$$

As m inreases mK increases and $2T\sqrt{\frac{5\ln T}{m}}$ decreases so the above quantity will be maximized when $mK = 2T\sqrt{\frac{5\ln T}{m}}$, i.e., when $m = (2\sqrt{5})^{2/3}\frac{T^{2/3}(\log T)^{1/3}}{K^{2/3}}$. Substituting this value of m we get $R(T)$ (conditioned on *Good*) equal to $O(T^{2/3}K^{1/3}(\log T)^{1/3})$. Obviously, we also have then
$E[R(T)|Good] = O(T^{2/3}K^{1/3}(\log T)^{1/3})$.

From our earlier calculation, $E[R(T)] = \frac{2}{T^8} + \cdot E[R(T)|Good] = O(T^{2/3}K^{1/3}(\log T)^{1/3})$. Now all it remains to prove the claim that $(\mu^* - \mu_{a'}) < 2\epsilon$. Note that $\hat{\mu}_{a'} \geq \hat{\mu}_{a^*} > \mu^* - \epsilon$ and also $\mu_{a'} > \hat{\mu}_{a'} - \epsilon$. Thus we have $\mu^* - \mu_{a'} < 2\epsilon$.

□

4 Successive Elimination Algorithm

One drawback of ETC algorithm is that it will continue to explore an arm large number of times (m times) even if an arm's reward history might suggest to not pull this arm further. In Successive Elimination algorithm, we discontinue the arm forever once we have belief that the arm is not good. Below is the high level description of this algorithm.

Algorithm 2: Successive Elimination - High Level Description

---

1) Pull every arm once;
2) If there is 'sufficient evidence' that some arm a is not a good arm then remove this arm;
Repeat the above steps over the remaining arms

Now to describe Successive Elimination fully, we just need to specify what 'sufficient evidence' is. For the same, we introduce some notations. For any arm a and round t, let $n_a(t)$ be the number of times the arm a is pulled till the round t. Obviously, we have $\sum_a n_a(t) = t$. Further, let $\hat{\mu}_a(t)$ be the empirical mean of received rewards from the arm a till round t. Formally, let r_{ai} be the reward received from the arm a on the ith pull. Then we have $\hat{\mu}_a(t) = \frac{\sum_{i=1}^{n_a(t)} r_{ai}}{n_a(t)}$. Let $\epsilon_a(t) = \sqrt{\frac{5 \log T}{n_a(t)}}$. Finally, let $UCB_a(t) = \hat{\mu}_a(t) + \epsilon_a(t)$ and $LCB_a(t) = \hat{\mu}_a(t) - \epsilon_a(t)$.

Now we describe the 'sufficent evidence' which Succeesive Elimination employs. Recall the analysis of ETC algorithm. There we define an event *Good* (and show that it holds with high probability) and show that conditioned on *Good*, $\hat{\mu}_a - \epsilon \le \mu \le \hat{\mu}_a + \epsilon$ where $\epsilon = \sqrt{\frac{5 \log T}{m}}$. Here also, we will define an event *Good* (and show it will hold with high probability) conditioned on which for all arm a and round t, we will have $LCB_a(t) \le \mu \le UCB_a(t)$. Now if at any time t, we have $UCB_a(t) < LCB_{a'}(t)$ for some arms a and a' then we know that $\mu_a < \mu_{a'}$. Hence, it is not a good strategy to pull arm a in any subsequent rounds because we will be better off pulling arm a'. In other words, we can eliminate the arm a for future.

Algorithm 3: Successive Elimination

Activate all the arms;
while $\#$-*rounds* $< T$ **do**
  Pull each active arm once (and receive rewards);
  Deactivate all arms a such that there exits some another arm a' with $UCB_a < LCB_{a'}$;
end

Theorem 5. *For all instances, i.e, for all values of $\mu_1, \ldots, \mu_K$,*

$$E[R(T)] = O(\sqrt{KT \log T})$$

Proof. Let *Good* be the following event: for all arms a and for all rounds t, we have $|\hat{\mu}_a(t) - \mu_a| < \epsilon_a(t)$ (that is, $LCB_a(t) \leq \mu_a \leq UCB_a(t)$). We will prove that

$$\Pr(Good) \geq 1 - O(\frac{1}{T^8})$$

For now, let us assume the above. Like ETC, it will suffice to bound $E[R(T)|Good]$.

$$\begin{aligned} E[R(T)] &= \Pr(Bad)E[R(T)|Bad] + \Pr(Good)E[R(T)|Good] \\ &\leq O(\frac{1}{T^8}) \cdot T + 1 \cdot E[R(T)|Good] \\ &\leq O(\frac{1}{T^7}) + \cdot E[R(T)|Good] \end{aligned}$$

the second inequality comes from $E[R(T)|Bad] \leq T$ as all rewards are in $[0, 1]$.

Bounding $E[R(T)|Good]$ *Each calculation in this paragraph is conditioned on Good.* Note that $R(T) = \sum_t (\mu^* - \mu_{a_t}) = \sum_a n_a(T)(\mu^* - \mu_a)$. If we show for each arm a, $(\mu^* - \mu_a) \leq (8\sqrt{\frac{5 \log T}{n_a(T)}})$ then we are done. This is because $R(T) = \sum_a n_a(T)(\mu^* - \mu_a) \leq \sum_a n_a(T)(8\sqrt{\frac{5 \log T}{n_a(T)}}) \leq \sum_a 8\sqrt{5 n_a(T) \log T}$. As $\sqrt{x}$ is a concave function so we have $\frac{\sum_a \sqrt{n_a(T)}}{K} \leq \sqrt{\frac{\sum_a n_a(T)}{K}} = \sqrt{T/K}$. Thus $R(T) \leq 8\sqrt{5KT \log T}$. So it remains to show that for any arm a, we have $(\mu^* - \mu_a) \leq 8\sqrt{\frac{5 \log T}{n_a(T)}}$. For the sake of analysis, we will refer to the ith iteration of while loop as phase i (for any i). Our first easy observation is that the arm a^* with will never be deactivated. Let t be the last round (corresponding to the end of some phase) where the arm a remained active (in other words, arm a was played exactly once after t). Note that $n_a(t) = n_{a^*}(t)$ (because both arms a and a^* are active till t so both of them are played equal number of times, which is the number of phases completed till t). As the arm a is not deactivated at t, we must have $UCB_a(t) \geq LCB_{a^*}(t)$. Further, we have $\epsilon_a(t) = \epsilon_{a^*}(t) = \sqrt{\frac{5 \log T}{n_a(t)}} = \sqrt{\frac{5 \log T}{n_a(T) - 1}}$. It is easy to now see that we have $\mu^* - \mu_a \leq 4\epsilon_a(t) = 4\sqrt{\frac{5 \log T}{n_a(T) - 1}} \leq 8\sqrt{\frac{5 \log T}{n_a(T)}}$.

Bounding $\Pr(Good)$ It remains to show :

$$\Pr(Good) \geq 1 - O(\frac{1}{T^8})$$

Let $r_{a1}, r_{a2}, \ldots, r_{aT}$ be T samples from the distribution D_a. Let $Bad_a(t)$ be the event that $|\frac{r_{a1} + \cdots + r_{at}}{t} - \mu_a| \geq \sqrt{\frac{5 \log T}{t}}$. By Hoffding's inequality, we have

$\Pr(Bad_a(t)) \leq O(1/e^{10\log T}) = O(\frac{1}{T^{10}})$. Let Bad_a be the event that for some $1 \leq t \leq T$, we have $|\frac{r_{a1}+\cdots+r_{at}}{t} - \mu_a| \geq \sqrt{\frac{5\log T}{t}}$. By union bound, $\Pr(Bad_a) \leq T \cdot O(\frac{1}{T^{10}}) = O(\frac{1}{T^9})$ Let $Bad = \cup_a Bad_a$. Again by union bound, $\Pr(Bad) \leq K \cdot O(\frac{1}{T^9}) = O(\frac{1}{T^8})$.

□

The regret in the above theorem is worst-case regret, i.e, for any problem-instance (that is, for any values of K, T and $\mu_1, \ldots, \mu_K$, the expected regret $E[R(T)] \leq O(\sqrt{KT\log T})$. Now we will show another type of bounds on expected regret, which will be instance -dependent bound.

Theorem 6. *The expected regret of Successive Elimination satisfies*

$$E[R(T)] \leq O(\log T) \sum_{a:\triangle_a>0} \frac{1}{\triangle_a}$$

where $\triangle_a = \mu^* - \mu_a$ *is the gap of arm* a.

Proof. We define the events *Bad* and *Good* as in the above theorem. Again it will suffice to bound $E[R(T)|Good]$. All calculations now are conditioned on *Good*. We claim that for any suboptimal arm a, we have $n_a(T) \leq 50000\frac{\log T}{\triangle_a^2}$. This implies that $R(T) = \sum_a n_a(T)\triangle_a \leq O(\log T)\sum_{a:\triangle_a>0}\frac{1}{\triangle_a}$. Suppose $n_a(T) > 50000\frac{\log T}{\triangle_a^2}$. Consider the time t when $n_a(t) = 50000\frac{\log T}{\triangle_a^2}$. As the arm a^* is always active we also have $n_{a^*}(t) = 50000\frac{\log T}{\triangle_a^2}$. So now we have $\epsilon_a(t) = \epsilon_{a^*}(t) = \sqrt{\frac{5\log T}{n_a(t)}} = \triangle_a/100$. As we have assumed *Good*, we have $LCB_{a^*}(t) \geq \mu^* - \triangle_a/100$ and $UCB_a(t) \leq \mu_a + \triangle_a/100$. So we have $UCB_a(t) < LCB_{a^*}(t)$ which implies that arm a will be eliminated in round t which contradicts that $n_a(T) > 50000\frac{\log T}{\triangle_a^2}$.

□

UCB Algorithm

Let $n_a(t), \epsilon_a(t), \hat{\mu}_a, UCB_a(t)$ be as defined before (in Successive Elimination algorithm).

The idea of UCB is to add bonus to the empirical mean and then pick an arm that has highest value of empirical mean plus bonus. The bonus is chosen to be $\epsilon_a(t)$ which is equal to $\sqrt{\frac{5 \log T}{n_a(t)}}$ (since $\hat{\mu}_a(t) + \epsilon_a(t) = UCB_a(t)$, the algorithm, at any time t, pulls an arm that has highest value of $UCB_a(t)$).

Let us now go into the intuition behind the UCB algorithm in detail. During the initial rounds, the difference between the empirical mean and the actual mean of an arm can be significant. Consequently, selecting an arm solely based on the maximum empirical mean value is not a good strategy. To address this issue, the algorithm incorporates a bonus term. If an arm a is underexplored, the bonus is large, which encourages the algorithm to explore that arm (even if its empirical mean is small at this time). One concern might be whether the bonus term could lead the algorithm to pull bad arms excessively. However, this scenario will not happen because the bonus term diminishes as the number of pulls for the arm increases.

Algorithm 4: UCB Algorithm

$UCB_a = \infty$ for all arms a;
`/* Initialization */`
while *#-rounds* $< T$ **do**
  Pull an arm that has the highest value of UCB_a;
  `/* Recall that` $UCB_a(t) = \hat{\mu}_a(t) + \epsilon_a(t)$ `*/`
end

UCB acheives the same guarantee on expected regret as that of Successive Elimination. The proof is almost same so here we do not give a complete proof.

Theorem 7. *For all instances, i.e, for all values of* $\mu_1, \ldots, \mu_K$,

$$E[R(T)] = O(\sqrt{KT \log T})$$

and

$$E[R(T)] = O(\log T) \sum_{a:\triangle_a > 0} \frac{1}{\triangle_a}$$

where $\triangle_a = \mu^* - \mu_a$ *is the gap of arm* a.

