L12 - Week 6
Introduction to Multi-armed Bandits

CS 295 Optimization for Machine Learning

Ioannis Panageas

The framework

Setting. *We are given K arms and time window T (known). At each time step* $t = 1 ... T$.

- *Player chooses arm* a_t.
- *Observes reward* $r_t \in [0, 1]$ *for the chosen arm.*

The framework

Setting. *We are given K arms and time window T (known). At each time step $t = 1...T$.*

- *Player chooses arm a_t.*
- *Observes reward $r_t \in [0, 1]$ for the chosen arm.*

- The algorithm observes only the reward for the selected action, and nothing else.
- The reward for each action is IID. For each arm $a \in [K]$, there is a distribution D_a over reals, called the reward distribution (unknown). Every time this action is chosen, the reward is sampled independently from this distribution.

The framework

Setting. *We are given K arms and time window T (known). At each time step $t = 1 \ldots T$.*

- *Player chooses arm a_t.*
- *Observes reward $r_t \in [0,1]$ for the chosen arm.*

- The algorithm observes only the reward for the selected action, and nothing else.
- The reward for each action is IID. For each arm $a \in [K]$, there is a distribution D_a over reals, called the reward distribution (unknown). Every time this action is chosen, the reward is sampled independently from this distribution.

Goal: Minimize the regret

$$\boxed{R(T) = \mu^* T - \sum_{t=1}^{T} \mu(a_t) \text{ or } \mathbb{E}[R(T)].}$$

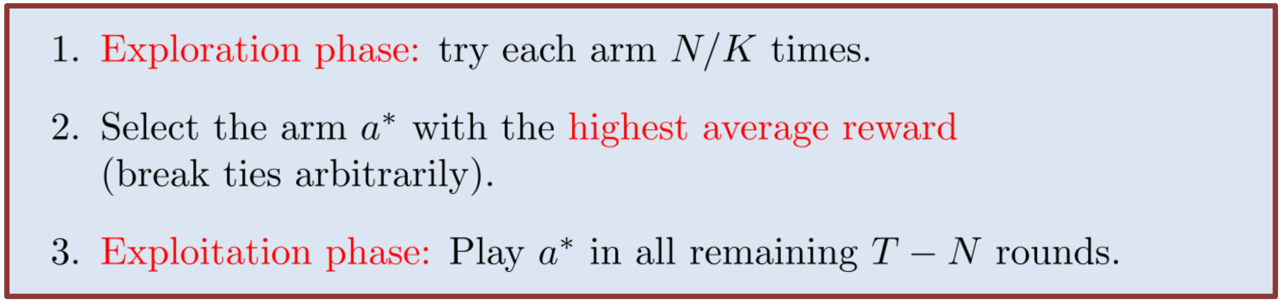
Explore-First

Maybe the most natural approach is to estimate first the expected rewards for all arms and then use the maximum.

Explore-First

Maybe the most natural approach is to estimate first the expected rewards for all arms and then use the maximum.

Definition (Explore-first). *Consider the following algorithm:*



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

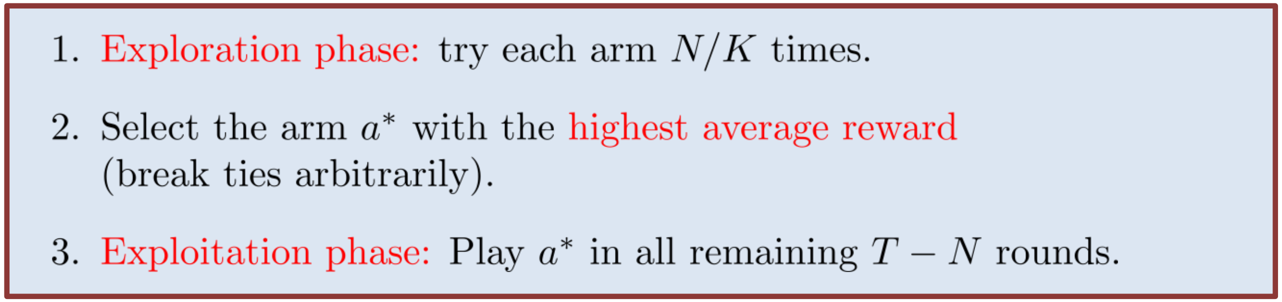
Remarks:

- N will be chosen later as a function of T, K.

