The Multi-Arm Bandit Framework

A. LAZARIC (SequeL Team @INRIA-Lille)
ENS Cachan - Master 2 MVA
In This Lecture

![Map of Lille with location markers A and B]
In This Lecture

**Question:** which route should we take?
Question: which route should we take?

Problem: each day we obtain a limited feedback: traveling time of the chosen route
In This Lecture

**Question**: which route should we take?

**Problem**: each day we obtain a *limited feedback*: traveling time of the *chosen route*

**Results**: if we do not repeatedly try different options we cannot learn.
In This Lecture

**Question**: which route should we take?

**Problem**: each day we obtain a *limited feedback*: traveling time of the *chosen route*

**Results**: if we do not repeatedly try different options we cannot learn.

**Solution**: trade off between *optimization* and *learning*.
Outline

Mathematical Tools

The General Multi-arm Bandit Problem

The Stochastic Multi-arm Bandit Problem

The Non-Stochastic Multi-arm Bandit Problem

Connections to Game Theory

Other Stochastic Multi-arm Bandit Problems
Concentration Inequalities

Proposition (Chernoff-Hoeffding Inequality)

Let $X_i \in [a_i, b_i]$ be $n$ independent r.v. with mean $\mu_i = \mathbb{E}X_i$. Then

$$\mathbb{P}\left[\left|\sum_{i=1}^{n}(X_i - \mu_i)\right| \geq \epsilon\right] \leq 2 \exp\left(-\frac{2\epsilon^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right).$$
Concentration Inequalities

**Proof.**

\[
\mathbb{P}\left( \sum_{i=1}^{n} X_i - \mu_i \geq \epsilon \right) = \mathbb{P}\left( e^{s} \sum_{i=1}^{n} X_i - \mu_i \geq e^{s\epsilon} \right)
\]

\[
\leq e^{-s\epsilon} \mathbb{E}[e^{s} \sum_{i=1}^{n} X_i - \mu_i], \quad \text{Markov inequality}
\]

\[
= e^{-s\epsilon} \prod_{i=1}^{n} \mathbb{E}[e^{s}(X_i - \mu_i)], \quad \text{independent random variables}
\]

\[
\leq e^{-s\epsilon} \prod_{i=1}^{n} e^{s^2(b_i - a_i)^2/8}, \quad \text{Hoeffding inequality}
\]

\[
= e^{-s\epsilon + s^2 \sum_{i=1}^{n} (b_i - a_i)^2 / 8}
\]

If we choose \( s = 4\epsilon / \sum_{i=1}^{n} (b_i - a_i)^2 \), the result follows. Similar arguments hold for \( \mathbb{P}\left( \sum_{i=1}^{n} X_i - \mu_i \leq -\epsilon \right) \).
Concentration Inequalities

**Finite sample guarantee:**

\[
\mathbb{P}\left[ \left| \frac{1}{n} \sum_{t=1}^{n} X_t - \mathbb{E}[X_1] \right| > \epsilon \right] \leq 2 \exp\left( - \frac{2n\epsilon^2}{(b-a)^2} \right)
\]

- \( \epsilon \): accuracy
- \( \epsilon \): confidence
- \( \text{deviation} \)
Concentration Inequalities

Finite sample guarantee:

\[ \mathbb{P} \left[ \left| \frac{1}{n} \sum_{t=1}^{n} X_t - \mathbb{E}[X_1] \right| > (b - a) \sqrt{\frac{\log 2/\delta}{2n}} \right] \leq \delta \]
Concentration Inequalities

Finite sample guarantee:

\[ \mathbb{P} \left[ \left| \frac{1}{n} \sum_{t=1}^{n} X_t - \mathbb{E}[X_1] \right| > \epsilon \right] \leq \delta \]

if \( n \geq \frac{(b-a)^2 \log 2/\delta}{2\epsilon^2} \).
Outline

Mathematical Tools

The General Multi-arm Bandit Problem

The Stochastic Multi-arm Bandit Problem

The Non-Stochastic Multi-arm Bandit Problem

Connections to Game Theory

Other Stochastic Multi-arm Bandit Problems
The learner has $i = 1, \ldots, N$ arms (options, experts, ...)

At each round $t = 1, \ldots, n$
The Multi–armed Bandit Game

The learner has $i = 1, \ldots, N$ arms (options, experts, ...)

At each round $t = 1, \ldots, n$

- At the same time
The Multi–armed Bandit Game

The learner has $i = 1, \ldots, N$ arms (options, experts, ...)

At each round $t = 1, \ldots, n$

- At the same time
  - The environment chooses a vector of rewards $\{X_{i,t}\}_{i=1}^{N}$
  - The learner chooses an arm $l_t$
The Multi–armed Bandit Game

The learner has $i = 1, \ldots, N$ arms (options, experts, ...)

At each round $t = 1, \ldots, n$

- At the same time
  - The environment chooses a vector of rewards $\{X_{i,t}\}_{i=1}^N$
  - The learner chooses an arm $I_t$
- The learner receives a reward $X_{I_t,t}$
The Multi–armed Bandit Game

The learner has $i = 1, \ldots, N$ arms (options, experts, ...)

At each round $t = 1, \ldots, n$

- At the same time
  - The environment chooses a vector of rewards $\{X_{i,t}\}_{i=1}^{N}$
  - The learner chooses an arm $l_t$

- The learner receives a reward $X_{l_t,t}$
- The environment does not reveal the rewards of the other arms
The General Multi-arm Bandit Problem

The Multi–armed Bandit Game (cont’d)

The regret

\[ R_n(A) = \max_{i=1,\ldots,N} \mathbb{E}\left[ \sum_{t=1}^{n} X_{i,t} \right] - \mathbb{E}\left[ \sum_{t=1}^{n} X_{I_t,t} \right] \]
The General Multi-arm Bandit Problem

The Multi–armed Bandit Game (cont’d)

The regret

\[ R_n(\mathcal{A}) = \max_{i=1,...,N} \mathbb{E} \left[ \sum_{t=1}^{n} X_{i,t} \right] - \mathbb{E} \left[ \sum_{t=1}^{n} X_{I_t,t} \right] \]

The expectation summarizes any possible source of randomness (either in \( X \) or in the algorithm)
The Exploration–Exploitation Lemma

**Problem 1:** The environment *does not* reveal the rewards of the arms not pulled by the learner.
The Exploration–Exploitation Lemma

**Problem 1**: The environment *does not* reveal the rewards of the arms not pulled by the learner
\[ \Rightarrow \text{the learner should } gain \textit{information} \text{ by repeatedly pulling all the arms} \]
The Exploration–Exploitation Lemma

**Problem 1:** The environment *does not* reveal the rewards of the arms not pulled by the learner
⇒ the learner should *gain information* by repeatedly pulling all the arms

**Problem 2:** Whenever the learner pulls a *bad arm*, it suffers some regret
The Exploration–Exploitation Lemma

**Problem 1**: The environment *does not* reveal the rewards of the arms not pulled by the learner
⇒ the learner should *gain information* by repeatedly pulling all the arms

**Problem 2**: Whenever the learner pulls a *bad arm*, it suffers some regret
⇒ the learner should *reduce the regret* by repeatedly pulling the best arm
The Exploration–Exploitation Lemma

**Problem 1:** The environment does not reveal the rewards of the arms not pulled by the learner
⇒ the learner should gain information by repeatedly pulling all the arms

**Problem 2:** Whenever the learner pulls a bad arm, it suffers some regret
⇒ the learner should reduce the regret by repeatedly pulling the best arm

**Challenge:** The learner should solve two opposite problems!
The Exploration–Exploitation Lemma

**Problem 1:** The environment *does not* reveal the rewards of the arms not pulled by the learner
⇒ the learner should *gain information* by repeatedly pulling all the arms ⇒ *exploration*

**Problem 2:** Whenever the learner pulls a *bad arm*, it suffers some regret
⇒ the learner should *reduce the regret* by repeatedly pulling the best arm

**Challenge:** The learner should solve two opposite problems!
The Exploration–Exploitation Lemma

**Problem 1:** The environment *does not* reveal the rewards of the arms not pulled by the learner
⇒ the learner should *gain information* by repeatedly pulling all the arms ⇒ *exploration*

**Problem 2:** Whenever the learner pulls a *bad arm*, it suffers some regret
⇒ the learner should *reduce the regret* by repeatedly pulling the best arm ⇒ *exploitation*

**Challenge:** The learner should solve two opposite problems!
The Exploration–Exploitation Lemma

**Problem 1**: The environment *does not* reveal the rewards of the arms not pulled by the learner
⇒ the learner should *gain information* by repeatedly pulling all the arms ⇒ *exploration*

**Problem 2**: Whenever the learner pulls a *bad arm*, it suffers some regret
⇒ the learner should *reduce the regret* by repeatedly pulling the best arm ⇒ *exploitation*

**Challenge**: The learner should solve the *exploration-exploitation* dilemma!
The Multi–armed Bandit Game (cont’d)

Examples

- Packet routing
- Clinical trials
- Web advertising
- Computer games
- Resource mining
- ...