Proof. The proof is almost same as that of Successive Elimination. To prove $E[R(T)] = O(\sqrt{KT \log T})$, we now show that for any arm a, $\mu^* - \mu_a \le 2\sqrt{\frac{5 \log T}{n_a(T)}}$ (assuming *Good*) (and then the rest of the proof is exactly same). Let t_a be the last round when the arm a was pulled. Thus $n_a(T) = n_a(t)$. Note $\mu_a \ge UCB_a(t) - 2\epsilon_a(t)$ (as we are assuming *good*), $UCB_{a^*} \ge \mu^*$ (as we are assuming

Good) and $UCB_{a^*}(t_a) \leq UCB_a(t_a)$ (as arm a was pulled in round t_a). Hence, $\mu^* - \mu_a \leq 2\sqrt{\frac{5\log T}{n_a(t_a)}} = 2\sqrt{\frac{5\log T}{n_a(T)}}$. Now the proof goes the same was as in Successive Elimination algorithm.

The proof of instance dependent bound is also similar. We here prove that (assuming *Good*) for any suboptimal arm a, we have $n_a(T) \leq 50000\frac{\log T}{\triangle_a^2}$ (and then the proof is exactly same). Suppose not. Consider a time t when $n_a(t) = 50000\frac{\log T}{\triangle_a^2}$. We have $\epsilon_a(t) = \triangle_a/100$. Note that for any time $t' > t$, we have $\epsilon_a(t') \leq \epsilon_a(t)$. For any time $t' > t$, we have $UCB_a(t') \leq \mu_a + 2\epsilon_a(t') \leq \mu_a + 2\epsilon_a(t) < \mu^* \leq UCB_{a^*}(t')$. This means that for any $t' > t$ we will have $UCB_{a^*}(t') > UCB_a(t')$ which means that arm a will never be pulled after t. This contradicts that $n_a(t) > 50000\frac{\log T}{\triangle_a^2}$.

$\square$

MOSS algorithm (UCB2)

MOSS is a variant of UCB and it has the expected regret of $O(\sqrt{KT})$ and hence it beats both Successive Elimination and UCB in theory. In the next lecture, we will see that no algorithm can have smaller expected regret than $O(\sqrt{KT})$ and hence MOSS is the best possible algorithm.

MOSS is same as UCB expect how bonus is calculated. Now the bonus is set to $\sqrt{\frac{\max(\log \frac{T}{Kn_a(t)}, 0)}{n_a(t)}}$. Let

$$I_a(t) = \hat{\mu}_a(t) + \sqrt{\frac{\max(\log \frac{T}{Kn_a(t)}, 0)}{n_a(t)}}.$$

At any time t, MOSS pulls an arm that has a maximum value of I_a. One reason MOSS has better expected regret bound than UCB is that the estimation precision when $n_a(t)$ is large is more accurate in MOSS than UCB. In other words, the bonus more quickly goes to 0 in MOSS than UCB as $n_a(t)$ increases.

Algorithm 5: MOSS Algorithm

```
I_a = ∞ for all arms a;
/* Initialization                                              */
while #-rounds < T do
 | Pull an arm that has the highest value of I_a;
end
```

Theorem 8. *The expected regret $E[R(T)]$ of MOSS satisfies*

$$E[R(T)] = O(\sqrt{KT})$$

Proof. We will use a trick that is frequently employed in analysis of randomized algorithms. Instead of sampling from the distribution D_a at the time when the arm a is pulled, we assume that T independent samples from every distribution D_a has already been sampled before the start of the algorithm. For each arm a, let $r_{a1}, \ldots, r_{aT}$ be the T independent samples drawn from the distribution D_a. Now when the algorithm pulls arm a, we provide sample (to the algorithm) from $r_{a1}, \ldots, r_{aT}$. In particular, the sample provided to the algorithm for ith pull of arm a is r_{ai}.

Let us define new notations. Let for any $1 \le x \le T$, $\hat{\mu}_{a\,x} = \frac{\sum_{j=1}^{x} r_{aj}}{x}$ be the average of first x rewards (of T samples drawn beforehand).With respect to this new notation, note that $\hat{\mu}_a(t) = \hat{\mu}_{a\,n_a(t)}$. Further, for any $1 \le x \le T$, let $I_{a\,x} = \hat{\mu}_{a\,x} + \sqrt{\frac{\max(\log \frac{T}{Kx}, 0)}{x}}$. Again note that $I_a(t) = I_{a\,n_a(t)}$. We will call $I_a(t)$ as the index of arm a after time t.

Let $\delta = \max\{\mu^* - \min_{1 \le x \le T} I_{a^*\,x}, 0\}$. Note that δ is a random variable. By definition only, the index of the best arm will never be less than $\mu^* - \delta$, i.e.,

$I_{a^*}(t) \geq \mu^* - \delta$ for all t. We will prove later that $E[\delta] \leq 10\sqrt{\frac{K}{T}}$. It will be helpful to keep this fact in mind.

Let us call an arm a as *Good* if $\triangle_a \leq 5\sqrt{\frac{K}{T}}$ (note that this is different - in previous algorithms, *Good* and *Bad* were events). An arm a is called *Bad* if $\triangle_a > 5\sqrt{\frac{K}{T}}$. Now it will be clear why we have defined *Good* and *Bad* arms in this way. Recall that $R(T) = \sum_a R_a(T)$ where $R_a(T) = n_a(T)\triangle_a$.

$$
\begin{aligned}
R(T) &= \sum_{a:a \text{ is } Good} R_a(T) + \sum_{a:a \text{ is } Bad} R_a(T) \\
&\leq \sum_{a:a \text{ is } Good} n_a(T) 5\sqrt{\frac{K}{T}} + \sum_{a:a \text{ is } Bad} R_a(T) \\
&\leq 5\sqrt{\frac{K}{T}} \sum_{a:a \text{ is } Good} n_a(T) + \sum_{a:a \text{ is } Bad} R_a(T) \\
&\leq 5\sqrt{\frac{K}{T}} \cdot T + \sum_{a:a \text{ is } Bad} R_a(T) \\
&\leq 5\sqrt{KT} + \sum_{a:a \text{ is } Bad} R_a(T)
\end{aligned}
$$

Thus it suffices to show $\sum_{a:a \text{ is } Bad} R_a(T) = O(\sqrt{KT})$. Let us introduce few more notations. For any bad arm a, we define a value k_a (which is a random variable) as follows:

$$k_a = |\{1 \leq x \leq T | I_{a\,x} > \mu_a + \frac{\triangle_a}{2}\}|$$

We also define J which is is a random subset of bad arms defined as below:

$$J = \{a \in [K] | a \text{ is } Bad \text{ and } \triangle_a > 2\delta\}$$

A very important observation is that for any arm in J, we have $n_a(T) \leq k_a$. This is the main crux of the analysis. As directly showing bounds for $E[n_a(t)]$ is difficult but later we will be able to show bounds for $E[k_a]$ for bad arms.

Now

$$
\begin{aligned}
\sum_{a:a \text{ is } Bad} R_a(T) &= \sum_{a \in J} n_a(T)\triangle_a + \sum_{a \notin J:a \text{ is } Bad} n_a(T)\triangle_a \\
&\leq \sum_{a:a \text{ is } Bad} k_a\triangle_a + 2\delta T
\end{aligned}
$$

Now

$$
\begin{aligned}
E[\sum_{a:a \text{ is } Bad} R_a(T)] &\leq \sum_{a:a \text{ is } Bad} E[k_a]\triangle_a + 2E[\delta]T \\
&\leq \sum_{a:a \text{ is } Bad} E[k_a]\triangle_a + 20\sqrt{KT}
\end{aligned}
$$

as we earlier claimed (without proof) that $E[\delta] \leq 10\sqrt{\frac{K}{T}}$. Thus it suffices to show that $\sum_{a:a \text{ is } Bad} E[k_a]\triangle_a = O(\sqrt{KT})$. Recall that for any event A, we use 1_A for indicator r.v. that takes the value 1 if A happens and 0 otherwise.

For any bad arm a, we have

$$\begin{aligned}
E[k_a] &= E[|\{1 \leq x \leq T | I_{a\,x} > \mu_a + \frac{\triangle_a}{2}\}|] \\
&= E[\sum_{x=1}^{T} 1_{I_{a\,x} > \mu_a + \frac{\triangle_a}{2}}] \\
&= \sum_{x=1}^{T} E[1_{I_{a\,x} > \mu_a + \frac{\triangle_a}{2}}] \\
&= \sum_{x=1}^{T} \Pr(I_{a\,x} > \mu_a + \frac{\triangle_a}{2}) \\
&= \sum_{x=1}^{T} \Pr(\hat{\mu}_{a\,x} + \sqrt{\frac{\max(\log\frac{T}{Kx}, 0)}{x}} > \mu_a + \frac{\triangle_a}{2}) \\
&= \sum_{x=1}^{T} \Pr(\hat{\mu}_{a\,x} - \mu_a > \frac{\triangle_a}{2} - \sqrt{\frac{\max(\log\frac{T}{Kx}, 0)}{x}}) \\
&\leq \sum_{x=1}^{8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a^2}} 1 + \sum_{x=8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a^2}}^{T} \Pr(\hat{\mu}_{a\,x} - \mu_a > \frac{\triangle_a}{2} - \sqrt{\frac{\max(\log\frac{T}{Kx}, 0)}{x}}) \\
&= 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a^2} + \sum_{x=\frac{8\log\frac{T\triangle_a^2}{K}}{\triangle_a^2}}^{T} \Pr(\hat{\mu}_{a\,x} - \mu_a > \frac{\triangle_a}{2} - \sqrt{\frac{\max(\log\frac{T}{Kx}, 0)}{x}})
\end{aligned}$$

As the arm a in the above calculations is bad, we have $\triangle_a > 5\sqrt{\frac{K}{T}}$. This implies that for $x \geq 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a^2}$, we have

$$\begin{aligned}
\frac{\max(\log\frac{T}{Kx}, 0)}{x} &\leq \frac{\max(\log\frac{T}{K8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a^2}}, 0)}{8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a^2}} \\
&= \frac{\triangle_a^2}{8} \cdot \frac{\log\frac{T\triangle_a^2}{8K\log(T\triangle_a^2/K)}}{\log(T\triangle_a^2/K)} \\
&\leq \frac{\triangle_a^2}{8}
\end{aligned}$$

The last inequality holds because $\log(T\triangle_a^2/K) > 1$ as $\triangle_a > 5\sqrt{\frac{K}{T}}$. Therefore

$$\begin{aligned}
\Pr(\hat{\mu}_{a\,x} - \mu_a > \frac{\triangle_a}{2} - \sqrt{\frac{\max(\log\frac{T}{Kx}, 0)}{x}}) &\leq \Pr(\hat{\mu}_{a\,x} - \mu_a > \frac{\triangle_a}{2} - \frac{\triangle_a}{2\sqrt{2}}) \\
&\leq 2exp(-2c^2\triangle_a^2 x)
\end{aligned}$$

where $c = \frac{1}{2}(1 - \frac{1}{\sqrt{2}})$.

Now

$$\begin{aligned}
E[k_a] &\leq 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a^2} + \sum_{x=\frac{8\log\frac{T\triangle_a^2}{K}}{\triangle_a^2}}^{T} 2exp(-2c^2\triangle_a^2 x) \\
&\leq 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a^2} + \sum_{x=\frac{8\log\frac{T\triangle_a^2}{K}}{\triangle_a^2}}^{\infty} 2exp(-2c^2\triangle_a^2 x) \\
&= 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a^2} + \frac{exp(-2c^2\triangle_a^2 x_0)}{1 - exp(-2c^2\triangle_a^2)}
\end{aligned}$$

where $x_0 = 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a^2}$ and last equality comes from the geometric series summation.