Explore-First

Maybe the most natural approach is to estimate first the expected rewards for all arms and then use the maximum.

Definition (Explore-first). *Consider the following algorithm:*



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

Remarks:

- N will be chosen later as a function of T, K.

Let's analyze the regret for Explore-first algorithm!

Analysis of Explore-First

Remark (Hoeffding Inequality)**.** *Let $\hat{\mu}(a)$ be the empirical (or average) reward for action a after exploration phase. It holds*

$$\Pr\left[|\hat{\mu}(a) - \mu(a)| \leq \sqrt{\frac{2K\log T}{N}}\right] \geq 1 - \frac{1}{T^4}.$$

$$\Pr[|\hat{\mu}(a) - \mu(a)| > \epsilon] \leq 2e^{-2\frac{N}{K}\epsilon^2}.$$

Analysis of Explore-First

Remark (Hoeffding Inequality)**.** *Let $\hat{\mu}(a)$ be the empirical (or average) reward for action a after exploration phase. It holds*

$$\Pr\left[|\hat{\mu}(a) - \mu(a)| \leq \sqrt{\frac{2K\log T}{N}}\right] \geq 1 - \frac{1}{T^4}.$$

$$\Pr[|\hat{\mu}(a) - \mu(a)| > \epsilon] \leq 2e^{-2\frac{N}{K}\epsilon^2}.$$

Let us condition on the "clean" event that the above holds for all arms. By union bound the probability of the "bad" event is at most

$$\frac{K}{T^4} \leq \frac{1}{T^3},$$

hence the "clean" event happens with probability at least $1 - \frac{1}{T^3}$.

Analysis of Explore-First

Let a_{best} be the arm with maximum mean reward. Suppose the algorithm chose $a^* \neq a_{best}$. What does this mean?

Analysis of Explore-First

Let a_{best} be the arm with maximum mean reward. Suppose the algorithm chose $a^* \neq a_{best}$. What does this mean?

It means that

$$\hat{\mu}(a^*) \geq \hat{\mu}(a_{best}).$$

Analysis of Explore-First

Let a_{best} be the arm with maximum mean reward. Suppose the algorithm chose $a^* \neq a_{best}$. What does this mean?

It means that

$$\hat{\mu}(a^*) \geq \hat{\mu}(a_{best}).$$

But since we condition on "clean event"

$$\mu(a^*) + \sqrt{\frac{2K \log T}{N}} \geq \hat{\mu}(a^*) \geq \hat{\mu}(a_{best}) \text{ and}$$

$$\hat{\mu}(a_{best}) \geq \mu(a_{best}) - \sqrt{\frac{2K \log T}{N}}.$$

Analysis of Explore-First

Let a_{best} be the arm with maximum mean reward. Suppose the algorithm chose $a^* \neq a_{best}$. What does this mean?

It means that

$$\hat{\mu}(a^*) \geq \hat{\mu}(a_{best}).$$

But since we condition on "clean event"

$$\mu(a^*) + \sqrt{\frac{2K \log T}{N}} \geq \hat{\mu}(a^*) \geq \hat{\mu}(a_{best}) \text{ and}$$

$$\hat{\mu}(a_{best}) \geq \mu(a_{best}) - \sqrt{\frac{2K \log T}{N}}.$$

Hence $\mu(a^*) \geq \mu(a_{best}) - 2\sqrt{\frac{2K \log T}{N}}$.