Outline

Mathematical Tools

The General Multi-arm Bandit Problem

The Stochastic Multi-arm Bandit Problem

The Non-Stochastic Multi-arm Bandit Problem

Connections to Game Theory

Other Stochastic Multi-arm Bandit Problems
The Stochastic Multi–armed Bandit Problem

**Definition**

The environment is *stochastic*

- Each arm has a *distribution* \( \nu_i \) bounded in \([0, 1]\) and characterized by an *expected value* \( \mu_i \)
- The rewards are *i.i.d.* \( X_{i,t} \sim \nu_i \)
Notation

- Number of times arm \( i \) has been pulled after \( n \) rounds

\[
T_{i,n} = \sum_{t=1}^{n} \mathbb{I}\{l_t = i\}
\]
The Stochastic Multi–armed Bandit Problem (cont’d)

Notation

- Number of times arm $i$ has been pulled after $n$ rounds
  \[ T_{i,n} = \sum_{t=1}^{n} \mathbb{I}\{I_t = i\} \]

- Regret
  \[ R_n(A) = \max_{i=1,\ldots,N} \mathbb{E}\left[ \sum_{t=1}^{n} X_{i,t} \right] - \mathbb{E}\left[ \sum_{t=1}^{n} X_{I_t,t} \right] \]
The Stochastic Multi–armed Bandit Problem

Notation

- Number of times arm $i$ has been pulled after $n$ rounds

$$T_{i,n} = \sum_{t=1}^{n} \mathbb{I}\{I_t = i\}$$

- Regret

$$R_n(A) = \max_{i=1,\ldots,N} (n\mu_i) - \mathbb{E}\left[\sum_{t=1}^{n} X_{I_t,t}\right]$$
The Stochastic Multi–armed Bandit Problem (cont’d)

Notation

- Number of times arm $i$ has been pulled after $n$ rounds

$$T_{i,n} = \sum_{t=1}^{n} \mathbb{I}\{I_t = i\}$$

- Regret

$$R_n(A) = \max_{i=1,\ldots,N} (n\mu_i) - \sum_{i=1}^{N} \mathbb{E}[T_{i,n}]\mu_i$$
The Stochastic Multi–armed Bandit Problem (cont’d)

Notation

- Number of times arm \( i \) has been pulled after \( n \) rounds

\[
T_{i,n} = \sum_{t=1}^{n} \mathbb{I}\{l_t = i\}
\]

- Regret

\[
R_n(A) = n\mu^* - \sum_{i=1}^{N} \mathbb{E}[T_{i,n}]\mu_i
\]
The Stochastic Multi-armed Bandit Problem (cont’d)

Notation

- Number of times arm $i$ has been pulled after $n$ rounds

\[ T_{i,n} = \sum_{t=1}^{n} \mathbb{I}\{l_t = i\} \]

- Regret

\[ R_n(A) = \sum_{i \neq i^*} \mathbb{E}[T_{i,n}](\mu_{i^*} - \mu_i) \]
The Stochastic Multi-armed Bandit Problem (cont’d)

Notation

- Number of times arm $i$ has been pulled after $n$ rounds

\[ T_{i,n} = \sum_{t=1}^{n} \mathbb{I}\{I_t = i\} \]

- Regret

\[ R_n(A) = \sum_{i \neq i^*} \mathbb{E}[T_{i,n}] \Delta_i \]
The Stochastic Multi–armed Bandit Problem (cont’d)

Notation

- Number of times arm $i$ has been pulled after $n$ rounds
  
  \[ T_{i,n} = \sum_{t=1}^{n} \mathbb{I}\{l_t = i\} \]

- Regret
  
  \[ R_n(A) = \sum_{i \neq i^*} \mathbb{E}[T_{i,n}] \Delta_i \]

- Gap $\Delta_i = \mu_{i^*} - \mu_i$
The Stochastic Multi–armed Bandit Problem (cont’d)

\[ R_n(\mathcal{A}) = \sum_{i \neq i^*} \mathbb{E}[T_{i,n}] \Delta_i \]

⇒ we only need to study the expected number of pulls of the suboptimal arms
Optimism in Face of Uncertainty Learning (OFUL)

Whenever we are uncertain about the outcome of an arm, we consider the best possible world and choose the best arm.
Optimism in Face of Uncertainty Learning (OFUL)

Whenever we are *uncertain* about the outcome of an arm, we consider the *best possible world* and choose the *best arm*.

**Why it works:**

- If the *best possible world* is correct ⇒ *no regret*
- If the *best possible world* is wrong ⇒ *the reduction in the uncertainty is maximized*
The Stochastic Multi–armed Bandit Problem (cont’d)

- Pulls = 100
- Pulls = 200
- Pulls = 50
- Pulls = 20
The Stochastic Multi-arm Bandit Problem (cont’d)

Optimism in face of uncertainty
The Upper–Confidence Bound (UCB) Algorithm

The idea

![Graph showing the relationship between arms and rewards.](image-url)
The Upper–Confidence Bound (UCB) Algorithm

Show time!
The Stochastic Multi-arm Bandit Problem

The Upper–Confidence Bound (UCB) Algorithm (cont’d)

At each round \( t = 1, \ldots, n \)

- Compute the **score** of each arm \( i \)

\[
B_i = (\text{optimistic score of arm } i)
\]

- Pull arm

\[
l_t = \arg \max_{i=1,\ldots,N} B_{i,s,t}
\]

- Update the number of pulls \( T_{l_t,t} = T_{l_t,t-1} + 1 \)
The Upper–Confidence Bound (UCB) Algorithm (cont’d)

The score (with parameters $\rho$ and $\delta$)

$$B_i = (\text{optimistic score of arm } i)$$
The score (with parameters $\rho$ and $\delta$)

$B_{i,s,t} = (\text{optimistic score of arm } i \text{ if pulled } s \text{ times up to round } t)$
The score (with parameters $\rho$ and $\delta$)

$$B_{i,s,t} = (\text{optimistic score of arm } i \text{ if pulled } s \text{ times up to round } t)$$

Optimism in face of uncertainty:

Current knowledge: average rewards $\hat{\mu}_{i,s}$

Current uncertainty: number of pulls $s$
The score (with parameters $\rho$ and $\delta$)

$$B_{i,s,t} = \text{knowledge} + \text{uncertainty}$$

Optimism in face of uncertainty:
- Current knowledge: average rewards $\hat{\mu}_{i,s}$
- Current uncertainty: number of pulls $s$
The score (with parameters $\rho$ and $\delta$)

$$B_{i,s,t} = \hat{\mu}_{i,s} + \rho \sqrt{\frac{\log 1/\delta}{2s}}$$

Optimism in face of uncertainty:

*Current knowledge:* average rewards $\hat{\mu}_{i,s}$

*Current uncertainty:* number of pulls $s$
The Upper–Confidence Bound (UCB) Algorithm (cont’d)

Do you remember Chernoff-Hoeffding?

**Theorem**

Let $X_1, \ldots, X_n$ be i.i.d. samples from a distribution bounded in $[a, b]$, then for any $\delta \in (0, 1)$

$$
P \left[ \left| \frac{1}{n} \sum_{t=1}^{n} X_t - \mathbb{E}[X_1] \right| > (b - a) \sqrt{\frac{\log 2/\delta}{2n}} \right] \leq \delta$$
After $s$ pulls, arm $i$

$$\mathbb{P}\left[ \mathbb{E}[X_i] \leq \frac{1}{s} \sum_{t=1}^{s} X_{i,t} + \sqrt{\frac{\log 1/\delta}{2s}} \right] \geq 1 - \delta$$
The Stochastic Multi-arm Bandit Problem

The Upper–Confidence Bound (UCB) Algorithm (cont’d)

After $s$ pulls, arm $i$

$$\mathbb{P} \left[ \mu_i \leq \hat{\mu}_{i,s} + \sqrt{\frac{\log 1/\delta}{2s}} \right] \geq 1 - \delta$$
The Upper–Confidence Bound (UCB) Algorithm (cont’d)

After $s$ pulls, arm $i$

$$\mathbb{P}\left[ \mu_i \leq \hat{\mu}_{i,s} + \sqrt{\frac{\log 1/\delta}{2s}} \right] \geq 1 - \delta$$

$\Rightarrow$ UCB uses an \textit{upper confidence bound} on the expectation
The Upper–Confidence Bound (UCB) Algorithm (cont’d)

**Theorem**

For any set of \( N \) arms with distributions bounded in \([0, b]\), if \( \delta = 1/t \), then \( UCB(\rho) \) with \( \rho > 1 \), achieves a regret

\[
R_n(A) \leq \sum_{i \neq i^*} \left[ \frac{4b^2}{\Delta_i} \rho \log(n) + \Delta_i \left( \frac{3}{2} + \frac{1}{2(\rho - 1)} \right) \right]
\]
The Stochastic Multi-arm Bandit Problem

The Upper–Confidence Bound (UCB) Algorithm (cont’d)