As $exp(-2c^2\triangle_a^2 x_0) < 1$ we have

$$E[k_a] \leq 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a^2} + \frac{1}{1 - exp(-2c^2\triangle_a^2)}$$

And hence

$$\triangle_a E[k_a] \leq 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a} + \frac{\triangle_a}{1 - exp(-2c^2\triangle_a^2)}$$

Using $1 - e^{-y} \geq y - y^2/2$ we have

$$\begin{aligned}
\triangle_a E[k_a] &\leq 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a} + \frac{\triangle_a}{2c^2\triangle_a^2 - 2c^4\triangle_a^4} \\
&\leq 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a} + \frac{1}{2c^2\triangle_a(1 - c^2\triangle_a^2)} \\
&\leq 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a} + \frac{1}{2c^2\triangle_a(1 - c^2)} \\
&\leq 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a} + \frac{1}{2c^2(1 - c^2)}\frac{\sqrt{T}}{5\sqrt{K}}
\end{aligned}$$

One can check that the maximum value of $f(y) = \frac{\log(Ty^2/K)}{y}$ is $O(\sqrt{\frac{T}{K}})$. Thus

$$\sum_{a:a \text{ is } bad} \triangle_a E[k_a] \leq \sum_{a:a \text{ is } bad} O(\frac{\sqrt{T}}{\sqrt{K}}) = O(\sqrt{KT})$$

All remains to show: $E[\delta] \leq 10\sqrt{\frac{K}{T}}$. $\square$

5 Lower Bounds

We started with ETC and show its expected regret of $O(T^{2/3}K^{1/3}(\log T)^{1/3})$. Then we show that the expected regret of Successive Elimination and UCB is $O(\sqrt{KT\log T})$. Finally, we show that MOSS beats UCB and Successive Elimination and has expected regret of $O(\sqrt{KT})$. What next? Can we have an algorithm that has even better regret than MOSS, i.e., $O(\sqrt{KT})$? What if researchers are unable to find a better regret bound then can we conclude that better bound is not possible? Surely not. It could be lack of understanding or knowledge, preventing us from coming up with a better algorithm.

Now our focus will be lower bounds, i.e., we will be show that no algorithm can have better regret bound than some threshold. In fact we will prove the following theorem

Theorem 9. *No algorithm (deterministic or randomized) can have expected regret of* $\leq \sqrt{KT}/100$ *for all instances.*

Note that the possibilities of algorithms are infinite. An algorithm can perhaps do anything. Formally, a deterministic algorithm is any function that takes as input K, T and for any $1 \leq t \leq T$ maps a sequence (of pairs consisting of arm and received reward) $\{(a_1, r_1), (a_2, r_2), \ldots, (a_{t-1}, r_{t-1})\}$ to some arm $a_t \in [K]$. Since rewards can take any value in $[0, 1]$, the number of distinct algorithms is infinite. Thus, it is not possible to prove the above theorem by showing that a particular algorithm does not have a better regret bound for all such algorithms (as the number of such algorithms is infinite).

Not surprisingly, we will need new tools to prove such lower bound. We now will study elementary tools from Information theory which will be sufficient to prove the above theorem. In fact we will not directly prove the Theorem 9. First we will consider a toy problem (coin testing problem of exercise) and show a lower bound for this problem. This itself will require new tools from Information Theory. Then after getting some familiarity with these tools, we will prove the Theorem 9.

Testing Coin Problem: It is promised that a gicen coin is either fair, i.e, $p = 1/2$ or $p = 1/2 + \epsilon$ (algorithm knows ϵ). Given an algorithm A that takes samples from that coin and has the following guarantee:

- If $p = 1/2$, algorithm must say 'fair' with at least $4/5$ probability.
- If $p = 1/2+\epsilon$, the algorithm must say 'biased' with at least $4/5$ probability.

Lemma 1. *There is a deterministic algorithm that solves Testing coin problem and needs only* $100/\epsilon^2$ *samples.*

We will now see the lower bound:

Lemma 2. *Let* $\epsilon \leq 1/3$. *Any algorithm (deterministic or randomized) will need at least* $\Omega(1/\epsilon^2)$ *samples to solve Testing Coin problem.*

Note that we are only concerned with the dependence on ϵ ignoring constances ($\Omega(1/\epsilon^2)$ means $\geq c \cdot 1/\epsilon^2$ for some constant c (that does not depend on ϵ) Now the goal is to prove Lemma 2. As said before, we begin with understanding new tools.

Total Variation Distance and KL divergence

Recall that a probability distribution P over a finite domain Ω is a function $P : \Omega \to [0, 1]$ that maps each $\omega \in \Omega$ to a real in $[0, 1]$ satisfying:

1. $P(\omega) \geq 0$ for all $\omega \in \Omega$
2. $\sum_{\omega \in \Omega} P(\omega) = 1$.

Now we will define the notion of distance between two probability distributions. Any notion of distance d must necessarily satisfy these two properies - $d(P, P) = 0$ for any distribution P and $d(P, Q) > 0$ for any two distinct distribution P and Q. There can be many ways to define distance. for instance, we can consider Euclidean distance or ℓ_2 norm: $d(P, Q) = \sum_\omega (P(\omega) - Q(\omega))^2$ or we can consider ℓ_1 norm called as total variation distance $d(P, Q) = \sum_\omega |P(\omega) - Q(\omega)|$. Each notion of distance has its own merits and limitaions. For us, the notion of total variation distance also known as statistical distance will be important.

Definition 1. *For any two distributions P and Q defined on a finite domain Ω, the total variation distance between P and Q, denoted as $d_{tv}(P, Q)$ is defined as*

$$d_{tv}(P, Q) = \sum_{\omega \in \Omega} |P(\omega) - Q(\omega)|$$

It is easy to see that:

1. $d_{tv}(P, Q) = 0$ if $P = Q$.
2. $d_{tv}(P, Q) > 0$ for any $P \neq Q$.
3. $d_{tv}(P, Q) = d_{tv}(Q, P)$.
4. $d_{tv}(P, Q) \leq d_{tv}(P, R) + d_{tv}(R, Q)$ (triangle inequality)

Now we are going to see one of the most useful and remarkable property of total variation distance.

Lemma 3.

$$d_{tv}(P, Q) = \max_{S \subseteq \Omega} |P(S) - Q(S)| \tag{1}$$

where note that $P(S) = \sum_{\omega \in S} P(\omega)$ and $Q(S) = \sum_{\omega \in S} Q(\omega)$

Now we define KL divergence between two distributions.

Definition 2. *The KL divergence $KL(P, Q)$ is defined as*

$$KL(P, Q) = \sum_{\omega \in \Omega} P(\omega) \ln \frac{P(\omega)}{Q(\omega)}$$

The KL divergence satisfy:

Lemma 4. • $KL(P, P) = 0$

• $KL(P, Q) > 0$ *if* $P \neq Q$

However, note that $KL(P, Q) \neq KL(Q, P)$ and also it does not satisfy the triangle inequality. That is why it is called divergence not distance. The usefulness of KL divergence comes from the fact that the KL divergence of a product distribution just adds up. Let us see first what is a product distribution and then the utility of KL divergence (I have talked about the product distribution while discussing independent trials).

Definition 3. *Given a distribution P, $P^{\otimes 2}$ is a distribution on domain $\Omega \times \Omega$ such that*

$$P^{\otimes 2}(a, b) = P(a)P(b) \quad \forall a, b \in \Omega$$

In general, n-fold product distrtion $P^{\otimes n}$ is a distribution on domain $\times_{i=1}^{n} \Omega$ such that

$$P^{\otimes 2}(\omega_1, \omega_2, \ldots \omega_n) = P(\omega_1)P(\omega_2) \cdots P(\omega_n) \quad \forall \omega_1, \ldots, \omega_n \in \Omega$$

The product distribution corresponds to independent trials. *Drawing n independent samples from a distribution P is exactly same as dwawing one sample from $P^{\otimes n}$.* Now we will see why KL divergence is extermly useful.

Lemma 5.

$$KL(P^{\otimes n}, Q^{\otimes n}) = n \cdot KL(P, Q)$$

The above property is not true for total variation distance. We do not have $d_{tv}(P^{\otimes n}, Q^{\otimes n}) = n d_{tv}(P, Q)$. In fact, other than $d_{tv}(P^{\otimes n}, Q^{\otimes n}) \leq n d_{tv}(P, Q)$, we do not know of any other inequality betwenn these two.

We have now seen both total variation distance and KL divergence, each having its own merits and limitations. Now we will see that these two are connected by an inequality, which forms the basis of powerful application.

Lemma 6 (Pinsker inequality)**.** *For any two probability distributions P and Q, we have*

$$d_{tv}(P, Q) \leq \sqrt{\frac{1}{2} KL(P, Q)}$$

Following is also useful.

Lemma 7 (The Bretagnolle–Huber bound)**.**

$$d_{tv}(P, Q) \leq \sqrt{1 - e^{-KL(P,Q)}} \leq 1 - \frac{1}{2} e^{-KL(P,Q)}$$

Now we have all the necessary tool to prove Lemma 2. First we will prove the lemma for deterministic algorithms.

Proof of Lemma 2 Suppose there is a deterministic algorithm A that solves the Testing Coin problem and uses m independent coin tosses. Note that algorithm A is just a function $A : \{H, T\}^m \to \{\text{Fair}, \text{Biased}\}$ that maps a sequence of head and tails of length m to Fair or Biased. We partition $\{H, T\}^m$ into two subsets as follows: $A_f = \{y \in \{0,1\}^m : A(y) = \text{Fair}\}$ and $A_b = \{y \in \{0,1\}^m : A(y) = \text{Biased}\}$, i.e., A_f consists of all those sequence which gets mapped to Fair and A_b consists of all those sequences that gets mapped to Biased.

Let $p = \Pr(Head)$. Let $(\frac{1}{2})^{\otimes m}$ and $(\frac{1}{2} + \epsilon)^{\otimes m}$ be the product distributions corresponding to m independent coin tosses with $p = 1/2$ and $p = 1/2 + \epsilon$ respectively. For convenience, let $D_f = (\frac{1}{2})^{\otimes m}$ and $D_b = (\frac{1}{2} + \epsilon)^{\otimes m}$. We will bound tv distance between $D_f = (\frac{1}{2})^{\otimes m}$ and $D_b = (\frac{1}{2} + \epsilon)^{\otimes m}$ in two different ways.

Note that $D_f(A_f) = \sum_{\omega \in A_f} D_f(\omega) \geq 4/5$ and $D_b(A_f) = \sum_{\omega \in A_f} D_b(\omega) < 1/5$ (because the algorithm A solves the Testing Coin problem). Thus by Lemma 3,

$$d_{tv}(D_f, D_b) \geq D_f(A_f) - D_b(A_f) \geq 3/5.$$

Now let us bound tv distance using Pinsker inequality. As said earlier, calculation of KL divergence between D_f and D_b is easy as they both are product distributions.

$$KL(\frac{1}{2})^{\otimes m}, (\frac{1}{2} + \epsilon)^{\otimes m}) = m \cdot KL(1/2, 1/2 + \epsilon)$$

where $KL(1/2, 1/2+\epsilon)$ is the KL divergence between the distributions $(1/2, 1/2)$ and $(1/2 + \epsilon, 1/2 - \epsilon)$.

$$\begin{aligned} KL(1/2, 1/2 + \epsilon) &= \frac{1}{2} \ln \frac{1}{2(\frac{1}{2} + \epsilon)} + \frac{1}{2} \ln \frac{1}{2(\frac{1}{2} - \epsilon)} \\ &= \frac{1}{2} \ln \frac{1}{4(1/4 - \epsilon^2)} = \frac{1}{2} \ln \frac{1}{(1 - 4\epsilon^2)} \\ &= \frac{1}{2}(4\epsilon^2 + \frac{(4\epsilon^2)^2}{2} + \frac{(4\epsilon^2)^3}{3} + \dots) = 2\epsilon^2(1 + 2\epsilon^2 + 16\epsilon^4/3 + \cdots) \leq 10\epsilon^2 \end{aligned}$$

for $\epsilon \leq 1/3$ (probably 10 can be replaced by a smaller number but as we are not optimizing constants I put here bigger number so that the inequality is easy to derive).