Analysis of Explore-First

We compute a bound on the regret (conditioned on clean event):

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

Analysis of Explore-First

We compute a bound on the regret (conditioned on clean event):

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

We set $N = 2T^{2/3}(K \log T)^{1/3}$ and we have

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

Analysis of Explore-First

Using law of total expectation we have

$$\begin{aligned}\mathbb{E}[R(T)] &= \mathbb{E}[R(T)|\text{clean}]\Pr[\text{clean}] + \mathbb{E}[R(T)|\text{bad}]\Pr[\text{bad}] \\ &\leq 4(K\log T)^{1/3}T^{2/3} + T \times \frac{1}{T^3} = O((K\log T)^{1/3}T^{2/3}).\end{aligned}$$

Analysis of Explore-First

Using law of total expectation we have

$$\begin{aligned}\mathbb{E}[R(T)] &= \mathbb{E}[R(T)|\text{clean}]\Pr[\text{clean}] + \mathbb{E}[R(T)|\text{bad}]\Pr[\text{bad}] \\ &\leq 4(K\log T)^{1/3}T^{2/3} + T \times \frac{1}{T^3} = O((K\log T)^{1/3}T^{2/3}).\end{aligned}$$

Namely, we showed:

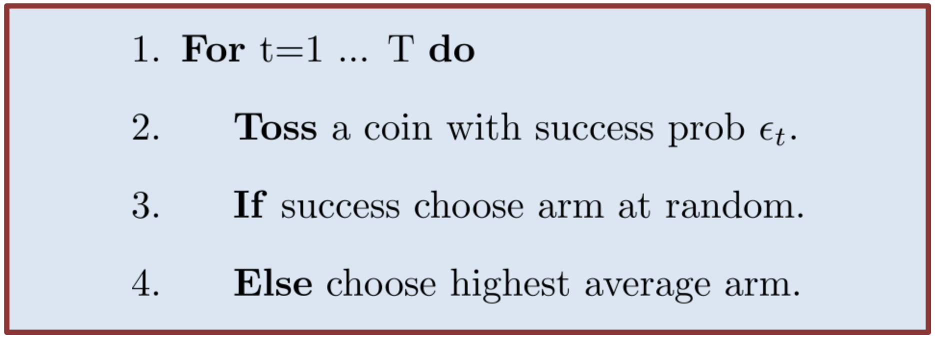
Theorem (Regret). *Explore-first algorithm achieves regret*

$$O((K\log T)^{1/3}T^{2/3}),$$

where K is the number of arms.

Epsilon-Greedy

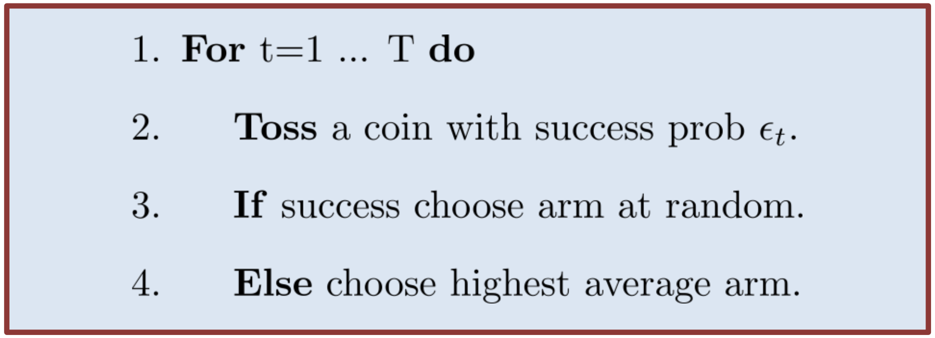
Definition (ϵ-greedy)**.** *Consider the following algorithm:*



```
For t=1 ... T do
    Toss a coin with success prob $\epsilon_t$.
    If success choose arm at random.
    Else choose highest average arm.
```

Epsilon-Greedy

Definition (ϵ-greedy). *Consider the following algorithm:*



```
For t=1 ... T do
    Toss a coin with success prob $\epsilon_t$.
    If success choose arm at random.
    Else choose highest average arm.
```

Theorem (Regret). *ϵ-greedy algorithm achieves regret*

$$\mathbb{E}[R(t)] \text{ to be } O((K \log t)^{1/3} t^{2/3}),$$

where K is the number of arms and $\epsilon_t \sim t^{-1/3}(K \log t)^{1/3}$.

Remarks:

- Same regret as before but for all rounds!