Let $N = 2$ with $i^* = 1$

$$R_n(A) \leq O\left(\frac{1}{\Delta} \rho \log(n)\right)$$

**Remark 1:** the *cumulative* regret slowly increases as $\log(n)$
Let $N = 2$ with $i^* = 1$

$$R_n(A) \leq O\left(\frac{1}{\Delta} \rho \log(n)\right)$$

**Remark 1**: the *cumulative* regret slowly increases as $\log(n)$

**Remark 2**: the *smaller the gap* the *bigger the regret*... why?
The Upper–Confidence Bound (UCB) Algorithm (cont’d)

Show time (again)!
The Worst–case Performance

**Remark:** the regret bound is *distribution–dependent*

\[
R_n(A; \Delta) \leq O\left( \frac{1}{\Delta} \rho \log(n) \right)
\]
The Worst–case Performance

**Remark**: the regret bound is *distribution–dependent*

\[
R_n(A; \Delta) \leq O\left( \frac{1}{\Delta} \rho \log(n) \right)
\]

**Meaning**: the algorithm is able to *adapt to the specific problem* at hand!
The Worst–case Performance

**Remark:** the regret bound is *distribution–dependent*

\[ R_n(\mathcal{A}; \Delta) \leq O\left(\frac{1}{\Delta} \rho \log(n)\right) \]

**Meaning:** the algorithm is able to *adapt to the specific problem* at hand!

**Worst–case performance:** what is the distribution which leads to the worst possible performance of UCB? what is the distribution–free performance of UCB?

\[ R_n(\mathcal{A}) = \sup_{\Delta} R_n(\mathcal{A}; \Delta) \]
The Worst–case Performance

Problems: it seems like if $\Delta \to 0$ then the regret tends to infinity...
The Worst–case Performance

**Problem:** it seems like if \( \Delta \to 0 \) then the regret tends to infinity...

... nosense because the regret is defined as

\[
R_n(\mathcal{A}; \Delta) = \mathbb{E}[T_{2,n}] \Delta
\]
Problem: it seems like if $\Delta \to 0$ then the regret tends to infinity...

... nonsense because the regret is defined as

$$R_n(\mathcal{A}; \Delta) = \mathbb{E}[T_{2,n}]\Delta$$

then if $\Delta_i$ is small, the regret is also small...
The Worst–case Performance

**Problem:** it seems like if $\Delta \to 0$ then the regret tends to infinity...

... nosense because the regret is defined as

$$R_n(\mathcal{A}; \Delta) = \mathbb{E}[T_{2,n}]\Delta$$

then if $\Delta_i$ is small, the regret is also small...

In fact

$$R_n(\mathcal{A}; \Delta) = \min \left\{ O\left(\frac{1}{\Delta} \rho \log(n)\right), \mathbb{E}[T_{2,n}]\Delta \right\}$$
The Worst–case Performance

Then

\[
R_n(\mathcal{A}) = \sup_{\Delta} R_n(\mathcal{A}; \Delta) = \sup_{\Delta} \min\left\{ O\left( \frac{1}{\Delta} \rho \log(n) \right), n\Delta \right\} \approx \sqrt{n}
\]

for \( \Delta = \sqrt{1/n} \).
Tuning the confidence $\delta$ of UCB

Remark: UCB is an \textit{anytime} algorithm ($\delta = 1/t$)

$$B_{i,s,t} = \hat{\mu}_{i,s} + \rho \sqrt{\frac{\log t}{2s}}$$
Tuning the confidence $\delta$ of UCB

**Remark:** UCB is an *anytime* algorithm ($\delta = 1/t$)

$$B_{i,s,t} = \hat{\mu}_{i,s} + \rho \sqrt{\frac{\log t}{2s}}$$

**Remark:** If the time horizon $n$ is known then the optimal choice is $\delta = 1/n$

$$B_{i,s,t} = \hat{\mu}_{i,s} + \rho \sqrt{\frac{\log n}{2s}}$$
Tuning the confidence $\delta$ of UCB (cont’d)

**Intuition:** UCB should pull the suboptimal arms

- *Enough:* so as to understand which arm is the best
- *Not too much:* so as to keep the regret as small as possible
Intuition: UCB should pull the suboptimal arms

- **Enough**: so as to understand which arm is the best
- **Not too much**: so as to keep the regret as small as possible

The confidence $1 - \delta$ has the following impact (similar for $\rho$)

- **Big $1 - \delta$**: high level of *exploration*
- **Small $1 - \delta$**: high level of *exploitation*
Tuning the confidence $\delta$ of UCB (cont’d)

**Intuition**: UCB should pull the suboptimal arms

- *Enough*: so as to understand which arm is the best
- *Not too much*: so as to keep the regret as small as possible

The confidence $1 - \delta$ has the following impact (similar for $\rho$)

- *Big* $1 - \delta$: high level of *exploration*
- *Small* $1 - \delta$: high level of *exploitation*

**Solution**: depending on the time horizon, we can tune how to trade-off between exploration and exploitation
Tuning the confidence $\delta$ of UCB (cont’d)

Let’s dig into the (1 page and half!!) proof.

Define the (high-probability) event \[statistics\]

\[ E = \left\{ \forall i, s \mid \hat{\mu}_{i,s} - \mu_i \leq \sqrt{\frac{\log 1/\delta}{2s}} \right\} \]

By Chernoff-Hoeffding \( P[E] \geq 1 - nN\delta. \)
Tuning the confidence $\delta$ of UCB (cont’d)

Let’s dig into the (1 page and half!!) proof.

Define the (high-probability) event $[\text{statistics}]$

$$\mathcal{E} = \left\{ \forall i, s \mid |\hat{\mu}_{i,s} - \mu_i| \leq \sqrt{\frac{\log 1/\delta}{2s}} \right\}$$

By Chernoff-Hoeffding $\mathbb{P}[\mathcal{E}] \geq 1 - nN\delta$.

At time $t$ we pull arm $i$ $[\text{algorithm}]

$$B_{i, T_{i,t-1}} \geq B_{i^*, T_{i^*,t-1}}$$
Tuning the confidence $\delta$ of UCB (cont’d)

Let’s dig into the (1 page and half!!) proof.

Define the (high-probability) event $[\text{statistics}]$

$$
\mathcal{E} = \left\{ \forall i, s \mid \hat{\mu}_{i,s} - \mu_i \leq \sqrt{\frac{\log 1/\delta}{2s}} \right\}
$$

By Chernoff-Hoeffding $\mathbb{P}[\mathcal{E}] \geq 1 - nN\delta$.

At time $t$ we pull arm $i$ $[\text{algorithm}]$

$$
\hat{\mu}_{i, T_{i,t-1}} + \sqrt{\frac{\log 1/\delta}{2 T_{i,t-1}}} \geq \hat{\mu}_{i^*, T_{i^*,t-1}} + \sqrt{\frac{\log 1/\delta}{2 T_{i^*,t-1}}}
$$
Tuning the confidence $\delta$ of UCB (cont’d)

Let’s dig into the (1 page and half!!) proof.

Define the (high-probability) event \([\text{statistics}]\)

$$\mathcal{E} = \left\{ \forall i, s \left| \hat{\mu}_{i,s} - \mu_i \right| \leq \sqrt{\frac{\log 1/\delta}{2s}} \right\}$$

By Chernoff-Hoeffding $\mathbb{P}[\mathcal{E}] \geq 1 - nN\delta$.

At time $t$ we pull arm $i$ \([\text{algorithm}]\)

$$\hat{\mu}_{i,T_i,t-1} + \sqrt{\frac{\log 1/\delta}{2T_i,t-1}} \geq \hat{\mu}_{i^*,T_{i^*},t-1} + \sqrt{\frac{\log 1/\delta}{2T_{i^*},t-1}}$$

On the event $\mathcal{E}$ we have \([\text{math}]\)

$$\mu_i + 2\sqrt{\frac{\log 1/\delta}{2T_i,t-1}} \geq \mu_{i^*}$$
Tuning the confidence $\delta$ of UCB (cont’d)

Assume $t$ is the last time $i$ is pulled, then $T_{i,n} = T_{i,t-1} + 1$, thus

$$\mu_i + 2\sqrt{\frac{\log 1/\delta}{2(T_{i,n} - 1)}} \geq \mu_{i^*}$$
Tuning the confidence $\delta$ of UCB (cont’d)

Assume $t$ is the last time $i$ is pulled, then $T_{i,n} = T_{i,t-1} + 1$, thus

$$\mu_i + 2\sqrt{\frac{\log 1/\delta}{2(T_{i,n} - 1)}} \geq \mu_i^*$$

Reordering $\mathit{[math]}$

$$T_{i,n} \leq \frac{\log 1/\delta}{2\Delta_i^2} + 1$$

under event $\mathcal{E}$ and thus with probability $1 - nN\delta$. 
Tuning the confidence $\delta$ of UCB (cont’d)

Assume $t$ is the last time $i$ is pulled, then $T_{i,n} = T_{i,t-1} + 1$, thus

$$\mu_i + 2\sqrt{\frac{\log 1/\delta}{2(T_{i,n} - 1)}} \geq \mu_i^*$$

Reordering $[\text{math}]$

$$T_{i,n} \leq \frac{\log 1/\delta}{2\Delta_i^2} + 1$$

under event $\mathcal{E}$ and thus with probability $1 - nN\delta$.