Now we will use Pinsker inequality to bound the tv distance.

$$\begin{aligned} d_{tv}((\frac{1}{2})^{\otimes m}, (\frac{1}{2} + \epsilon)^{\otimes m}) &\leq \sqrt{\frac{1}{2} KL((\frac{1}{2})^{\otimes m}, (\frac{1}{2} + \epsilon)^{\otimes m})} \\ &\leq \sqrt{5m}\epsilon \end{aligned}$$

Now we have

$$3/5 \leq d_{tv}(D_f, D_b) \leq \sqrt{\frac{1}{2} KL((\frac{1}{2})^{\otimes m}, (\frac{1}{2} + \epsilon)^{\otimes m})} \leq \sqrt{5m}\epsilon$$

which implies that $m \geq \frac{9}{125} \frac{1}{\epsilon^2}$.

Stochastic Bandits

January 27, 2025

1 Concentration Inequalities

Theorem 1 (Chebyshev's Inequality). *Let X be any random variable. Then for all $t > 0$,*

$$\Pr(|X - E[X]| \geq t) \leq \frac{Var(X)}{t^2}.$$

Theorem 2 (Markov inequality). *Let X be any random variable that takes only non negative values. Then for any $c > 0$,*

$$\Pr(X \geq cE[X]) \leq 1/c.$$

Theorem 3 (Hoffding's inequality). *Suppose $X_1, \ldots, X_n$ be independent bounded random variables such that for all $i \in [n]$, we have $a_i \leq X_i \leq b_i$ for all $i \in [n]$. Let $S_n = X_1 + \cdots + X_n$ and so $E[S_n] = \sum_i E[X_i]$.*

For any $t > 0$, we have

$$\Pr(|S_n - E[S_n]| \geq t) \leq 2e^{-\frac{2t^2}{\sum_i (b_i - a_i)^2}}$$

If $0 \leq X_i \leq 1$ for all $i \in [n]$ then we have,

$$\Pr(|S_n - E[S_n]| \geq t) \leq 2e^{-\frac{2t^2}{n}}$$

Let $\bar{X}_n = S_n/n = \frac{X_1 + \cdots + X_n}{n}$. So $E[\bar{X}_n] = \frac{\sum_i E[X_i]}{n}$ For any $t > 0$, we have

$$\Pr(|\bar{X}_n - E[\bar{X}_n]| \geq t) \leq 2e^{-\frac{2t^2 n^2}{\sum_i (b_i - a_i)^2}}$$

When for all i, $0 \leq X_i \leq 1$ then

$$\Pr(|\bar{X}_n - E[\bar{X}_n]| \geq t) \leq 2e^{-2t^2 n}$$

Remark. *Chebyshev's and Hoffding's inequalities works even for r.v. that can take negative values.*

2 Weak Law of Large Numbers

Let $X_1, \ldots, X_n$ be n iid (independent and identically distributed) random variables. Let $E[X] = \mu$ and $Var(X) = v$. Let $\bar{X}_n = \frac{\sum_i X_i}{n}$ be the empirical average. Note that $E[\bar{X}_n] = \mu$.

When $n \to \infty$ then the empirical average $\bar{X}_n$ converges to the true mean μ. We can see this from Chebyshev's inequality. As $X_1, \ldots, X_n$ are independent we have

$$Var(\bar{X}_n) = Var(\frac{\sum_i X_i}{n}) == \frac{nVar(X)}{n^2} = \frac{Var(X)}{n}.$$

So as $n \to \infty$, the variance of $\bar{X}_n$ tends to 0. So we have

$$\Pr(|\bar{X}_n - \mu| \geq \epsilon) \leq \frac{Var(X)}{n\epsilon^2} = \frac{v}{n\epsilon^2}$$

So for any fixed $\epsilon > 0$, for sufficiently large values of n (this value will depend on ϵ), the probability $\Pr(|\bar{X}_n - \mu| \geq \epsilon)$ will approach 0.

3 Stochastic Bandits

We start with the basic model of bandits, which is independent rewards. An algorithm has K possible arms to choose from, and there are T rounds, both K and T are known in advance (we will see later that one can relax the assumption that T should be known in advance). Each arm a is associated with a reward distribution D_a (pmf/pdf) which is unknown to the algorithm. The mean reward of the distribution D_a will be most relevant for us and is denoted by μ_a, i.e., $\mu_a = \underset{r \sim D_a}{E}[r]$ ($r \sim D_a$ denotes that r is sampled/drawn from the distribution D_a).

In each round $t \in T$:

1. algorithm picks an arm $a_t \in [K]$.

2. reward r_t is sampled from the distribution D_{a_t}.

3. algorithm receives the reward r_t.

The algorithm can be randomized, i.e., in any round it can fix a probability distribution over arms and can pick an arm from this distribution.

Again, it is important to note that $\mu_1, \ldots, \mu_K$ (and distributions $D_1, \ldots, D_K$) are unknown to the algorithm. The algorithm only sees the rewards (of the arms pulled by the algorithm).

We make the following assumptions:

1. rewards are bounded. For simplicty, we will assume rewards at any round will be in $[0, 1]$. So the means $\mu_1, \ldots, \mu_K$ all are in $[0, 1]$.

2. as we are in stochastic setting, we assume all drawn rewards are independent.

An important point to note that, regardless of the algorithm is deterministic or randomized, the arms pulled by the algorithm is a random variable. Because the algorithm's decision to pull any arm in some round will depend on past history of rewards observed by the algorithm. Since the received rewards are itself a random variable, the arms pulled are also random variable. This should become clear when we will see some algorithms.

Notations: Throughout the course, we will stick to the following notations. The set of arms will be denoted by $[K]$ and the total number of rounds will be denoted by T. The best arm is the arm a that has highest mean reward. We will use a^* for the best arm and μ^* for the mean reward of the best arm. That is $a^* = \arg\max_a \mu_a$ and $\mu^* = \mu_{a^*} = \max_a \mu_a$. For any suboptimal arm, we will use $\Delta_a := \mu^* - \mu_a$ for the gap of arm a. Finally, the arm pulled by the algorithm in tth round will be denoted by a_t.

Regret: How do we measure the performance of an algorithm? Recall that the goal of the algorithm is to maximize the sum of rewards received in all rounds.

One standard approach is to compare the algorithm performance with the best possible algorithm that knows the distributions D_a for all $a \in [K]$. Note that if means $\mu_1, \ldots, \mu_K$ are known then best strategy to maximize total expected rewards is to always pick the arm a^* to get total expected reward of $T\mu^*$. It makes sense to define the regret of an algorithm after T rounds as

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

where a_t is the arm pulled by the algorithm in the t th round. Thus $R(T)$ is the regret incurred by the algorithm after T rounds of not knowing the means $\mu_1, \mu_2, \ldots, \mu_K$. Note that the regret $R(T)$ is a random variable. We are interested in the expected regret $E[R(T)]$ of the algorithm.

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

The expectation in the above definition is taken over all the randomness, i.e., randomness over the draw of rewards from the distributions $D_1, \ldots, D_K$ and internal randomness of the algorithm (if the algorithm is randomized).

Remark. *1. By the definition only, $R(T)$ and $E[R(T)]$ of an algorithm depends on the problem-instance, i.e., means $\mu_1, \ldots, \mu_K$. Of course, we want an algorithm whose expected regret is small for all instances. In the previous line, 'for all problem instances' is important. For example, consider a stupid algorithm that always pulls arm 1. This algorithm will have regret 0 if the arm 1 has mean 1 and other arms have mean 0. But of course this algorithm will suffer for other instance such as if $\mu_1 = 0$ and $\mu_2 = 1$*

and will have regret of T. So we want an algorithm that performs well on all instances.

2. *If the regret is small then the algorithm's performance is close to best performance when the distributions are known. So we kind of learn the distributions.*

3. *If $r_1, \dots, r_T$ are rewards received by the algorithm in rounds $1, 2, \dots, T$ respectively then one can see that $E[\sum_{i=1}^{T} r_t] = E[\sum_{t=1}^{T} \mu_{a_t}]$. Hence the expected regret of an algorithm is the difference of expected total rewards of best strategy that knows $\mu_1, \dots, \mu_K$ and the expected total rewards of the algorithm.*

Note: In some text books, $R(T)$ is called random-regret or realized regret and $E[R(T)]$ is called Regret. Its just a matter of convention, what names should we use. Often, from the context it will be clear whether we are talking about $R[T]$ or $E[R(T)]$.

Our goal is to design an algorithm whose expected regret $E[R(T)]$ is as small as possible for all problem-instances. Like a general discussion on algorithms, we will ignore constants and only consider Big-O dependence on T. Also, we assume $T \geq K$ as any reasonable algorithm will pull each arm atleast once. Note that since reward in any round is in $[0, 1]$, $R(T) \leq T$ always. We want an algorithms for which $R(T)$ grows sublinear in T, i.e., $\frac{R(T)}{T} \to 0$ as $T \to \infty$. In other words, average regret per round should be 0 when T is large (so we essentially we have learned the unknown distributions). Smaller the regret's dependence on T, faster the rate of convergence to 0 for the average regret per round.

3.1 Explore Then Commit Algorithm

This algorithm is very intuitive and perhaps we will first come up with this algorithm. If we know the means $\mu_1, \dots, \mu_K$ then in each round the best algorithm will pick the arm a^* (recall notation, a^* has highest mean $\mu^* = \max_{a \in A}\{\mu_a\}$) to maximize total expected rewards. This algorithm first try each arm m times and finds the estimate $\hat{\mu}_a$ of the mean μ_a for every arm a. Thereafter, the algorithm will always pick an arm that maximizes the empirical mean $\hat{\mu}_a$ (we will call this arm a'). The value of m is set to $\frac{T^{2/3}(\log T)^{1/3}}{K^{2/3}}$ to optimze the expected regret.

We will now prove our first theorem of this course.

Theorem 4. *For any instance (i.e., for any values of unknowns $\mu_1, \dots, \mu_K$), the expected regret of the algorithm ETC is $O(T^{2/3}(\log T)^{1/3}K^{1/3})$.*

Proof. Let $\epsilon = \sqrt{\frac{5 \ln T}{m}}$. For any arm a, let Bad_a be the event that $|\hat{\mu}_a - \mu_a| \geq \epsilon$. As $\hat{\mu}_a = \frac{\sum_{i=1}^{m} r_{ai}}{m}$ and all rewards are in $[0, 1]$, from Hoffding's inequality, we have

$$\Pr(Bad_a) = \Pr(|\hat{\mu}_a - \mu_a| \geq \epsilon) \leq \frac{2}{e^{2\epsilon^2 m}} = \frac{2}{T^{10}}.$$

Algorithm 1: Explore Then Commit

1. $m = \frac{T^{2/3}(\log T)^{1/3}}{K^{2/3}}$;
2. Try each arm m times. Let for arm a, $r_{a1}, r_{a2}, \ldots, r_{am}$ be the rewards received.;

```
/* Line 2 is exploration/investment phase.  In this phase,
   the algorithm tries each arm so even the worst arms.  But
   we hope that we will be able to find near-optimal arm for
   remaining rounds.                                        */
```

3. For each arm a, set $\hat{\mu}_a$ as the average received reward for the arm a, i.e., we have $\hat{\mu}_a = \frac{\sum_{i \in [m]} r_{ia}}{m}$. Let a' be the arm that maximizes $\hat{\mu}_a$, i.e., $a' = \arg\max_a \{\hat{\mu}_a\}$.;

/* We hope a' is the near-optimal arm, i.e., $\mu^* - \mu_{a'}$ is very small. */

4. From the round $m \cdot K + 1$, always pick the arm a'.

Let $Bad = \cup_a Bad_a$ be the event that for some arm a, $|\hat{\mu}_a - \mu_a| \geq \epsilon$ holds. So $Good = Bad^c$ is the event that, for all arms a, we have $|\hat{\mu}_a - \mu_a| < \epsilon$.