Epsilon-Greedy

Definition (ϵ-greedy). *Consider the following algorithm:*

```
For t=1 ... T do
    Toss a coin with success prob $\epsilon_t$.
    If success choose arm at random.
    Else choose highest average arm.
```

Theorem (Regret). *ϵ-greedy algorithm achieves regret*

$$\mathbb{E}[R(t)] \text{ to be } O((K \log t)^{1/3} t^{2/3}),$$

where K is the number of arms and $\epsilon_t \sim t^{-1/3}(K \log t)^{1/3}$.

Can we do better? Yes, adaptive exploration!

Remarks:

- Same regret as before but for all rounds!

Upper Confidence Bounds

One natural idea (suppose we have two arms): Alternate them until we find that one arm is much better than the other, at which time we abandon the inferior one.

Upper Confidence Bounds

One natural idea (suppose we have two arms): Alternate them until we find that one arm is much better than the other, at which time we abandon the inferior one.

How to define ”one arm is much better” exactly?

Upper Confidence Bounds

One natural idea (suppose we have two arms): Alternate them until we find that one arm is much better than the other, at which time we abandon the inferior one.

How to define ”one arm is much better” exactly?

Recall (Hoeffding). *Let $n_t(a)$ be the number of samples from arm a in round $1, ..., t$, $\hat{\mu}_t(a)$ be the average reward of arm a so far. Hoeffding Inequality suggests*

$$\Pr[|\hat{\mu}_t(a) - \mu(a)| \leq r_t(a)] \geq 1 - \frac{2}{T^4},$$

where $r_t(a) = \sqrt{\frac{2 \log T}{n_t(a)}}$, and $r_t(a)$ is called the confidence radius.

Upper Confidence Bounds

One natural idea (suppose we have two arms): Alternate them until we find that one arm is much better than the other, at which time we abandon the inferior one.

How to define "one arm is much better" exactly?

Recall (Hoeffding). *Let $n_t(a)$ be the number of samples from arm a in round $1, \ldots, t$, $\hat{\mu}_t(a)$ be the average reward of arm a so far. Hoeffding Inequality suggests*

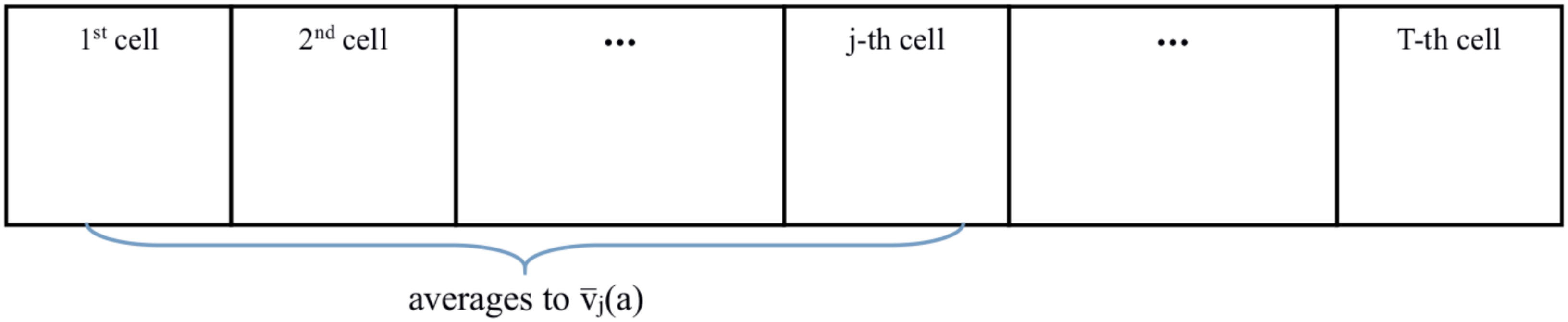
$$\Pr[|\hat{\mu}_t(a) - \mu(a)| \leq r_t(a)] \geq 1 - \frac{2}{T^4},$$

where $r_t(a) = \sqrt{\frac{2\log T}{n_t(a)}}$, and $r_t(a)$ is called the confidence radius.

However $n_t(a)$ should not be fixed (r.v)… Samples Are not independent anymore!