Moving to the expectation $[\text{statistics}]$

$$\mathbb{E}[T_{i,n}] = \mathbb{E}[T_{i,n}\mathbb{1}_\mathcal{E}] + \mathbb{E}[T_{i,n}\mathbb{1}_\mathcal{E}^C]$$
Tuning the confidence $\delta$ of UCB (cont’d)

Assume $t$ is the last time $i$ is pulled, then $T_{i,n} = T_{i,t-1} + 1$, thus

$$\mu_i + 2\sqrt{\frac{\log 1/\delta}{2(T_{i,n} - 1)}} \geq \mu_i^*$$

Reordering $[math]$

$$T_{i,n} \leq \frac{\log 1/\delta}{2\Delta_i^2} + 1$$

under event $\mathcal{E}$ and thus with probability $1 - nN\delta$.

Moving to the expectation $[statistics]$

$$\mathbb{E}[T_{i,n}] \leq \frac{\log 1/\delta}{2\Delta_i^2} + 1 + n(nN\delta)$$
The Stochastic Multi-arm Bandit Problem

Tuning the confidence $\delta$ of UCB (cont’d)

Assume $t$ is the last time $i$ is pulled, then $T_{i,n} = T_{i,t-1} + 1$, thus

$$\mu_i + 2\sqrt{\frac{\log 1/\delta}{2(\frac{T_{i,n}}{2} - 1)}} \geq \mu_i^*$$

Reordering $\left[math\right]$

$$T_{i,n} \leq \frac{\log 1/\delta}{2\Delta_i^2} + 1$$

under event $E$ and thus with probability $1 - nN\delta$.

Moving to the expectation $\left[statistics\right]$

$$E[T_{i,n}] \leq \frac{\log 1/\delta}{2\Delta_i^2} + 1 + n(nN\delta)$$

Trading-off the two terms $\delta = 1/n^2$, we obtain

$$\hat{\mu}_{i,T_{i,t-1}} + \sqrt{\frac{2\log n}{2T_{i,t-1}}}$$
Trading-off the two terms $\delta = 1/n^2$, we obtain

$$\hat{\mu}_{i, T_{i, t-1}} + \sqrt{\frac{2 \log n}{2T_{i, t-1}}}$$

and

$$\mathbb{E}[T_{i, n}] \leq \frac{\log n}{\Delta_i^2} + 1 + N$$
Tuning the confidence $\delta$ of UCB (cont’d)

Multi–armed Bandit: the same for $\delta = 1/t$ and $\delta = 1/n$...
Tuning the confidence $\delta$ of UCB (cont’d)

**Multi–armed Bandit:** the same for $\delta = 1/t$ and $\delta = 1/n$...

... **almost** (i.e., in expectation)
The Stochastic Multi-arm Bandit Problem

Tuning the confidence $\delta$ of UCB (cont’d)

The value–at–risk of the regret for UCB-anytime
Tuning the $\rho$ of UCB (cont’d)

UCB values (for the $\delta = 1/n$ algorithm)

$$B_{i,s} = \hat{\mu}_{i,s} + \rho \sqrt{\frac{\log n}{2s}}$$
Tuning the $\rho$ of UCB (cont’d)

UCB values (for the $\delta = 1/n$ algorithm)

$$B_{i,s} = \hat{\mu}_{i,s} + \rho \sqrt{\frac{\log n}{2s}}$$

Theory
- $\rho < 0.5$, polynomial regret w.r.t. $n$
- $\rho > 0.5$, logarithmic regret w.r.t. $n$
Tuning the $\rho$ of UCB (cont’d)

UCB values (for the $\delta = 1/n$ algorithm)

$$B_{i,s} = \hat{\mu}_{i,s} + \rho \sqrt{\frac{\log n}{2s}}$$

Theory
- $\rho < 0.5$, polynomial regret w.r.t. $n$
- $\rho > 0.5$, logarithmic regret w.r.t. $n$

Practice: $\rho = 0.2$ is often the best choice
Tuning the $\rho$ of UCB (cont’d)

UCB values (for the $\delta = 1/n$ algorithm)

$$B_{i,s} = \hat{\mu}_{i,s} + \rho \sqrt{\frac{\log n}{2s}}$$

Theory
- $\rho < 0.5$, polynomial regret w.r.t. $n$
- $\rho > 0.5$, logarithmic regret w.r.t. $n$

Practice: $\rho = 0.2$ is often the best choice
Improvements over UCB: UCB-V

**Idea:** use Bernstein bounds with empirical variance
Improvements over UCB: UCB-V

**Idea:** use Bernstein bounds with empirical variance

**Algorithm:**

\[
B_{i,s,t} = \hat{\mu}_{i,s} + \sqrt{\frac{\log t}{2s}}
\]

\[
B_{i,s,t}^V = \hat{\mu}_{i,s} + \sqrt{\frac{2\hat{\sigma}_{i,s}^2 \log t}{s}} + \frac{8 \log t}{3s}
\]

\[
R_n \leq O\left(\frac{1}{\Delta} \log n\right)
\]

\[
R_n \leq O\left(\frac{\sigma^2}{\Delta} \log n\right)
\]
Improvements over UCB: KL-UCB

**Idea**: use Kullback–Leibler bounds which are tighter than other bounds
Improvements over UCB: KL-UCB

**Idea:** use Kullback–Leibler bounds which are tighter than other bounds

**Algorithm:** the algorithm is still index-based but a bit more complicated

\[ R_n \leq O\left(\frac{1}{\Delta} \log n\right) \]

\[ R_n \leq O\left(\frac{1}{KL(\nu, \nu_{i*})} \log n\right) \]
Improvements over UCB: Thompson strategy

**Idea:** Keep a distribution over the possible values of $\mu_i$
Improvements over UCB: Thompson strategy

**Idea**: Keep a distribution over the possible values of $\mu_i$

**Algorithm**: Bayesian approach. Compute the posterior distributions given the samples.
Back to UCB: the Lower Bound

**Theorem**

*For any stochastic bandit \( \{\nu_i\} \), any algorithm \( A \) has a regret*

\[
\lim_{n \to \infty} \frac{R_n}{\log n} \geq \inf_{\nu} KL(\nu_i, \nu)
\]

\[
\Delta_i
\]

\[
\inf_{\nu} KL(\nu_i, \nu)
\]
**Theorem**

For any stochastic bandit \( \{\nu_i\} \), any algorithm \( \mathcal{A} \) has a regret

\[
\lim_{n \to \infty} \frac{R_n}{\log n} \geq \inf_{\nu} KL(\nu_i, \nu) \Delta_i
\]

**Problem**: this is just asymptotic
The Stochastic Multi-arm Bandit Problem

Back to UCB: the Lower Bound

Theorem

For any stochastic bandit \( \{\nu_i\} \), any algorithm \( A \) has a regret

\[
\lim_{n \to \infty} \frac{R_n}{\log n} \geq \inf_{\nu} \frac{\Delta_i}{KL(\nu, \nu)}
\]

**Problem**: this is just asymptotic

**Open Question**: what is the finite-time lower bound?
The Non-Stochastic Multi-arm Bandit Problem

Outline

Mathematical Tools

The General Multi-arm Bandit Problem

The Stochastic Multi-arm Bandit Problem

The Non-Stochastic Multi-arm Bandit Problem

Connections to Game Theory

Other Stochastic Multi-arm Bandit Problems
The Non–Stochastic Multi–armed Bandit Problem

Definition

The environment is adversarial

- Arms have no fixed distribution
- The rewards $X_{i,t}$ are arbitrarily chosen by the environment
The (non–stochastic bandit) regret

\[ R_n(A) = \max_{i=1,\ldots,N} \mathbb{E}\left[ \sum_{t=1}^{n} X_{i,t} \right] - \mathbb{E}\left[ \sum_{t=1}^{n} X_{I_t,t} \right] \]
The (non-stochastic bandit) regret

\[ R_n(A) = \max_{i=1,\ldots,N} \sum_{t=1}^{n} X_{i,t} - \mathbb{E} \left[ \sum_{t=1}^{n} X_{I_t,t} \right] \]
The Exponentially Weighted Average Forecaster

Initialize the weights $w_{i,0} = 1$

- Compute ($W_{t-1} = \sum_{i=1}^{N} w_{i,t-1}$)

  $\hat{p}_{i,t} = \frac{w_{i,t-1}}{W_{t-1}}$

- Choose the arm at random

  $I_t \sim \hat{p}_t$

- Observe the rewards $\{X_{i,t}\}$

- Receive a reward $X_{I_{t},t}$

- Update

  $w_{i,t} = w_{i,t-1} \exp\left( + \eta X_{i,t,t} \right)$
**Problem:** we only observe the reward of the specific arm chosen at time $t$ (i.e., only $X_{l_t,t}$ is observed)
The Exponentially Weighted Average Forecaster