By union bound, we have

$$\Pr(Bad) \leq \sum_a \Pr(Bad_a) \leq K \cdot \frac{2}{T^{10}} \leq \frac{2}{T^9}$$

(as we are assuming $T \geq K$ as any reasonable algorithm will try each arm at least once).

Also,

$$\Pr(Good) = 1 - \Pr(Bad) \geq 1 - \frac{2}{T^9}$$

As the $\Pr(Bad)$ is negligible (we have chosen $\epsilon = \sqrt{\frac{5 \ln T}{m}}$ to ensure that this happens), it will suffice for us to only bound the expeceted regret conditioned on event Good. Following calculation formally show this.

$$\begin{aligned}
E[R(T)] &= \Pr(Bad)E[R(T)|Bad] + \Pr(Good)E[R(T)|Good] \\
&\leq \frac{2}{T^9} \cdot T + 1 \cdot E[R(T)|Good] \\
&\leq \frac{2}{T^8} + \cdot E[R(T)|Good]
\end{aligned}$$

the second inequality comes from $E[R(T)|Bad] \leq T$ as all rewards are in $[0, 1]$.

Bounding $E[R(T)|Good]$**.** We will now assume event *Good*, i.e., for all arms a, we have $|\hat{\mu}_a - \mu_a| < \epsilon$. Recall that $R(T) = \sum_{t=1}^{T} \mu^* - \mu_{a_t}$.

The contribution to regret in the investment phase can be at most mK (assuming worst case contribution of 1 in each round). The contribution to the

regret from $(mK+1)$st round till end is $(T-mK)(\mu^*-\mu_{a'})$ (recall that a' is the arm that maximizes $\hat{\mu}_a$ and is always chosen from $(mK+1)$st round) . We now claim that $(\mu^*-\mu_{a'}) < 2\epsilon$. For now let us assume this claim and bound the regret. We will prove the claim later.

$$\begin{aligned} R(T)|Good &\leq mK + (T-mK)2\epsilon \\ &\leq mK + 2T\epsilon \\ &= mK + 2T\sqrt{\frac{5\ln T}{m}} \end{aligned}$$

As m inreases mK increases and $2T\sqrt{\frac{5\ln T}{m}}$ decreases so the above quantity will be maximized when $mK = 2T\sqrt{\frac{5\ln T}{m}}$, i.e., when $m = (2\sqrt{5})^{2/3}\frac{T^{2/3}(\log T)^{1/3}}{K^{2/3}}$. Substituting this value of m we get $R(T)$ (conditioned on *Good*) equal to $O(T^{2/3}K^{1/3}(\log T)^{1/3})$. Obviously, we also have then
$E[R(T)|Good] = O(T^{2/3}K^{1/3}(\log T)^{1/3})$.

From our earlier calculation, $E[R(T)] = \frac{2}{T^8} + \cdot E[R(T)|Good] = O(T^{2/3}K^{1/3}(\log T)^{1/3})$. Now all it remains to prove the claim that $(\mu^*-\mu_{a'}) < 2\epsilon$. Note that $\hat{\mu}_{a'} \geq \hat{\mu}_{a^*} > \mu^* - \epsilon$ and also $\mu_{a'} > \hat{\mu}_{a'} - \epsilon$. Thus we have $\mu^* - \mu_{a'} < 2\epsilon$.

□

4 Successive Elimination Algorithm

One drawback of ETC algorithm is that it will continue to explore an arm large number of times (m times) even if an arm's reward history might suggest to not pull this arm further. In Successive Elimination algorithm, we discontinue the arm forever once we have belief that the arm is not good. Below is the high level description of this algorithm.

Algorithm 2: Successive Elimination - High Level Description

---

1) Pull every arm once;
2) If there is 'sufficient evidence' that some arm a is not a good arm then remove this arm;

Repeat the above steps over the remaining arms

Now to describe Successive Elimination fully, we just need to specify what 'sufficient evidence' is. For the same, we introduce some notations. For any arm a and round t, let $n_a(t)$ be the number of times the arm a is pulled till the round t. Obviously, we have $\sum_a n_a(t) = t$. Further, let $\hat{\mu}_a(t)$ be the empirical mean of received rewards from the arm a till round t. Formally, let r_{ai} be the reward received from the arm a on the ith pull. Then we have $\hat{\mu}_a(t) = \frac{\sum_{i=1}^{n_a(t)} r_{ai}}{n_a(t)}$. Let $\epsilon_a(t) = \sqrt{\frac{5 \log T}{n_a(t)}}$. Finally, let $UCB_a(t) = \hat{\mu}_a(t) + \epsilon_a(t)$ and $LCB_a(t) = \hat{\mu}_a(t) - \epsilon_a(t)$.

Now we describe the 'sufficent evidence' which Succeesive Elimination employs. Recall the analysis of ETC algorithm. There we define an event *Good* (and show that it holds with high probability) and show that conditioned on *Good*, $\hat{\mu}_a - \epsilon \leq \mu \leq \hat{\mu}_a + \epsilon$ where $\epsilon = \sqrt{\frac{5 \log T}{m}}$. Here also, we will define an event *Good* (and show it will hold with high probability) conditioned on which for all arm a and round t, we will have $LCB_a(t) \leq \mu \leq UCB_a(t)$. Now if at any time t, we have $UCB_a(t) < LCB_{a'}(t)$ for some arms a and a' then we know that $\mu_a < \mu_{a'}$. Hence, it is not a good strategy to pull arm a in any subsequent rounds because we will be better off pulling arm a'. In other words, we can eliminate the arm a for future.

Algorithm 3: Successive Elimination

Activate all the arms;
while $\#$-*rounds* $< T$ **do**
  Pull each active arm once (and receive rewards);
  Deactivate all arms a such that there exits some another arm a' with $UCB_a < LCB_{a'}$;
end

Theorem 5. *For all instances, i.e, for all values of* $\mu_1, \ldots, \mu_K$*,*

$$E[R(T)] = O(\sqrt{KT \log T})$$

Proof. Let *Good* be the following event: for all arms a and for all rounds t, we have $|\hat{\mu}_a(t) - \mu_a| < \epsilon_a(t)$ (that is, $LCB_a(t) \leq \mu_a \leq UCB_a(t)$). We will prove that

$$\Pr(Good) \geq 1 - O(\frac{1}{T^8})$$

For now, let us assume the above. Like ETC, it will suffice to bound $E[R(T)|Good]$.

$$\begin{aligned} E[R(T)] &= \Pr(Bad)E[R(T)|Bad] + \Pr(Good)E[R(T)|Good] \\ &\leq O(\frac{1}{T^8}) \cdot T + 1 \cdot E[R(T)|Good] \\ &\leq O(\frac{1}{T^7}) + \cdot E[R(T)|Good] \end{aligned}$$

the second inequality comes from $E[R(T)|Bad] \leq T$ as all rewards are in $[0, 1]$.

Bounding $E[R(T)|Good]$ *Each calculation in this paragraph is conditioned on Good.* Note that $R(T) = \sum_t(\mu^* - \mu_{a_t}) = \sum_a n_a(T)(\mu^* - \mu_a)$. If we show for each arm a, $(\mu^* - \mu_a) \leq (8\sqrt{\frac{5 \log T}{n_a(T)}})$ then we are done. This is because $R(T) = \sum_a n_a(T)(\mu^* - \mu_a) \leq \sum_a n_a(T)(8\sqrt{\frac{5 \log T}{n_a(T)}}) \leq \sum_a 8\sqrt{5 n_a(T) \log T}$. As $\sqrt{x}$ is a concave function so we have $\frac{\sum_a \sqrt{n_a(T)}}{K} \leq \sqrt{\frac{\sum_a n_a(T)}{K}} = \sqrt{T/K}$. Thus $R(T) \leq 8\sqrt{5KT \log T}$. So it remains to show that for any arm a, we have $(\mu^* - \mu_a) \leq 8\sqrt{\frac{5 \log T}{n_a(T)}}$. For the sake of analysis, we will refer to the ith iteration of while loop as phase i (for any i). Our first easy observation is that the arm a^* with will never be deactivated. Let t be the last round (corresponding to the end of some phase) where the arm a remained active (in other words, arm a was played exactly once after t). Note that $n_a(t) = n_{a^*}(t)$ (because both arms a and a^* are active till t so both of them are played equal number of times, which is the number of phases completed till t). As the arm a is not deactivated at t, we must have $UCB_a(t) \geq LCB_{a^*}(t)$. Further, we have $\epsilon_a(t) = \epsilon_{a^*}(t) = \sqrt{\frac{5 \log T}{n_a(t)}} = \sqrt{\frac{5 \log T}{n_a(T) - 1}}$. It is easy to now see that we have $\mu^* - \mu_a \leq 4\epsilon_a(t) = 4\sqrt{\frac{5 \log T}{n_a(T) - 1}} \leq 8\sqrt{\frac{5 \log T}{n_a(T)}}$.

Bounding $\Pr(Good)$ It remains to show :

$$\Pr(Good) \geq 1 - O(\frac{1}{T^8})$$

Let $r_{a1}, r_{a2}, \ldots, r_{aT}$ be T samples from the distribution D_a. Let $Bad_a(t)$ be the event that $|\frac{r_{a1} + \cdots + r_{at}}{t} - \mu_a| \geq \sqrt{\frac{5 \log T}{t}}$. By Hoffding's inequality, we have

$\Pr(Bad_a(t)) \leq O(1/e^{10\log T}) = O(\frac{1}{T^{10}})$. Let Bad_a be the event that for some $1 \leq t \leq T$, we have $|\frac{r_{a1}+\cdots+r_{at}}{t} - \mu_a| \geq \sqrt{\frac{5\log T}{t}}$. By union bound, $\Pr(Bad_a) \leq T \cdot O(\frac{1}{T^{10}}) = O(\frac{1}{T^9})$ Let $Bad = \cup_a Bad_a$. Again by union bound, $\Pr(Bad) \leq K \cdot O(\frac{1}{T^9}) = O(\frac{1}{T^8})$.

□

The regret in the above theorem is worst-case regret, i.e, for any problem-instance (that is, for any values of K, T and $\mu_1, \ldots, \mu_K$, the expected regret $E[R(T)] \leq O(\sqrt{KT\log T})$. Now we will show another type of bounds on expected regret, which will be instance -dependent bound.

Theorem 6. *The expected regret of Successive Elimination satisfies*

$$E[R(T)] \leq O(\log T) \sum_{a:\triangle_a>0} \frac{1}{\triangle_a}$$

where $\triangle_a = \mu^* - \mu_a$ *is the gap of arm* a.

Proof. We define the events Bad and $Good$ as in the above theorem. Again it will suffice to bound $E[R(T)|Good]$. All calculations now are conditioned on $Good$. We claim that for any suboptimal arm a, we have $n_a(T) \leq 50000\frac{\log T}{\triangle_a^2}$. This implies that $R(T) = \sum_a n_a(T)\triangle_a \leq O(\log T)\sum_{a:\triangle_a>0} \frac{1}{\triangle_a}$. Suppose $n_a(T) > 50000\frac{\log T}{\triangle_a^2}$. Consider the time t when $n_a(t) = 50000\frac{\log T}{\triangle_a^2}$. As the arm a^* is always active we also have $n_{a^*}(t) = 50000\frac{\log T}{\triangle_a^2}$. So now we have $\epsilon_a(t) = \epsilon_{a^*}(t) = \sqrt{\frac{5\log T}{n_a(t)}} = \triangle_a/100$. As we have assumed $Good$, we have $LCB_{a^*}(t) \geq \mu^* - \triangle_a/100$ and $UCB_a(t) \leq \mu_a + \triangle_a/100$. So we have $UCB_a(t) < LCB_{a^*}(t)$ which implies that arm a will be eliminated in round t which contradicts that $n_a(T) > 50000\frac{\log T}{\triangle_a^2}$.