Upper Confidence Bounds

For each arm a, imagine there is a reward tape $1 \times T$ table with each cell independently sampled from D_a. The j-th time a given arm a is chosen by the algorithm, its reward is taken from the j-th cell in this arm's tape.



1st cell

2nd cell

•••

j-th cell

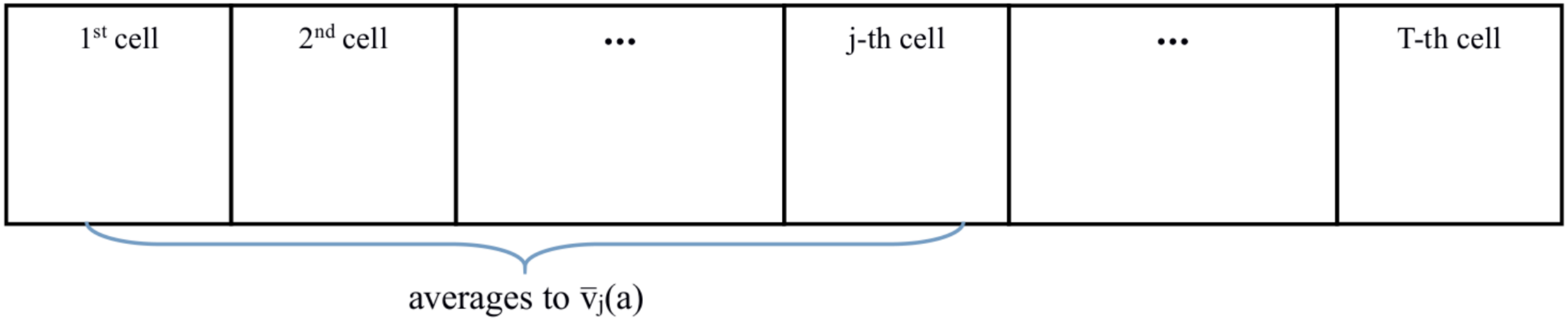
•••

T-th cell

averages to $\bar{v}_j(a)$

Upper Confidence Bounds

For each arm a, imagine there is a reward tape $1 \times T$ table with each cell independently sampled from D_a. The j-th time a given arm a is chosen by the algorithm, its reward is taken from the j-th cell in this arm's tape.



1st cell

2nd cell

…

j-th cell

…

T-th cell

averages to $\bar{v}_j(a)$

Now we can use Hoeffding Inequality hence for all j

$$\Pr[|\hat{v}_j(a) - \mu(a)| \leq r_j(a)] \geq 1 - \frac{2}{T^4},$$

therefore by union bound on j and arms we get

$$\Pr[\forall j, a \; |\hat{v}_j(a) - \mu(a)| \geq r_j(a)] \geq 1 - \frac{1}{T^2},$$

Upper Confidence Bounds

Definition (Confidence bounds)**.** *We define upper/lower confidence bounds for every arm a and round t*

$$UCB_t(a) = \hat{\mu}_t(a) + r_t(a), \;\; LCB_t(a) = \hat{\mu}_t(a) - r_t(a).$$

Upper Confidence Bounds

Definition (Confidence bounds)**.** *We define upper/lower confidence bounds for every arm a and round t*

$$UCB_t(a) = \hat{\mu}_t(a) + r_t(a), \;\; LCB_t(a) = \hat{\mu}_t(a) - r_t(a).$$

Definition (UCB Elimination)**.** *Consider the following algorithm:*

1. Alternate two arms a, a' until $UCB_t(a) < LCB_t(a')$.
2. Abandon arm a, and use arm a'. forever since.

Upper Confidence Bounds

Definition (Confidence bounds). *We define upper/lower confidence bounds for every arm a and round t*

$$UCB_t(a) = \hat{\mu}_t(a) + r_t(a), \;\; LCB_t(a) = \hat{\mu}_t(a) - r_t(a).$$

Definition (UCB Elimination). *Consider the following algorithm:*

1. Alternate two arms a, a' until $UCB_t(a) < LCB_t(a')$.
2. Abandon arm a, and use arm a'. forever since.

Theorem (Regret). *UCB Elimination algorithm achieves regret*

$$\mathbb{E}[R(T)] \text{ to be } O(\sqrt{T \log T}).$$

Upper Confidence Bounds

Definition (Confidence bounds). *We define upper/lower confidence bounds for every arm a and round t*

