Sequential Decision Making

Lecture 3 : Beyond Classical Bandits

Emilie Kaufmann

M2 Data Science, 2022/2023

Recap from last class

Several important ideas to tackle the exploration/exploitation challenge in a simple multi-armed bandit model with independent arms :

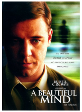
- Explore then Commit
- ε-greedy
- Optimistic algorithms : Upper Confidence Bounds strategies
- Bayesian algorithms : Thompson Sampling

Some of these can be extended to more realistic **structured** models that are suited for different applications.

Outline

Contextual Bandits

Example : movie recommendation



What movie should Netflix recommend to a particular user, given the ratings provided by previous users ?

➜ to make good recommendation, we should take into account the characteristics of the movies / users

Contextual bandit problem : at time t

- a context c_t is observed
- an arm A_t is chosen
- a reward R_t that depends on c_t, A_t is received.

Mixing bandits and regression models

A **contextual bandit model** incorporates two components :

- a sequential interaction protocol :
  pick an arm, receive a reward
- a regression model for the dependency between context and reward

Mixing bandits and regression models

A (stochastic) **contextual bandit model** incorporates two components :

- a sequential interaction protocol :
  pick an arm, receive a (random) reward
- a regression model for the dependency between context and reward

Mixing bandits and regression models

A (stochastic) **contextual bandit model** incorporates two components :

- a sequential interaction protocol :
  pick an arm, receive a (random) reward
- a regression model for the dependency between context and reward

General stochastic contextual bandit model

In each round t, the agent

- observes a context $c_t \in \mathcal{C}$ *(user characteristics)*
- selects an arm $A_t \in \mathcal{A}_t$ *(an item out of a possibly changing pool)*
- the agent receives a reward

$$r_t = f_{A_t}(c_t) + \varepsilon_t$$

where ε_t is an independent noise : $\mathbb{E}[\varepsilon_t] = 0$.

$f_a : \mathcal{C} \to \mathbb{R}$ maps a context c to the average reward of arm a, $f_a(c)$

Examples

Example 1

- user t : descriptor $c_t \in \mathbb{R}^p$
- item a : descriptor $\theta_a \in \mathbb{R}^p$

$$r_t = \theta_{A_t}^\top c_t + \varepsilon_t$$

Linear function $f_a(c) = \theta_a^\top c$

Observation : if $\mathcal{A}_t = \{1, \dots, K\}$ is a fixed set of items

- the model is parameterized by $\theta_1, \theta_2, \dots, \theta_K \in (\mathbb{R}^p)^K$
- it can also be rewritten $r_t = \theta_\star^\top (x_{t,A_t}) + \varepsilon_t$ with

$$\theta_\star = \begin{pmatrix} \theta_1 \\ \dots \\ \theta_a \\ \dots \\ \theta_K \end{pmatrix} \in \mathbb{R}^{p \times K}, \qquad x_{t,a} = \begin{pmatrix} 0 \\ \dots \\ c_t \\ \dots \\ 0 \end{pmatrix} \in \mathbb{R}^{p \times K}$$

$x_{t,a}$: feature vector for the user-item pair (t, a)

Examples

Example 2

- user t : descriptor $c_t \in \mathbb{R}^p$
- item a : descriptor $x_a \in \mathbb{R}^{p'}$

➜ build a user-item feature vector for (t, a) : $x_{t,a} \in \mathbb{R}^d$

(feature engineering)

$$r_t = \theta_\star^\top x_{t,A_t} + \varepsilon_t$$

Observation :

- the model is parameterized by $\theta_\star \in \mathbb{R}^d$
- in each round t, the user-item feature vectors belong to the set

$$\mathcal{X}_t = \{x_{t,a}, a \in \mathcal{A}_t\} \subseteq \mathbb{R}^d$$

- picking an arm $a \leftrightarrow$ picking a feature vector $x_t \in \mathcal{X}_t$

$$r_t = \theta_\star^\top x_t + \varepsilon_t$$

Examples

Example 2

- user t : descriptor $c_t \in \mathbb{R}^p$
- item a : descriptor $x_a \in \mathbb{R}^{p'}$

➜ build a user-item feature vector for (t, a) : $x_{t,a} \in \mathbb{R}^d$

(feature engineering)

$$r_t = \theta_\star^\top x_{t,A_t} + \varepsilon_t$$

Observation :