□

UCB Algorithm

Let $n_a(t), \epsilon_a(t), \hat{\mu}_a, UCB_a(t)$ be as defined before (in Successive Elimination algorithm).

The idea of UCB is to add bonus to the empirical mean and then pick an arm that has highest value of empirical mean plus bonus. The bonus is chosen to be $\epsilon_a(t)$ which is equal to $\sqrt{\frac{5\log T}{n_a(t)}}$ (since $\hat{\mu}_a(t) + \epsilon_a(t) = UCB_a(t)$, the algorithm, at any time t, pulls an arm that has highest value of $UCB_a(t)$).

Let us now go into the intuition behind the UCB algorithm in detail. During the initial rounds, the difference between the empirical mean and the actual mean of an arm can be significant. Consequently, selecting an arm solely based on the maximum empirical mean value is not a good strategy. To address this issue, the algorithm incorporates a bonus term. If an arm a is underexplored, the bonus is large, which encourages the algorithm to explore that arm (even if its empirical mean is small at this time). One concern might be whether the bonus term could lead the algorithm to pull bad arms excessively. However, this scenario will not happen because the bonus term diminishes as the number of pulls for the arm increases.

Algorithm 4: UCB Algorithm

```
UCB_a = ∞ for all arms a;
/* Initialization                                   */
while #-rounds < T do
    Pull an arm that has the highest value of UCB_a;
    /* Recall that UCB_a(t) = μ̂_a(t) + ε_a(t)          */
end
```

UCB acheives the same guarantee on expected regret as that of Successive Elimination. The proof is almost same so here we do not give a complete proof.

Theorem 7. *For all instances, i.e, for all values of* $\mu_1, \ldots, \mu_K$*,*

$$E[R(T)] = O(\sqrt{KT\log T})$$

and

$$E[R(T)] = O(\log T) \sum_{a:\triangle_a > 0} \frac{1}{\triangle_a}$$

where $\triangle_a = \mu^* - \mu_a$ *is the gap of arm* a*.*

Proof. The proof is almost same as that of Successive Elimination. To prove $E[R(T)] = O(\sqrt{KT\log T})$, we now show that for any arm a, $\mu^* - \mu_a \leq 2\sqrt{\frac{5\log T}{n_a(T)}}$ (assuming *Good*) (and then the rest of the proof is exactly same). Let t_a be the last round when the arm a was pulled. Thus $n_a(T) = n_a(t)$. Note $\mu_a \geq UCB_a(t) - 2\epsilon_a(t)$ (as we are assuming *good*), $UCB_{a^*} \geq \mu^*$ (as we are assuming

Good) and $UCB_{a^*}(t_a) \leq UCB_a(t_a)$ (as arm a was pulled in round t_a). Hence, $\mu^* - \mu_a \leq 2\sqrt{\frac{5 \log T}{n_a(t_a)}} = 2\sqrt{\frac{5 \log T}{n_a(T)}}$. Now the proof goes the same was as in Successive Elimination algorithm.

The proof of instance dependent bound is also similar. We here prove that (assuming *Good*) for any suboptimal arm a, we have $n_a(T) \leq 50000\frac{\log T}{\triangle_a^2}$ (and then the proof is exactly same). Suppose not. Consider a time t when $n_a(t) = 50000\frac{\log T}{\triangle_a^2}$. We have $\epsilon_a(t) = \triangle_a/100$. Note that for any time $t' > t$, we have $\epsilon_a(t') \leq \epsilon_a(t)$. For any time $t' > t$, we have $UCB_a(t') \leq \mu_a + 2\epsilon_a(t') \leq \mu_a + 2\epsilon_a(t) < \mu^* \leq UCB_{a^*}(t')$. This means that for any $t' > t$ we will have $UCB_{a^*}(t') > UCB_a(t')$ which means that arm a will never be pulled after t. This contradicts that $n_a(t) > 50000\frac{\log T}{\triangle_a^2}$.

$\square$

MOSS algorithm (UCB2)

MOSS is a variant of UCB and it has the expected regret of $O(\sqrt{KT})$ and hence it beats both Successive Elimination and UCB in theory. In the next lecture, we will see that no algorithm can have smaller expected regret than $O(\sqrt{KT})$ and hence MOSS is the best possible algorithm.

MOSS is same as UCB expect how bonus is calculated. Now the bonus is set to $\sqrt{\frac{\max(\log \frac{T}{Kn_a(t)}, 0)}{n_a(t)}}$. Let

$$I_a(t) = \hat{\mu}_a(t) + \sqrt{\frac{\max(\log \frac{T}{Kn_a(t)}, 0)}{n_a(t)}}.$$

At any time t, MOSS pulls an arm that has a maximum value of I_a. One reason MOSS has better expected regret bound than UCB is that the estimation precision when $n_a(t)$ is large is more accurate in MOSS than UCB. In other words, the bonus more quickly goes to 0 in MOSS than UCB as $n_a(t)$ increases.

Algorithm 5: MOSS Algorithm

```
I_a = ∞ for all arms a;
/* Initialization                                              */
while #-rounds < T do
  | Pull an arm that has the highest value of I_a;
end
```

Theorem 8. *The expected regret $E[R(T)]$ of MOSS satisfies*

$$E[R(T)] = O(\sqrt{KT})$$

Proof. We will use a trick that is frequently employed in analysis of randomized algorithms. Instead of sampling from the distribution D_a at the time when the arm a is pulled, we assume that T independent samples from every distribution D_a has already been sampled before the start of the algorithm. For each arm a, let $r_{a1}, \ldots, r_{aT}$ be the T independent samples drawn from the distribution D_a. Now when the algorithm pulls arm a, we provide sample (to the algorithm) from $r_{a1}, \ldots, r_{aT}$. In particular, the sample provided to the algorithm for ith pull of arm a is r_{ai}.

Let us define new notations. Let for any $1 \leq x \leq T$, $\hat{\mu}_{a\,x} = \frac{\sum_{j=1}^{x} r_{aj}}{x}$ be the average of first x rewards (of T samples drawn beforehand).With respect to this new notation, note that $\hat{\mu}_a(t) = \hat{\mu}_{a\,n_a(t)}$. Further, for any $1 \leq x \leq T$, let $I_{a\,x} = \hat{\mu}_{a\,x} + \sqrt{\frac{\max(\log \frac{T}{Kx}, 0)}{x}}$. Again note that $I_a(t) = I_{a\,n_a(t)}$. We will call $I_a(t)$ as the index of arm a after time t.

Let $\delta = \max\{\mu^* - \min_{1 \leq x \leq T} I_{a^*\,x}, 0\}$. Note that δ is a random variable. By definition only, the index of the best arm will never be less than $\mu^* - \delta$, i.e.,

$I_{a^*}(t) \geq \mu^* - \delta$ for all t. We will prove later that $E[\delta] \leq 10\sqrt{\frac{K}{T}}$. It will be helpful to keep this fact in mind.

Let us call an arm a as *Good* if $\triangle_a \leq 5\sqrt{\frac{K}{T}}$ (note that this is different - in previous algorithms, *Good* and *Bad* were events). An arm a is called *Bad* if $\triangle_a > 5\sqrt{\frac{K}{T}}$. Now it will be clear why we have defined *Good* and *Bad* arms in this way. Recall that $R(T) = \sum_a R_a(T)$ where $R_a(T) = n_a(T)\triangle_a$.

$$
\begin{aligned}
R(T) &= \sum_{a:a \text{ is } Good} R_a(T) + \sum_{a:a \text{ is } Bad} R_a(T) \\
&\leq \sum_{a:a \text{ is } Good} n_a(T) 5\sqrt{\frac{K}{T}} + \sum_{a:a \text{ is } Bad} R_a(T) \\
&\leq 5\sqrt{\frac{K}{T}} \sum_{a:a \text{ is } Good} n_a(T) + \sum_{a:a \text{ is } Bad} R_a(T) \\
&\leq 5\sqrt{\frac{K}{T}} \cdot T + \sum_{a:a \text{ is } Bad} R_a(T) \\
&\leq 5\sqrt{KT} + \sum_{a:a \text{ is } Bad} R_a(T)
\end{aligned}
$$

Thus it suffices to show $\sum_{a:a \text{ is } Bad} R_a(T) = O(\sqrt{KT})$. Let us introduce few more notations. For any bad arm a, we define a value k_a (which is a random variable) as follows:

$$
k_a = |\{1 \leq x \leq T | I_{a\,x} > \mu_a + \frac{\triangle_a}{2}\}|
$$

We also define J which is is a random subset of bad arms defined as below:

$$
J = \{a \in [K] | a \text{ is } Bad \text{ and } \triangle_a > 2\delta\}
$$

A very important observation is that for any arm in J, we have $n_a(T) \leq k_a$. This is the main crux of the analysis. As directly showing bounds for $E[n_a(t)]$ is difficult but later we will be able to show bounds for $E[k_a]$ for bad arms.

Now

$$
\begin{aligned}
\sum_{a:a \text{ is } Bad} R_a(T) &= \sum_{a \in J} n_a(T)\triangle_a + \sum_{a \notin J: a \text{ is } Bad} n_a(T)\triangle_a \\
&\leq \sum_{a:a \text{ is } Bad} k_a \triangle_a + 2\delta T
\end{aligned}
$$

Now

$$
\begin{aligned}
E[\sum_{a:a \text{ is } Bad} R_a(T)] &\leq \sum_{a:a \text{ is } Bad} E[k_a]\triangle_a + 2E[\delta]T \\
&\leq \sum_{a:a \text{ is } Bad} E[k_a]\triangle_a + 20\sqrt{KT}
\end{aligned}
$$

as we earlier claimed (without proof) that $E[\delta] \leq 10\sqrt{\frac{K}{T}}$. Thus it suffices to show that $\sum_{a:a \text{ is } Bad} E[k_a]\triangle_a = O(\sqrt{KT})$. Recall that for any event A, we use 1_A for indicator r.v. that takes the value 1 if A happens and 0 otherwise.

For any bad arm a, we have

$$
\begin{aligned}
E[k_a] &= E[|\{1 \leq x \leq T | I_{a\,x} > \mu_a + \frac{\triangle_a}{2}\}|] \\
&= E[\sum_{x=1}^{T} 1_{I_{a\,x} > \mu_a + \frac{\triangle_a}{2}}] \\
&= \sum_{x=1}^{T} E[1_{I_{a\,x} > \mu_a + \frac{\triangle_a}{2}}] \\
&= \sum_{x=1}^{T} \Pr(I_{a\,x} > \mu_a + \frac{\triangle_a}{2}) \\
&= \sum_{x=1}^{T} \Pr(\hat{\mu}_{a\,x} + \sqrt{\frac{\max(\log \frac{T}{Kx}, 0)}{x}} > \mu_a + \frac{\triangle_a}{2}) \\
&= \sum_{x=1}^{T} \Pr(\hat{\mu}_{a\,x} - \mu_a > \frac{\triangle_a}{2} - \sqrt{\frac{\max(\log \frac{T}{Kx}, 0)}{x}}) \\
&\leq \sum_{x=1}^{8\frac{\log \frac{T\triangle_a^2}{K}}{\triangle_a^2}} 1 + \sum_{x=8\frac{\log \frac{T\triangle_a^2}{K}}{\triangle_a^2}}^{T} \Pr(\hat{\mu}_{a\,x} - \mu_a > \frac{\triangle_a}{2} - \sqrt{\frac{\max(\log \frac{T}{Kx}, 0)}{x}}) \\
&= 8\frac{\log \frac{T\triangle_a^2}{K}}{\triangle_a^2} + \sum_{x=\frac{8\log \frac{T\triangle_a^2}{K}}{\triangle_a^2}}^{T} \Pr(\hat{\mu}_{a\,x} - \mu_a > \frac{\triangle_a}{2} - \sqrt{\frac{\max(\log \frac{T}{Kx}, 0)}{x}})
\end{aligned}
$$

As the arm a in the above calculations is bad, we have $\triangle_a > 5\sqrt{\frac{K}{T}}$. This implies that for $x \geq 8\frac{\log \frac{T\triangle_a^2}{K}}{\triangle_a^2}$, we have

$$
\begin{aligned}
\frac{\max(\log \frac{T}{Kx}, 0)}{x} &\leq \frac{\max(\log \frac{T}{K8\frac{\log \frac{T\triangle_a^2}{K}}{\triangle_a^2}}, 0)}{8\frac{\log \frac{T\triangle_a^2}{K}}{\triangle_a^2}} \\
&= \frac{\triangle_a^2}{8} \cdot \frac{\log \frac{T\triangle_a^2}{8K\log(T\triangle_a^2/K)}}{\log(T\triangle_a^2/K)} \\
&\leq \frac{\triangle_a^2}{8}
\end{aligned}
$$

The last inequality holds because $\log(T\triangle_a^2/K) > 1$ as $\triangle_a > 5\sqrt{\frac{K}{T}}$. Therefore

$$\Pr(\hat{\mu}_{a\,x} - \mu_a > \frac{\triangle_a}{2} - \sqrt{\frac{\max(\log\frac{T}{Kx}, 0)}{x}}) \leq \Pr(\hat{\mu}_{a\,x} - \mu_a > \frac{\triangle_a}{2} - \frac{\triangle_a}{2\sqrt{2}})$$
$$\leq 2exp(-2c^2\triangle_a^2 x)$$

where $c = \frac{1}{2}(1 - \frac{1}{\sqrt{2}})$.