$$UCB_t(a) = \hat{\mu}_t(a) + r_t(a), \;\; LCB_t(a) = \hat{\mu}_t(a) - r_t(a).$$

Definition (UCB Elimination). *Consider the following algorithm:*

1. Alternate two arms a, a' until $UCB_t(a) < LCB_t(a')$.
2. Abandon arm a, and use arm a'. forever since.

Theorem (Regret). *UCB Elimination algorithm achieves regret*

$$\mathbb{E}[R(T)] \textit{ to be } O(\sqrt{T \log T}).$$

Much better than before!

Analysis of UCB Elimination

Let us define the "clean" event (we condition on that)

$$\mathcal{E} = \{\forall j, a \, |\hat{\mu}_j(a) - \mu(a)| \leq r_j(a)\}.$$

Analysis of UCB Elimination

Let us define the "clean" event (we condition on that)

$$\mathcal{E} = \{\forall j, a \; |\hat{\mu}_j(a) - \mu(a)| \leq r_j(a)\}.$$

Observe that the disqualified arm cannot be the best arm. How long did it take to disqualify it?

Let τ be the last round when we did not invoke the stopping rule, namely when the confidence intervals of the two arms still overlap. It holds that

$$\boxed{|\mu(a) - \mu(a')| \leq 2(r_\tau(a) + r_\tau(a'))}$$

Analysis of UCB Elimination

Let us define the "clean" event (we condition on that)

$$\mathcal{E} = \{\forall j, a \ |\hat{\mu}_j(a) - \mu(a)| \leq r_j(a)\}.$$

Observe that the disqualified arm cannot be the best arm. How long did it take to disqualify it?

Let τ be the last round when we did not invoke the stopping rule, namely when the confidence intervals of the two arms still overlap. It holds that

$$\boxed{|\mu(a) - \mu(a')| \leq 2(r_\tau(a) + r_\tau(a'))}$$

Moreover because we alternated we have $n_\tau(a) = n_\tau(a') = \frac{\tau}{2}$ hence

$$\boxed{r_\tau(a) \text{ and } r_\tau(a') \text{ are } O\left(\sqrt{\frac{\log T}{\tau}}\right).}$$

Analysis of UCB Elimination

Using law of total expectation we have

$$\mathbb{E}[R(T)] = \mathbb{E}[R(T)|\mathcal{E}]\Pr[\mathcal{E}] + \mathbb{E}[R(T)|\not\mathcal{E}]\Pr[\not\mathcal{E}]$$

Analysis of UCB Elimination

Using law of total expectation we have

$$\mathbb{E}[R(T)] = \mathbb{E}[R(T)|\mathcal{E}]\Pr[\mathcal{E}] + \mathbb{E}[R(T)|\not\mathcal{E}]\Pr[\not\mathcal{E}]$$
$$\leq \Delta \times \tau + T \times O\left(\frac{1}{T^2}\right).$$

The above gives $O(\sqrt{T \log T})$.

More than two arms

Definition (UCB Elimination)**.** *Consider the following algorithm:*

1. Initially all arms are set "active";
2. Try all active arms once.
3. Deactivate all arms a s.t. there exists an arm a' with $UCB_t(a) < LCB_t(a')$
4. Repeat until there is one arm left.

More than two arms

Definition (UCB Elimination). *Consider the following algorithm:*

1. Initially all arms are set "active";
2. Try all active arms once.
3. Deactivate all arms a s.t. there exists an arm a' with $UCB_t(a) < LCB_t(a')$
4. Repeat until there is one arm left.

Theorem (Regret). *UCB Elimination algorithm achieves regret*

$$\mathbb{E}[R(T)] \text{ to be } O(\sqrt{KT\log T}).$$

Remarks:

- The proof is almost the same as before. Try to prove it alone.

Conclusion

- Introduction to Multi-armed bandits.
  - Explore-first.
  - Epsilon-greedy
  - UCB Elimination

- Next lecture we will talk more about Exploration-Exploitation tradeoff.