Initialize the weights $w_{i,0} = 1$

- Compute ($W_{t-1} = \sum_{i=1}^{N} w_{i,t-1}$)
  \[
  \hat{p}_{i,t} = \frac{w_{i,t-1}}{W_{t-1}}
  \]

- Choose the arm at random
  \[I_t \sim \hat{p}_t\]

- Observe the rewards $\{X_{i,t}\}$

- Receive a reward $X_{I_t,t}$

- Update
  \[w_{i,t} = w_{i,t-1} \exp(\eta X_{i,t}) \Rightarrow \text{this update is not possible}\]
We use the *importance weight* trick

\[
\hat{X}_{i,t} = \begin{cases} 
\frac{X_{i,t}}{\hat{p}_{i,t}} & \text{if } i = l_t \\
0 & \text{otherwise}
\end{cases}
\]
The Non–Stochastic Multi–armed Bandit Problem (cont’d)

We use the *importance weight* trick

\[ \hat{X}_{i,t} = \begin{cases} \frac{X_{i,t}}{\hat{p}_{i,t}} & \text{if } i = t \\ 0 & \text{otherwise} \end{cases} \]

Why it is a good idea:

\[ \mathbb{E} [ \hat{X}_{i,t} ] = \frac{X_{i,t}}{\hat{p}_{i,t}} \hat{p}_{i,t} + 0 (1 - \hat{p}_{i,t}) = X_{i,t} \]

\( \hat{X}_{i,t} \) is an *unbiased* estimator of \( X_{i,t} \)
The Exp3 Algorithm

**Exp3**: Exponential-weight algorithm for Exploration and Exploitation

1. Initialize the weights $w_{i,0} = 1$
   - Compute ($W_{t-1} = \sum_{i=1}^{N} w_{i,t-1}$)
     $$\hat{p}_{i,t} = \frac{w_{i,t-1}}{W_{t-1}}$$
   - Choose the arm at random
     $$l_t \sim \hat{p}_t$$
   - Receive a reward $X_{l_t,t}$
   - Update
     $$w_{i,t} = w_{i,t-1} \exp (\eta \hat{X}_{l_t,t})$$
The Exp3 Algorithm

**Question**: is this enough? is this algorithm actually exploring enough?
The Exp3 Algorithm

**Question**: is this enough? is this algorithm actually exploring enough?

**Answer**: more or less...

- Exp3 has a small regret *in expectation*
- Exp3 might have large deviations with *high probability* (ie, from time to time it may *concentrate* $\hat{p}_t$ on the wrong arm for too long and then incur a large regret)
The Exp3 Algorithm

**Fix:** add some extra uniform exploration

Initialize the weights \( w_{i,0} = 1 \)

- Compute \( W_{t-1} = \sum_{i=1}^{N} w_{i,t-1} \)
  \[
  \hat{p}_{i,t} = (1 - \gamma) \frac{w_{i,t-1}}{W_{t-1}} + \frac{\gamma}{K}
  \]

- Choose the arm at random
  \[
  I_t \sim \hat{p}_t
  \]

- Receive a reward \( X_{I_t,t} \)

- Update
  \[
  w_{i,t} = w_{i,t-1} \exp \left( \eta \hat{X}_{I_t,t} \right)
  \]
The Non-Stochastic Multi-arm Bandit Problem

The Exp3 Algorithm

**Theorem**

If Exp3 is run with $\gamma = \eta$, then it achieves a regret

$$R_n(A) = \max_{i=1, \ldots, N} \sum_{t=1}^{n} X_{i,t} - \mathbb{E} \left[ \sum_{t=1}^{n} X_{I_t,t} \right] \leq (e - 1) \gamma G_{\max} + \frac{N \log N}{\gamma}$$

with $G_{\max} = \max_{i=1, \ldots, N} \sum_{t=1}^{n} X_{i,t}$. 
The Exp3 Algorithm

Theorem

If Exp3 is run with

\[ \gamma = \eta = \sqrt{\frac{N \log N}{(e - 1)n}} \]

then it achieves a regret

\[ R_n(A) \leq O(\sqrt{nN \log N}) \]
The Exp3 Algorithm

Comparison with online learning

\[ R_n(\text{Exp3}) \leq O(\sqrt{nN \log N}) \]

\[ R_n(\text{EWA}) \leq O(\sqrt{n \log N}) \]
The Exp3 Algorithm

Comparison with online learning

\[ R_n(\text{Exp3}) \leq O(\sqrt{nN \log N}) \]

\[ R_n(\text{EWA}) \leq O(\sqrt{n \log N}) \]

**Intuition**: in online learning at each round we obtain \( N \) feedbacks, while in bandits we receive 1 feedback.
The Improved-Exp3 Algorithm

Initialize the weights $w_{i,0} = 1$

1. Compute $(W_{t-1} = \sum_{i=1}^{N} w_{i,t-1})$
   
   \[
   \hat{p}_{i,t} = (1 - \gamma) \frac{w_{i,t-1}}{W_{t-1}} + \frac{\gamma}{K} 
   \]

2. Choose the arm at random
   
   $I_t \sim \hat{p}_t$

3. Receive a reward $X_{I_t,t}$

4. Compute
   
   \[
   \tilde{X}_{i,t} = \hat{X}_{i,t} + \frac{\beta}{\hat{p}_{i,t}} 
   \]

5. Update
   
   \[
   w_{i,t} = w_{i,t-1} \exp \left( \eta \tilde{X}_{i_t,t} \right) 
   \]
The Non-Stochastic Multi-arm Bandit Problem

The Improved-Exp3 Algorithm

Theorem

If Improved-Exp3 is run with parameters in the ranges

\[
\gamma \leq \frac{1}{2}; \quad 0 \leq \eta \leq \frac{\gamma}{2N}; \quad \sqrt{\frac{1}{nN} \log \frac{N}{\delta}} \leq \beta \leq 1
\]

then it achieves a regret

\[
R_n^{HP}(A) \leq n(\gamma + \eta(1 + \beta)N) + \frac{\log N}{\eta} + 2nN\beta
\]

with probability at least \(1 - \delta\).
The Non-Stochastic Multi-arm Bandit Problem

The Improved-Exp3 Algorithm

Theorem

If Improved-Exp3 is run with parameters in the ranges

\[ \beta = \sqrt{\frac{1}{nN} \log \frac{N}{\delta}}; \quad \gamma = \frac{4N\beta}{3 + \beta}; \quad \eta = \frac{\gamma}{2N} \]

then it achieves a regret

\[ R_n^{HP}(\mathcal{A}) \leq \frac{11}{2} \sqrt{nN \log(\frac{N}{\delta})} + \frac{\log N}{2} \]

with probability at least \(1 - \delta\).
Outline

Mathematical Tools

The General Multi-arm Bandit Problem

The Stochastic Multi-arm Bandit Problem

The Non-Stochastic Multi-arm Bandit Problem

Connections to Game Theory

Other Stochastic Multi-arm Bandit Problems
### Repeated Two-Player Zero-Sum Games

A two-player zero-sum game

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30, -30</td>
<td>-10, 10</td>
<td>20, -20</td>
</tr>
<tr>
<td>2</td>
<td>10, -10</td>
<td>-20, 20</td>
<td>-20, 20</td>
</tr>
</tbody>
</table>
Repeate Two–Player Zero–Sum Games

A two–player zero–sum game

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30, -30</td>
<td>-10, 10</td>
<td>20, -20</td>
</tr>
<tr>
<td>2</td>
<td>10, -10</td>
<td>-20, 20</td>
<td>-20, 20</td>
</tr>
</tbody>
</table>

Nash equilibrium:
A set of strategies is a Nash equilibrium if no player can do better by unilaterally changing his strategy.
Repeated Two–Player Zero–Sum Games

A two–player zero–sum game

\[
\begin{array}{c|ccc}
 & A & B & C \\
\hline
1 & 30, -30 & -10, 10 & 20, -20 \\
2 & 10, -10 & -20, 20 & -20, 20 \\
\end{array}
\]

Nash equilibrium:

Red: take action 1 with prob. 4/7 and action 2 with prob. 3/7

Blue: take action A with prob. 0, action B with prob. 4/7, and action C with prob. 3/7
Repeated Two–Player Zero–Sum Games

A two–player zero–sum game

\[
\begin{array}{|c|c|c|}
\hline
 & A & B \\
\hline
1 & 30, -30 & -10, 10 \\
2 & 10, -10 & -20, 20 \\
\hline
\end{array}
\]

Nash equilibrium:
Value of the game: \( V = 20/7 \) (reward of Red at the equilibrium)
Repeated Two–Player Zero–Sum Games

At each round $t$

- Row player computes a mixed strategy $\hat{p}_t = (\hat{p}_1, t, \ldots, \hat{p}_N, t)$
- Column player computes a mixed strategy $\hat{q}_t = (\hat{q}_1, t, \ldots, \hat{q}_M, t)$
Repeate Two–Player Zero–Sum Games