Now

$$E[k_a] \leq 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a^2} + \sum_{x=\frac{8\log\frac{T\triangle_a^2}{K}}{\triangle_a^2}}^{T} 2exp(-2c^2\triangle_a^2 x)$$
$$\leq 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a^2} + \sum_{x=\frac{8\log\frac{T\triangle_a^2}{K}}{\triangle_a^2}}^{\infty} 2exp(-2c^2\triangle_a^2 x)$$
$$= 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a^2} + \frac{exp(-2c^2\triangle_a^2 x_0)}{1 - exp(-2c^2\triangle_a^2)}$$

where $x_0 = 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a^2}$ and last equality comes from the geometric series summation.

As $exp(-2c^2\triangle_a^2 x_0) < 1$ we have

$$E[k_a] \leq 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a^2} + \frac{1}{1 - exp(-2c^2\triangle_a^2)}$$

And hence

$$\triangle_a E[k_a] \leq 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a} + \frac{\triangle_a}{1 - exp(-2c^2\triangle_a^2)}$$

Using $1 - e^{-y} \geq y - y^2/2$ we have

$$\triangle_a E[k_a] \leq 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a} + \frac{\triangle_a}{2c^2\triangle_a^2 - 2c^4\triangle_a^4}$$
$$\leq 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a} + \frac{1}{2c^2\triangle_a(1 - c^2\triangle_a^2)}$$
$$\leq 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a} + \frac{1}{2c^2\triangle_a(1 - c^2)}$$
$$\leq 8\frac{\log\frac{T\triangle_a^2}{K}}{\triangle_a} + \frac{1}{2c^2(1 - c^2)}\frac{\sqrt{T}}{5\sqrt{K}}$$

One can check that the maximum value of $f(y) = \frac{\log(Ty^2/K)}{y}$ is $O(\sqrt{\frac{T}{K}})$. Thus

$$\sum_{a:a\,\text{is}\,bad} \triangle_a E[k_a] \leq \sum_{a:a\,\text{is}\,bad} O(\frac{\sqrt{T}}{\sqrt{K}}) = O(\sqrt{KT})$$

All remains to show: $E[\delta] \leq 10\sqrt{\frac{K}{T}}$. $\square$

5 Lower Bounds

We started with ETC and show its expected regret of $O(T^{2/3}K^{1/3}(\log T)^{1/3})$. Then we show that the expected regret of Successive Elimination and UCB is $O(\sqrt{KT \log T})$. Finally, we show that MOSS beats UCB and Successive Elimination and has expected regret of $O(\sqrt{KT})$. What next? Can we have an algorithm that has even better regret than MOSS, i.e., $O(\sqrt{KT})$? What if researchers are unable to find a better regret bound then can we conclude that better bound is not possible? Surely not. It could be lack of understanding or knowledge, preventing us from coming up with a better algorithm.

Now our focus will be lower bounds, i.e., we will be show that no algorithm can have better regret bound than some threshold. In fact we will prove the following theorem

Theorem 9. *No algorithm (deterministic or randomized) can have expected regret of $o(\sqrt{KT})$ for all instances.*

Note that the possibilities of algorithms are infinite. An algorithm can perhaps do anything. Formally, a deterministic algorithm is any function that takes as input K, T and for any $1 \leq t \leq T$ maps a sequence (of pairs consisting of arm and received reward) $\{(a_1, r_1), (a_2, r_2), \ldots, (a_{t-1}, r_{t-1})\}$ to some arm $a_t \in [K]$. Since rewards can take any value in $[0, 1]$, the number of distinct algorithms is infinite. Thus, it is not possible to prove the above theorem by showing that a particular algorithm does not have a better regret bound for all such algorithms (as the number of such algorithms is infinite).

Not surprisingly, we will need new tools to prove such lower bound. We now will study elementary tools from Information theory which will be sufficient to prove the above theorem. In fact we will not directly prove the Theorem 9. First we will consider a toy problem (coin testing problem of exercise) and show a lower bound for this problem. This itself will require new tools from Information Theory. Then after getting some familiarity with these tools, we will prove the Theorem 9.

Testing Coin(ϵ, δ) Problem: It is promised that a given coin is either fair, i.e, $p = 1/2$ or ϵ-biased, i.e., $p = 1/2 + \epsilon$ (algorithm knows ϵ). Give an algorithm A that takes samples from that coin (as input) and has the following guarantee:

- If $p = 1/2$, algorithm must say 'Fair' with at least $1 - \delta$ probability.
- If $p = 1/2 + \epsilon$, the algorithm must say 'Biased' with at least $1 - \delta$ probability.

From Hoffding's inequality, it is easy to show following:

Lemma 1. *There is a deterministic algorithm that solves Testing coin problem and needs only $O(1/\epsilon^2)$ samples for any constant $\delta < 1/2$.*

We will now show the following lower bound:

Lemma 2. *Let $\epsilon \leq 1/3$. Any deterministic algorithm will need at least $\frac{(1-2\delta)^2}{5\epsilon^2}$ samples to solve Testing Coin(ϵ, δ) problem.*

Now the goal is to prove Lemma 2. As said before, we begin with understanding new tools.

Total Variation Distance and KL divergence

Recall that a probability distribution P over a finite domain Ω is a function $P : \Omega \to [0, 1]$ that maps each $\omega \in \Omega$ to a real in $[0, 1]$ satisfying:

1. $P(\omega) \geq 0$ for all $\omega \in \Omega$
2. $\sum_{\omega \in \Omega} P(\omega) = 1$.

Now we will define the notion of distance between two probability distributions. Any notion of distance d must necessarily satisfy these two properies - $d(P, P) = 0$ for any distribution P and $d(P, Q) > 0$ for any two distinct distribution P and Q. There can be many ways to define distance. for instance, we can consider Euclidean distance or ℓ_2 norm: $d(P, Q) = \sum_{\omega}(P(\omega) - Q(\omega))^2$ or we can consider ℓ_1 norm called as total variation distance $d(P, Q) = \frac{1}{2}\sum_{\omega} |P(\omega) - Q(\omega)|$. Each notion of distance has its own merits and limitaions. For us, the notion of total variation distance also known as statistical distance will be important.

Definition 1. *For any two distributions P and Q defined on a finite domain Ω, the total variation distance between P and Q, denoted as $d_{tv}(P, Q)$ is defined as*

$$d_{tv}(P, Q) = \frac{1}{2} \sum_{\omega \in \Omega} |P(\omega) - Q(\omega)|$$

It is easy to see that:

1. $d_{tv}(P, Q) = 0$ if $P = Q$.
2. $d_{tv}(P, Q) > 0$ for any $P \neq Q$.
3. $d_{tv}(P, Q) = d_{tv}(Q, P)$.
4. $d_{tv}(P, Q) \leq d_{tv}(P, R) + d_{tv}(R, Q)$ (triangle inequality)

Now we are going to see one of the most useful and remarkable property of total variation distance.

Lemma 3.

$$d_{tv}(P, Q) = \max_{S \subseteq \Omega} |P(S) - Q(S)| \tag{1}$$

where note that $P(S) = \sum_{\omega \in S} P(\omega)$ and $Q(S) = \sum_{\omega \in S} Q(\omega)$.

Now we define KL divergence between two distributions.

Definition 2. *The KL divergence $KL(P, Q)$ is defined as*

$$KL(P, Q) = \sum_{\omega \in \Omega} P(\omega) \ln \frac{P(\omega)}{Q(\omega)}$$

The KL divergence satisfy:

Lemma 4.
- *$KL(P, P) = 0$*
- *$KL(P, Q) > 0$ if $P \neq Q$*

However, note that $KL(P, Q) \neq KL(Q, P)$ and also it does not satisfy the triangle inequality. That is why it is called divergence not distance. The usefulness of KL divergence comes from the fact that the KL divergence of a product distribution just adds up. Let us see first what is a product distribution and then the utility of KL divergence (I have talked about the product distribution while discussing independent trials).

Definition 3. *Given two distributions P and Q on the domains Ω_1 and Ω_2 respectively, then $P \times Q$ is a distribution on domain $\Omega_1 \times \Omega_2$ such that*

$$(P \times Q)(\omega_1, \omega_2) = P(\omega_1)Q(\omega_b) \quad \forall a \in \Omega_1, b \in \Omega_2$$

In general, if P_i is a distribution on Ω_i then the product distribution $P_1 \times P_2 \times \cdots \times P_m$ is a distribution on domain $\Omega_1 \times \cdots \times \Omega_m$ such that

$$(P_1 \times P_2 \times \cdots P_m)(\omega_1, \omega_2, \ldots \omega_m) = P_1(\omega_1)P_2(\omega_2) \cdots P_m(\omega_m)$$

We will use, for any distribution P, $P^{\otimes 2}$ for $P \times P$. In general, we have $P^{\otimes m} = P \times \cdots \times P$ (m-times).

The product distribution corresponds to independent trials. *Drawing m independent samples from a distribution P is exactly same as dwawing one sample from $P^{\otimes m}$.* Now we will see why KL divergence is extremely useful.

Lemma 5. *If for all $i \in [m]$, P_i and Q_i are distributions on Ω_i then*

$$KL(P_1 \times P_2 \times \cdots \times P_m, Q_1 \times Q_2 \times \cdots \times Q_m) = KL(P_1, Q_1) + KL(P_2, Q_2) + \cdots + KL(P_m, Q_m)$$

Corollary 1.

$$KL(P^{\otimes m}, Q^{\otimes m}) = m \cdot KL(P, Q)$$

The above property is not true for total variation distance. We do not have $d_{tv}(P^{\otimes m}, Q^{\otimes m}) = m\, d_{tv}(P, Q)$. In fact, other than $d_{tv}(P^{\otimes m}, Q^{\otimes m}) \leq m\, d_{tv}(P, Q)$, we do not know of any other inequality between these two.

We have now seen both total variation distance and KL divergence, each having its own merits and limitations. Now we will see that these two are connected by an inequality, which forms the basis of powerful application.

Lemma 6 (Pinsker inequality)**.** *For any two probability distributions P and Q, we have*

$$d_{tv}(P, Q) \leq \sqrt{\frac{1}{2} KL(P, Q)}$$

Following is also useful.

Lemma 7 (The Bretagnolle–Huber bound)**.**

$$d_{tv}(P, Q) \leq \sqrt{1 - e^{-KL(P,Q)}} \leq 1 - \frac{1}{2} e^{-KL(P,Q)}$$

Now we have all the necessary tool to prove Lemma 2. First we will prove the lemma for deterministic algorithms.

Proof of Lemma 2 Suppose there is a deterministic algorithm A that solves the Testing Coin problem and uses m independent coin tosses. Note that algorithm A is just a function $A : \{H, T\}^m \to \{\text{Fair}, \text{Biased}\}$ that maps a sequence of head and tails of length m to Fair or Biased. We partition $\{H, T\}^m$ into two subsets as follows: $A_f = \{y \in \{0,1\}^m : A(y) = \text{Fair}\}$ and $A_b = \{y \in \{0,1\}^m : A(y) = \text{Biased}\}$, i.e., A_f consists of all those sequence which gets mapped to Fair and A_b consists of all those sequences that gets mapped to Biased.

Let $p = \Pr(Head)$. Let $(\frac{1}{2})^{\otimes m}$ and $(\frac{1}{2} + \epsilon)^{\otimes m}$ be the product distributions corresponding to m independent coin tosses with $p = 1/2$ and $p = 1/2 + \epsilon$ respectively. For convenience, let $D_f = (\frac{1}{2})^{\otimes m}$ and $D_b = (\frac{1}{2} + \epsilon)^{\otimes m}$. We will bound tv distance between $D_f = (\frac{1}{2})^{\otimes m}$ and $D_b = (\frac{1}{2} + \epsilon)^{\otimes m}$ in two different ways.

Note that $D_f(A_f) = \sum_{\omega \in A_f} D_f(\omega) \geq 1 - \delta$ and $D_b(A_f) = \sum_{\omega \in A_f} D_b(\omega) \leq \delta$ (because the algorithm A solves the Testing Coin(ϵ,δ) problem). Thus by Lemma 3,

$$d_{tv}(D_f, D_b) \geq D_f(A_f) - D_b(A_f) \geq 1 - 2\delta.$$

Now let us bound tv distance using Pinsker inequality. As said earlier, calculation of KL divergence between D_f and D_b is easy as they both are product distributions.

$$KL((\frac{1}{2})^{\otimes m}, (\frac{1}{2} + \epsilon)^{\otimes m}) = m \cdot KL(1/2, 1/2 + \epsilon)$$

where $KL(1/2, 1/2+\epsilon)$ is the KL divergence between the distributions $(1/2, 1/2)$ and $(1/2 + \epsilon, 1/2 - \epsilon)$.

$$\begin{aligned}
KL(1/2, 1/2 + \epsilon) &= \frac{1}{2} \ln \frac{1}{2(\frac{1}{2} + \epsilon)} + \frac{1}{2} \ln \frac{1}{2(\frac{1}{2} - \epsilon)} \\
&= \frac{1}{2} \ln \frac{1}{4(1/4 - \epsilon^2)} = \frac{1}{2} \ln \frac{1}{(1 - 4\epsilon^2)} \\
&= \frac{1}{2}(4\epsilon^2 + \frac{(4\epsilon^2)^2}{2} + \frac{(4\epsilon^2)^3}{3} + \dots) = 2\epsilon^2(1 + 2\epsilon^2 + 16\epsilon^4/3 + \cdots) \leq 10\epsilon^2
\end{aligned}$$

for $\epsilon \leq 1/3$ (probably 10 can be replaced by a smaller number but as we are not optimizing constants I put here bigger number so that the inequality is easy to derive).

Now we will use Pinsker inequality to bound the tv distance.

$$\begin{aligned}
d_{tv}((\frac{1}{2})^{\otimes m}, (\frac{1}{2} + \epsilon)^{\otimes m}) &\leq \sqrt{\frac{1}{2} KL((\frac{1}{2})^{\otimes m}, (\frac{1}{2} + \epsilon)^{\otimes m})} \\
&\leq \sqrt{5m}\epsilon
\end{aligned}$$

Now we have

$$1 - 2\delta \leq d_{tv}(D_f, D_b) \leq \sqrt{\frac{1}{2} KL((\frac{1}{2})^{\otimes m}, (\frac{1}{2} + \epsilon)^{\otimes m})} \leq \sqrt{5m}\epsilon$$

which implies that $m \geq \frac{(1-2\delta)^2}{5\epsilon^2}$.

Lower Bound for randomized algorithms

Lemma 8. *Let $\epsilon \leq 1/3$. The expected number of coin tosses by any randomized algorithm that solves Testing Coin(ϵ, δ) problem is at least $\frac{(1-6\delta)^2}{15\epsilon^2}$.*

In the class, I discussed that any randomized algorithm R is just a distribution over some deterministic algorithms, i.e., for any randomized algorithm R, there exits deterministic algorithms $A_1, \ldots, A_s$ (for some finite s) and a distribution $(q_1, \ldots, q_s)$ over these deterministic algorithms such that R picks a A_i ($i \in [s]$) (with probability q_i) and then run the deterministic algorithm A_i on the given input.

Proof of Lemma 8 Let R be a randomized algorithm that solves the Testing Coin (ϵ, δ) using at most $m = \frac{(1-6\delta)^2}{15\epsilon^2}$ expected number of coin tosses. Let $A_1, \ldots, A_s$ be deterministic algorithms each requiring at most $m_1, m_2, \ldots, m_s$ coin tosses and let R picks each A_i with probability q_i. Now consider a table with two rows and s columns — one row corresponding to $p = 1/2$ (called row 1) and one corresponding to $p = 1/2 + \epsilon$ (called row 2), and s columns corresponding to each deterministic algorithms $A_1, \ldots, A_s$.

We put ✓ in the cell corresponding to row 1 and column A_j if $\Pr(A_j \text{ returns 'Biased' when } p = 1/2) \leq 3\delta$. Otherwise we put $\times$, i.e, if $\Pr(A_j \text{ returns 'Biased' when } p = 1/2) > 3\delta$. For row 2, we put ✓ in the cell corresponding to row 2 and column A_j if $\Pr(A_j \text{ returns 'Fair' when } p = 1/2 + \epsilon) \leq 3\delta$. Otherwise we put $\times$, i.e, if $\Pr(A_j \text{ returns 'Fair' when } p = 1/2 + \epsilon) > 3\delta$. By $cell(i, A_j)$ ($i \in \{1, 2\}, j \in [s]$), we mean the symbol (✓ or $\times$) in the cell corresponding to row i and column A_j in the above table.

Now,

$$\begin{aligned}
\delta &\geq \Pr(R \text{ returns 'Biased' when } p = 1/2) \quad (\text{because } R \text{ solves Testing Coin}(\epsilon, \delta)) \\
&= \sum q_j \Pr(A_j \text{ returns 'biased' when } p = 1/2) \\
&\geq \sum_{j:cell(1,A_j)=\times} q_j 3\delta \\
&\implies \sum_{j:cell(1,A_j)=\times} q_j \leq 1/3
\end{aligned}$$

Similarly,

$$\sum_{j:cell(2,A_j)=\times} q_j \leq 1/3$$

This means that $\sum_{j:cell(1,A_j)=\checkmark \text{ and } cell(2,A_j)=\checkmark} q_j \geq 1-(1/3+1/3) \geq 1/3$.That is, R puts at least $1/3$ probability on those A_j's for which both $\Pr(A_i \text{ returns 'Biased' when } p = 1/2) \leq 3\delta$ and $\Pr(A_i \text{ returns 'Fair' when } p = 1/2 + \epsilon) \leq 3\delta$ are true. Let $J = \{j : cell(1, A_j) = \checkmark \text{ and } cell(2, A_j) = \checkmark\}$. Note that for any $j \in J$, A_j solves Testing Coin (ϵ, δ) problem and is a deterministic algorithm. By Lemma

2, $m_j \geq \frac{(1-2\delta)^2}{5\epsilon^2}$ (recall m_j is the number of coin tosses by A_j). Therefore, the expected number of coin tosses by R, is $\sum_j q_j m_j \geq \frac{1}{3} \cdot \frac{(1-2\delta)^2}{5\epsilon^2} = \frac{(1-2\delta)^2}{15\epsilon^2}$.

Remark. *I presented the Lemma 8 and its proof in the class in a slightly different way. The Lemma 8 is more stronger than the one I stated in the class. There, I assumed that the running time (or the number of samples) of R is a deterministic so each $A_1, \ldots, A_s$ uses fixed number of samples. As the above proof shows, we can show lower bound on the expected number of samples (by same proof).*

Lower Bounds for Bandits

We start with showing lower bounds when there are only two arms. The best upper bound in this case is $O(\sqrt{T})$. We show a matching lower bound.

Theorem 10. *No algorithm can achieve an expected regret of $O(T^{0.5-\alpha})$ for any constant $\alpha > 0$ for all instances.*

Proof. Let there exists an algorithm B that has an expected regret of $O(T^{0.5-\alpha})$ for some $\alpha > 0$ for all instances. We will show that, using B, we can get a randomized algorithm R that solves Testing Coin$(\epsilon, 1/50)$ in $o(1/\epsilon^2)$ coin tosses contradicting Lemma 8.

Let $\epsilon < 1/3$ be the input of Testing Coin(ϵ, δ) problem. Consider a following bandit instance:

- $K = 2$ (there are two arms)
- Arm 1's reward takes value either 0 or 1. The probability of reward being 1 is the probability that the coin (of Testing Coin$(\epsilon, 1/50)$) outputs a Head.
- Arm 2's reward distribution is also a Bernoulli with mean reward of $\frac{1}{2} + \frac{\epsilon}{2}$, i.e., reward takes either 0 or 1 and the probability of reward being one is $\frac{1}{2} + \frac{\epsilon}{2}$.
- $T = (200/\epsilon)^{\frac{1}{0.5+\alpha}}$

Note in the above that $T = o(1/\epsilon^2)$. Now we describe a randomized algorithm R. The algorithm R runs B on the above bandit instance (recall that we assumed there exists an algorithm B that has an expected regret of $O(T^{0.5-\alpha})$ for some $\alpha > 0$ for all bandit instances). Whenever B pulls arm 2, the algorithm R generates a sample from the arm 2's reward distribution (Bernoulli with mean reward of $\frac{1}{2} + \frac{\epsilon}{2}$) and give it to the bandit algorithm B (these are internal randomness of the randomized algorithm R). Whenever B pulls arm 1, R uses coin toss of the input coin (of Testing Coin (ϵ, δ)) to provide the reward to B — if the coin toss outputs Head, it provides a reward of 1 to B and if the result of coin toss is Tail, R provides a reward of 0 to B. Finally, R returns 'Fair' if B (after T rounds) pulls the arm 2 at least $T/2$ times and otherwise return 'Biased' (i.e., if B pulls arm 1 more than $T/2$ times).

Note that R uses at most $T = o(1/\epsilon)^2$ coin tosses. Now we prove correctness of the algorithm R — if the coin is fair, i.e., $p = \Pr(Head) = 1/2$ then B pulls the arm 2 more than $T/2$ times (and hence outputs 'Fair') with at least $49/50$ probability and if the coin is biased, i.e., $p = 1/2+\epsilon$ then B pulls the arm 1 more than $T/2$ times (and hence outputs 'Biased') with at least $49/50$ probability. In other words, B will always pull the best arm $> T/2$ times with at least $49/50$ probability. To see this, note that in both cases $E[R(T)] = \frac{\epsilon}{2}E[N_s]$ where N_s is the number of times B pulls the suboptimal arm (which is arm 1 when $p = 1/2$ and arm 2 when $p = 1/2 + \epsilon$). Now we have

$$E[R(T)] = \frac{\epsilon}{2}E[N_s] \leq T^{\frac{1}{2}-\alpha} \implies E[N_s] \leq \frac{T^{1/2+\alpha}}{100}T^{\frac{1}{2}-\alpha} = T/100$$

By Markov's inequality, we have $\Pr(N_s \geq T/2) \leq 1/50$. Thus, R solves Testing Coin $(\epsilon, 1/50)$ problem in $T = o(1/\epsilon)^2$ coin tosses which contradicts Lemma 8.

□