At each round $t$

- Row player computes a mixed strategy $\hat{p}_t = (\hat{p}_{1,t}, \ldots, \hat{p}_{N,t})$
- Column player computes a mixed strategy $\hat{q}_t = (\hat{q}_{1,t}, \ldots, \hat{q}_{M,t})$
- Row player selects action $I_t \in \{1, \ldots, N\}$
- Column player selects action $J_t \in \{1, \ldots, M\}$
Repeated Two–Player Zero–Sum Games

At each round $t$

- Row player computes a mixed strategy $\hat{p}_t = (\hat{p}_{1,t}, \ldots, \hat{p}_{N,t})$
- Column player computes a mixed strategy $\hat{q}_t = (\hat{q}_{1,t}, \ldots, \hat{q}_{M,t})$
- Row player selects action $I_t \in \{1, \ldots, N\}$
- Column player selects action $J_t \in \{1, \ldots, M\}$
- Row player suffers $\ell(I_t, J_t)$
- Column player suffers $-\ell(I_t, J_t)$
Repeated Two–Player Zero–Sum Games

At each round $t$

- Row player computes a mixed strategy $\hat{p}_t = (\hat{p}_{1,t}, \ldots, \hat{p}_{N,t})$
- Column player computes a mixed strategy $\hat{q}_t = (\hat{q}_{1,t}, \ldots, \hat{q}_{M,t})$
- Row player selects action $I_t \in \{1, \ldots, N\}$
- Column player selects action $J_t \in \{1, \ldots, M\}$
- Row player suffers $\ell(I_t, J_t)$
- Column player suffers $-\ell(I_t, J_t)$

Value of the game

$$V = \max_p \min_q \bar{\ell}(p, q)$$

with

$$\bar{\ell}(p, q) = \sum_{i=1}^{N} \sum_{j=1}^{M} p_i q_j \ell(i, j)$$
Repeated Two-Player Zero-Sum Games

**Question**: what if the two players are both bandit algorithms (e.g., Exp3)?
Repetitive Two-Player Zero-Sum Games

Question: what if the two players are both bandit algorithms (e.g., Exp3)?

Row player: a bandit algorithm is able to minimize

$$R_n(\text{row}) = \sum_{t=1}^{n} \ell_{l_t, J_t} - \min_{i=1,...,N} \sum_{t=1}^{n} \ell_{i, J_t}$$
Repeated Two–Player Zero–Sum Games

**Question:** what if the two players are both bandit algorithms (e.g., Exp3)?

**Row player:** a bandit algorithm is able to minimize

\[
R_n(\text{row}) = \sum_{t=1}^{n} \ell_{l_t,j_t} - \min_{i=1,\ldots,N} \sum_{t=1}^{n} \ell_{i,j_t}
\]

**Col player:** a bandit algorithm is able to minimize

\[
R_n(\text{col}) = \sum_{t=1}^{n} \ell_{l_t,j_t} - \min_{j=1,\ldots,M} \sum_{t=1}^{n} \ell_{l_t,j}
\]
Reepeated Two–Player Zero–Sum Games

**Theorem**

If both the row and column players play according to an Hannan-consistent strategy, then

\[ \limsup_{n \to \infty} \frac{1}{n} \sum_{t=1}^{n} \ell(I_t, J_t) = V \]
Repeated Two–Player Zero–Sum Games

**Theorem**

The empirical distribution of plays

\[ \hat{p}_{i,n} = \frac{1}{n} \sum_{t=1}^{n} \mathbb{I}\{I_t = i\} \quad \hat{q}_{j,n} = \frac{1}{n} \sum_{t=1}^{n} \mathbb{I}\{J_t = j\} \]

induces a product distribution \( \hat{p}_n \times \hat{q}_n \) which converges to the set of Nash equilibria \( p \times q \).
Repeated Two–Player Zero–Sum Games

Proof idea.
Since $\overline{\ell}(p, J_t)$ is linear, over the simplex, the minimum is at one of the corners

$$\min_{i=1,...,N} \frac{1}{N} \sum_{t=1}^{n} \ell(i, J_t) = \min_p \frac{1}{n} \sum_{t=1}^{n} \overline{\ell}(p, J_t)$$
Repeated Two–Player Zero–Sum Games

Proof idea.
Since $\bar{\ell}(p, J_t)$ is linear, over the simplex, the minimum is at one of the corners $[math]$

$$\min_{i=1,\ldots,N} \frac{1}{N} \sum_{t=1}^{n} \ell(i, J_t) = \min_p \frac{1}{n} \sum_{t=1}^{n} \bar{\ell}(p, J_t)$$

We consider the empirical probability of the row player $[def]$

$$\hat{q}_{j,n} = \frac{1}{n} \sum_{t=1}^{n} \mathbb{I} J_t = j$$
Repeated Two–Player Zero–Sum Games

Proof idea.
Since $\bar{\ell}(p, J_t)$ is linear, over the simplex, the minimum is at one of the corners [math]

$$\min_{i=1,\ldots,N} \frac{1}{N} \sum_{t=1}^{n} \ell(i, J_t) = \min_p \frac{1}{n} \sum_{t=1}^{n} \bar{\ell}(p, J_t)$$

We consider the empirical probability of the row player [def]

$$\hat{q}_{j,n} = \frac{1}{n} \sum_{t=1}^{n} \mathbb{I}_{J_t = j}$$

Elaborating on it [math]

$$\min_p \frac{1}{n} \sum_{t=1}^{n} \bar{\ell}(p, J_t) = \min_p \sum_{j=1}^{M} \hat{q}_{j,n} \bar{\ell}(p, j)$$

$$= \min_p \bar{\ell}(p, \hat{q}_n)$$

$$\leq \max_q \min_p \bar{\ell}(p, q) = V$$
Repeated Two–Player Zero–Sum Games

Proof idea.
By definition of Hannan’s consistent strategy [def]

\[
\lim_{n \to \infty} \frac{1}{n} \sum_{t=1}^{n} \ell(I_t, J_t) = \min_{i=1,\ldots,N} \frac{1}{n} \sum_{t=1}^{n} \ell(i, J_t)
\]
Repeated Two–Player Zero–Sum Games

Proof idea.
By definition of Hannan’s consistent strategy \([\text{def}]\)

\[
\lim_{n \to \infty} \sup \frac{1}{n} \sum_{t=1}^{n} \ell(I_t, J_t) = \min_{i=1,\ldots,N} \frac{1}{n} \sum_{t=1}^{n} \ell(i, J_t)
\]

Then

\[
\lim_{n \to \infty} \sup \frac{1}{n} \sum_{t=1}^{n} \ell(I_t, J_t) \leq V
\]
Repetitive Two-Player Zero-Sum Games

Proof idea.

By definition of Hannan’s consistent strategy \[\text{def}\]

\[
\lim_{n \to \infty} \sup \frac{1}{n} \sum_{t=1}^{n} \ell(I_t, J_t) = \min_{i=1,\ldots,N} \frac{1}{n} \sum_{t=1}^{n} \ell(i, J_t)
\]

Then

\[
\lim_{n \to \infty} \sup \frac{1}{n} \sum_{t=1}^{n} \ell(I_t, J_t) \leq V
\]

If we do the same for the other player \[\text{zero–sum game}\]

\[
\lim_{n \to \infty} \sup \frac{1}{n} \sum_{t=1}^{n} \ell(I_t, J_t) \geq V
\]
Repeated Two–Player Zero–Sum Games

**Question:** how fast do they converge to the Nash equilibrium?
Repeated Two–Player Zero–Sum Games

**Question**: how fast do they converge to the Nash equilibrium?  
**Answer**: it depends on the specific algorithm. For EWA(\(\eta\)), we now that

\[
\sum_{t=1}^{n} \ell(I_t, J_t) - \min_{i=1,\ldots,N} \sum_{t=1}^{n} \ell(i, J_t) \leq \frac{\log N}{\eta} + \frac{n\eta}{8} + \sqrt{\frac{n}{2 \log \frac{1}{\delta}}}.
\]
Recurrent Two-Player Zero-Sum Games

Generality of the results
- Players do not know the payoff matrix
Repeatead Two–Player Zero–Sum Games

Generality of the results

- Players do not know the payoff matrix
- Players do not observe the loss of the other player
Repeat Two-Player Zero-Sum Games

Generality of the results

- Players do not know the payoff matrix
- Players do not observe the loss of the other player
- Players do not even observe the action of the other player
Internal Regret and Correlated Equilibria

External (expected) regret

\[ R_n = \sum_{t=1}^{n} \bar{\ell}(\hat{p}_t, y_t) - \min_{i=1,\ldots,N} \sum_{t=1}^{n} \ell(i, y_t) \]

\[ = \max_{i=1,\ldots,N} \sum_{t=1}^{n} \sum_{j=1}^{N} \hat{p}_{j,t} (\ell(j, y_t) - \ell(i, y_t)) \]
Connections to Game Theory

Internal Regret and Correlated Equilibria

External (expected) regret

\[
R_n = \sum_{t=1}^{n} \bar{\ell}(\hat{p}_t, y_t) - \min_{i=1,\ldots,N} \sum_{t=1}^{n} \ell(i, y_t)
\]

\[
= \max_{i=1,\ldots,N} \sum_{t=1}^{n} \sum_{j=1}^{N} \hat{p}_{j,t} (\ell(j, y_t) - \ell(i, y_t))
\]

Internal (expected) regret

\[
R_n^I = \max_{i,j=1,\ldots,N} \sum_{t=1}^{n} \hat{p}_{j,t} (\ell(i, y_t) - \ell(j, y_t))
\]
Internal Regret and Correlated Equilibria

Internal (expected) regret

\[ R_n^I = \max_{i,j=1,\ldots,N} \sum_{t=1}^{n} \hat{p}_{j,t} (\ell(i, y_t) - \ell(j, y_t)) \]

**Intuition:** an algorithm has *small internal regret* if, for each pair of experts \((i,j)\), the learner does not regret of not having followed expert \(j\) each time it followed expert \(i\).
Internal Regret and Correlated Equilibria

Theorem

Given a K–person game with a set of correlated equilibria $C$. If all the players are internal–regret minimizers, then the distance between the empirical distribution of plays and the set of correlated equilibria $C$ converges to 0.
Nash Equilibria in Extensive Form Games

A powerful model for *sequential* games

- Checkers / Chess / Go
- Poker
- Bargaining
- Monitoring
- Patrolling
- ...

Connections to Game Theory
Nash Equilibria in Extensive Form Games

The diagram illustrates a game with two players, where player 1 has two strategies (U and D), and player 2 has two strategies (U' and D'). The payoffs for each player are given at the end of each branch. For example, if player 1 chooses U and player 2 chooses U', the payoff for player 1 is 2 and for player 2 is 1 (since the payoffs are given in a format where the first number is the payoff for the left player and the second number is the payoff for the right player).
Nash Equilibria in Extensive Form Games
Nash Equilibria in Extensive Form Games

No details about the algorithm... but...
Nash Equilibria in Extensive Form Games

No details about the algorithm... but...

Theorem

If player \( k \) selects actions according to the counterfactual regret minimization algorithm, then it achieves a regret

\[
R_{k,T} \leq \# \text{ states} \sqrt{\frac{\# \text{ actions}}{T}}
\]
Nash Equilibria in Extensive Form Games

No details about the algorithm... but...

**Theorem**

*If player \( k \) selects actions according to the counterfactual regret minimization algorithm, then it achieves a regret*

\[
R_{k,T} \leq \# \text{ states} \sqrt{\frac{\# \text{ actions}}{T}}
\]

**Theorem**

*In a two–player zero–sum extensive form game, counterfactual regret minimization algorithms achieves an \( 2\epsilon \)-Nash equilibrium, with*

\[
\epsilon \leq \# \text{ states} \sqrt{\frac{\# \text{ actions}}{T}}
\]
Outline

Mathematical Tools

The General Multi-arm Bandit Problem

The Stochastic Multi-arm Bandit Problem

The Non-Stochastic Multi-arm Bandit Problem

Connections to Game Theory

Other Stochastic Multi-arm Bandit Problems
The Best Arm Identification Problem

Motivating Examples

- Find the best shortest path in a limited number of days
- Maximize the confidence about the best treatment after a finite number of patients
- Discover the best advertisements after a training phase
- ...

The Best Arm Identification Problem

Objective: given a fixed budget $n$, return the best arm $i^* = \arg \max_i \mu_i$ at the end of the experiment
The Best Arm Identification Problem

**Objective**: given a fixed budget $n$, return the best arm $i^* = \arg \max_i \mu_i$ at the end of the experiment.

**Measure of performance**: the probability of error

$$
\mathbb{P}[J_n \neq i^*] \leq \sum_{i=1}^{N} \exp \left( - T_{i,n} \Delta_i^2 \right)
$$
The Best Arm Identification Problem

**Objective:** given a fixed budget \( n \), return the best arm \( i^* = \arg \max_i \mu_i \) at the end of the experiment

**Measure of performance:** the probability of error

\[
P[J_n \neq i^*] \leq \sum_{i=1}^{N} \exp\left(-T_{i,n}\Delta_i^2\right)
\]

**Algorithm idea:** mimic the behavior of the optimal strategy

\[
T_{i,n} = \frac{1}{\sum_{j=1}^{N} \Delta_j^2} n
\]
The Best Arm Identification Problem

The Successive Reject Algorithm

- Divide the budget in $N - 1$ phases. Define
  $$\log(N) = 0.5 + \sum_{i=2}^{N} 1/i$$

$$n_k = \frac{1}{\log K} \frac{n - N}{N + 1 - k}$$
The Best Arm Identification Problem

The Successive Reject Algorithm

- Divide the budget in $N - 1$ phases. Define 
  \[
  \log(N) = 0.5 + \sum_{i=2}^{N} \frac{1}{i}
  \]
  \[
  n_k = \frac{1}{\log K} \frac{n - N}{N + 1 - k}
  \]

- Set of active arms $A_k$ at phase $k$ ($A_1 = \{1, \ldots, N\}$)
The Best Arm Identification Problem

The Successive Reject Algorithm

- Divide the budget in $N - 1$ phases. Define 
  $$(\log(N) = 0.5 + \sum_{i=2}^{N} 1/i)$$

  $$n_k = \frac{1}{\log K} \frac{n - N}{N + 1 - k}$$

- Set of active arms $A_k$ at phase $k$ ($A_1 = \{1, \ldots, N\}$)
- For each phase $k = 1, \ldots, N - 1$
  - For each arm $i \in A_k$, pull arm $i$ for $n_k - n_{k-1}$ rounds
The Best Arm Identification Problem

The Successive Reject Algorithm

- Divide the budget in \( N - 1 \) phases. Define \((\log(N) = 0.5 + \sum_{i=2}^{N} 1/i)\)

\[
n_k = \frac{1}{\log K} \frac{n - N}{N + 1 - k}
\]

- Set of active arms \( A_k \) at phase \( k \) \((A_1 = \{1, \ldots, N\})\)
- For each phase \( k = 1, \ldots, N - 1 \)
  - For each arm \( i \in A_k \), pull arm \( i \) for \( n_k - n_{k-1} \) rounds
  - Remove the worst arm

\[
A_{k+1} = A_k \setminus \arg\min_{i \in A_k} \hat{\mu}_{i,n_k}
\]
The Best Arm Identification Problem

The Successive Reject Algorithm

- Divide the budget in $N - 1$ phases. Define
  \[ \log(N) = 0.5 + \sum_{i=2}^{N} \frac{1}{i} \]
  \[ n_k = \frac{1}{\log K} \frac{n - N}{N + 1 - k} \]

- Set of active arms $A_k$ at phase $k$ ($A_1 = \{1, \ldots, N\}$)
- For each phase $k = 1, \ldots, N - 1$
  - For each arm $i \in A_k$, pull arm $i$ for $n_k - n_{k-1}$ rounds
  - Remove the worst arm
    \[ A_{k+1} = A_k \setminus \arg \min_{i \in A_k} \hat{\mu}_{i,n_k} \]
- Return the only remaining arm $J_n = A_N$
The Best Arm Identification Problem

The Successive Reject Algorithm

Theorem

The successive reject algorithm have a probability of doing a mistake of

\[ P[J_n \neq i^*] \leq \frac{K(K - 1)}{2} \exp \left( - \frac{n - N}{\log NH_2} \right) \]

with \( H_2 = \max_{i=1,\ldots,N} i \Delta_{(i)}^{\frac{2}{2}}. \)
The Best Arm Identification Problem

The UCB-E Algorithm

- Define an exploration parameter $a$
- Compute

$$B_{i,s} = \hat{\mu}_{i,s} + \sqrt{\frac{a}{s}}$$
The Best Arm Identification Problem

The UCB-E Algorithm

- Define an exploration parameter $a$
- Compute

$$B_{i,s} = \hat{\mu}_{i,s} + \sqrt{\frac{a}{s}}$$

- Select

$$I_t = \arg \max_{B_{i,s}}$$
The Best Arm Identification Problem

The UCB-E Algorithm

- Define an exploration parameter $a$
- Compute

$$B_{i,s} = \hat{\mu}_{i,s} + \sqrt{\frac{a}{s}}$$

- Select

$$I_t = \arg \max_{B_{i,s}}$$

- At the end return

$$J_n = \arg \max_i \hat{\mu}_i, T_{i,n}$$
The Best Arm Identification Problem

The UCB-E Algorithm

**Theorem**

The UCB-E algorithm with
\[ a = \frac{25}{36} \frac{n-N}{H_1} \]
has a probability of doing a mistake of

\[ \mathbb{P}[J_n \neq i^*] \leq 2nN \exp \left( - \frac{2a}{25} \right) \]

with
\[ H_1 = \sum_{i=1}^{N} \frac{1}{\Delta_i^2}. \]
The Best Arm Identification Problem
The Active Bandit Problem

Motivating Examples

- $N$ production lines
- The test of the performance of a line is expensive
- We want an accurate estimation of the performance of each production line
The Active Bandit Problem

**Objective**: given a fixed budget \( n \), return the an estimate of the means \( \hat{\mu}_{i,t} \) which is as accurate as possible for all the arms.
The Active Bandit Problem

**Objective**: given a fixed budget $n$, return the an estimate of the means $\hat{\mu}_{i,t}$ which is as accurate as possible for all the arms

**Notice**: Given an arm has a mean $\mu_i$ and a variance $\sigma_i^2$, if it is pulled $T_{i,n}$ times, then

$$L_{i,n} = \mathbb{E}[(\hat{\mu}_{i,T_{i,n}} - \mu_i)^2] = \frac{\sigma_i^2}{T_{i,n}}$$
The Active Bandit Problem

Objective: given a fixed budget $n$, return the estimate of the means $\hat{\mu}_{i,t}$ which is as accurate as possible for all the arms.

Notice: Given an arm has a mean $\mu_i$ and a variance $\sigma_i^2$, if it is pulled $T_{i,n}$ times, then

$$L_{i,n} = \mathbb{E}[(\hat{\mu}_{i,T_{i,n}} - \mu_i)^2] = \frac{\sigma_i^2}{T_{i,n}}$$

$$L_n = \max_i L_{i,n}$$
The Active Bandit Problem

**Problem**: what are the number of pulls \((T_1,n, \ldots, T_N,n)\) (such that \(\sum T_{i,n} = n\)) which minimizes the loss?

\[
(T_1^*, n, \ldots, T_N^*, n) = \arg \min_{(T_1,n, \ldots, T_N,n)} L_n
\]
The Active Bandit Problem

**Problem:** what are the number of pulls \((T_{1,n}, \ldots, T_{N,n})\) (such that \(\sum T_{i,n} = n\)) which minimizes the loss?

\[
(T_{1,n}^*, \ldots, T_{N,n}^*) = \arg \min_{(T_{1,n}, \ldots, T_{N,n})} L_n
\]

**Answer**

\[
T_{i,n}^* = \frac{\sigma_i^2}{\sum_{j=1}^N \sigma_j^2} n
\]
The Active Bandit Problem

Problem: what are the number of pulls \((T_{1,n}, \ldots, T_{N,n})\) (such that \(\sum T_{i,n} = n\)) which minimizes the loss?

\[
(T_{1,n}^*, \ldots, T_{N,n}^*) = \arg \min_{(T_{1,n}, \ldots, T_{N,n})} L_n
\]

Answer

\[
T_{i,n}^* = \frac{\sigma_i^2 \sum_{j=1}^{N} \sigma_j^2 n}{\sum_{j=1}^{N} \sigma_j^2 n}
\]

\[
L_n^* = \frac{\sum_{i=1}^{N} \sigma_i^2}{n} = \frac{\Sigma}{n}
\]
The Active Bandit Problem

Objective: given a fixed budget $n$, return the an estimate of the means $\hat{\mu}_{i,t}$ which is as accurate as possible for all the arms
The Active Bandit Problem

**Objective:** given a fixed budget $n$, return the an estimate of the means $\hat{\mu}_{i,t}$ which is as accurate as possible for all the arms

**Measure of performance:** the regret on the quadratic error

$$R_n(A) = \max_i L_n(A) - \sum_{i=1}^{N} \frac{\sigma_i^2}{n}$$
The Active Bandit Problem

**Objective:** given a fixed budget $n$, return the estimate of the means $\hat{\mu}_{i,t}$ which is as accurate as possible for all the arms

**Measure of performance:** the regret on the quadratic error

$$R_n(\mathcal{A}) = \max_i L_n(\mathcal{A}) - \frac{\sum_{i=1}^{N} \sigma_i^2}{n}$$

**Algorithm idea:** mimic the behavior of the optimal strategy

$$T_{i,n} = \frac{\sigma_i^2}{\sum_{j=1}^{N} \sigma_j^2} n = \lambda_i n$$
The Active Bandit Problem

An UCB–based strategy
At each time step $t = 1, \ldots, n$

- Estimate

$$\hat{\sigma}^2_{i, T_i, t-1} = \frac{1}{T_{i, t-1}} \sum_{s=1}^{T_{i, t-1}} X_{s, i}^2 - \hat{\mu}^2_{i, T_{i, t-1}}$$

- Compute

$$B_{i, t} = \frac{1}{T_{i, t-1}} \left( \hat{\sigma}^2_{i, T_i, t-1} + 5 \sqrt{\frac{\log 1/\delta}{2T_{i, t-1}}} \right)$$

- Pull arm

$$I_t = \arg \max B_{i, t}$$
The Active Bandit Problem

Theorem

The UCB–based algorithm achieves a regret

\[ R_n(A) \leq \frac{98 \log(n)}{n^{3/2}} \chi^{5/2}_{\min} + O\left(\frac{\log n}{n^2}\right) \]
The Active Bandit Problem

Theorem

The UCB–based algorithm achieves a regret

\[ R_n(A) \leq \frac{98 \log(n)}{n^{3/2}} \lambda_{\min}^{5/2} + O\left(\frac{\log n}{n^2}\right) \]
The Contextual Linear Bandit Problem

Motivating Examples

- Different users may have different preferences
- The set of available news may change over time
- We want to minimise the regret w.r.t. the best news for each user
The Contextual Linear Bandit Problem

The problem: at each time $t = 1, \ldots, n$

- User $u_t$ arrives and a set of news $A_t$ is provided
- The user $u_t$ together with a news $a \in A_t$ are described by a feature vector $x_{t,a}$
- The learner chooses a news $a_t$ and receives a reward $r_{t,a_t}$

The optimal news: at each time $t = 1, \ldots, n$, the optimal news is

$$a^*_t = \arg \max_{a \in A_t} \mathbb{E}[r_{t,a}]$$

The regret:

$$R_n = \mathbb{E}\left[ \sum_{t=1}^{n} r_{t,a^*_t} \right] - \mathbb{E}\left[ \sum_{t=1}^{n} r_{t,a_t} \right]$$
The Contextual Linear Bandit Problem

**The linear assumption:** the reward is a linear combination between the context and an unknown parameter vector

\[ \mathbb{E}[r_{t,a} | x_{t,a}] = x_{t,a}^\top \theta_a \]
The Contextual Linear Bandit Problem

The linear regression estimate:

- \( \mathcal{T}_a = \{ t : a_t = a \} \)
- Construct the design matrix of all the contexts observed when action \( a \) has been taken \( D_a \in \mathbb{R}^{|\mathcal{T}_a| \times d} \)
- Construct the reward vector of all the rewards observed when action \( a \) has been taken \( c_a \in \mathbb{R}^{|\mathcal{T}_a|} \)
- Estimate \( \theta_a \) as

\[
\hat{\theta}_a = (D_a^\top D_a + I)^{-1} D_a^\top c_a
\]
The Contextual Linear Bandit Problem

Optimism in face of uncertainty: the LinUCB algorithm

- Chernoff-Hoeffding in this case becomes

\[
|x_t, a^\top \hat{\theta}_a - \mathbb{E}[r_{t,a}|x_t,a]| \leq \alpha \sqrt{x_t, a^{\top}(D_a^\top D_a + I)^{-1}x_t, a}
\]

- and the UCB strategy is

\[
a_t = \arg\max_{a \in \mathcal{A}_t} x_t, a^\top \hat{\theta}_a + \alpha \sqrt{x_t, a^{\top}(D_a^\top D_a + I)^{-1}x_t, a}
\]
The Contextual Linear Bandit Problem

The evaluation problem

- Online evaluation: too expensive
- Offline evaluation: how to use the logged data?
The Contextual Linear Bandit Problem

Evaluation from logged data

- Assumption 1: contexts and rewards are i.i.d. from a stationary distribution

\[(x_1, \ldots, x_K, r_1, \ldots, r_K) \sim D\]

- Assumption 2: the logging strategy is random
The Contextual Linear Bandit Problem

**Evaluation from logged data**: given a bandit strategy \( \pi \), a desired number of samples \( T \), and a (infinite) stream of data

---

**Algorithm 3 Policy_Evaluator.**

0: Inputs: \( T > 0 \); policy \( \pi \); stream of events
1: \( h_0 \leftarrow \emptyset \) \{An initially empty history\}
2: \( R_0 \leftarrow 0 \) \{An initially zero total payoff\}
3: **for** \( t = 1, 2, 3, \ldots, T \) **do**
4: \hspace{1em} **repeat**
5: \hspace{2em} Get next event \( (x_1, \ldots, x_K, a, r_a) \)
6: \hspace{2em} **until** \( \pi(h_{t-1}, (x_1, \ldots, x_K)) = a \)
7: \hspace{2em} \( h_t \leftarrow \text{CONCATENATE}(h_{t-1}, (x_1, \ldots, x_K, a, r_a)) \)
8: \hspace{2em} \( R_t \leftarrow R_{t-1} + r_a \)
9: **end for**
10: Output: \( R_T / T \)
Bibliography I
Reinforcement Learning

Alessandro Lazaric
alessandro.lazaric@inria.fr
sequel.lille.inria.fr