- the model is parameterized by $\theta_\star \in \mathbb{R}^d$
- in each round t, the user-item feature vectors belong to the set

$$\mathcal{X}_t = \{x_{t,a}, a \in \mathcal{A}_t\} \subseteq \mathbb{R}^d$$

- picking an arm $a \leftrightarrow$ picking a feature vector $x_t \in \mathcal{X}_t$

$$r_t = f_\star(x_t) + \varepsilon_t$$

Two formulations

Contextual MAB, version 1

In each round t, the agent

- observes a context $c_t \in \mathcal{C}$
- selects an arm $A_t \in \mathcal{A}_t$ *(set of arm can vary in each round)*
- the agent receives a reward $r_t = f_{A_t}(c_t) + \varepsilon_t$

Unknown : regression functions (f_a) for all possible arm a

Contextual MAB (more general)

In each round t, the agent

- is given a set of *arms* $\mathcal{X}_t$ *(can be different in each round)*
- selects an *arm* $x_t \in \mathcal{X}_t$
- the agent receives a reward $r_t = f_\star(x_t) + \varepsilon_t$

Unknown : regression function $f_\star$

Two formulations

Contextual MAB, version 1

In each round t, the agent

- observes a context $c_t \in \mathcal{C}$
- selects an arm $A_t \in \mathcal{A}_t$ *(set of arm can vary in each round)*
- the agent receives a reward $r_t = f_{A_t}(c_t) + \varepsilon_t$

Unknown : regression functions (f_a) for all possible arm a

Contextual MAB (more general)

In each round t, the agent

- is given a set of *arms* $\mathcal{X}_t$ *(can be different in each round)*
- selects an *arm* $x_t \in \mathcal{X}_t$
- the agent receives a reward $r_t = f_\star(x_t) + \varepsilon_t$

Unknown : regression function $f_\star$

→ **Goal :** learn the unknown function $f_\star$... while maximizing rewards !

Outline

Contextual linear bandits

In each round t, the agent

- receives a (finite) set of arms $\mathcal{X}_t \subseteq \mathbb{R}^d$
- chooses an arm $x_t \in \mathcal{X}_t$
- gets a reward $r_t = \theta_\star^\top x_t + \varepsilon_t$

where

- $\theta_\star \in \mathbb{R}^d$ is an unknown regression vector
- ε_t is a centered noise, independent from past data

Assumption : σ^2- sub-Gaussian noise

$$\forall \lambda \in \mathbb{R}, \ \mathbb{E}\left[e^{\lambda X}\right] \leq e^{\frac{\lambda^2 \sigma^2}{2}}$$

e.g., Gaussian noise, bounded noise.

Contextual linear bandits

In each round t, the agent

- receives a (finite) set of arms $\mathcal{X}_t \subseteq \mathbb{R}^d$
- chooses an arm $x_t \in \mathcal{X}_t$
- gets a reward $r_t = \theta_\star^\top x_t + \varepsilon_t$

where

- $\theta_\star \in \mathbb{R}^d$ is an unknown regression vector
- ε_t is a centered noise, independent from past data

(Pseudo)-regret for contextual bandit

maximizing expected total reward $\leftrightarrow$ minimizing the expectation of

$$R_T(\mathcal{A}) = \sum_{t=1}^{T} \left(\max_{x \in \mathcal{X}_t} \theta_\star^\top x - \theta_\star^\top x_t \right)$$

➜ in each round, comparison to a possibly different optimal action !

Tools

Algorithms will rely on estimates / confidence regions / posterior distributions for $\theta_\star \in \mathbb{R}^d$.

- design matrix (with regularization parameter $\lambda > 0$)

$$B_t^\lambda = \lambda I_d + \sum_{s=1}^{t} x_s x_s^\top$$

- regularized least-square estimate

$$\hat{\theta}_t^\lambda = \left(B_t^\lambda\right)^{-1} \left(\sum_{s=1}^{t} r_t x_t\right)$$

Recap from lecture 1 : easy online update !

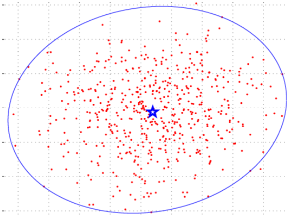
- estimate of the expected reward of an arm $x \in \mathbb{R}^d$: $x^\top \hat{\theta}_t^\lambda$

➜ sufficient for Follow the Leader, but not for smarter algorithms !

Outline

How to build (tight) confidence interval on the mean rewards ?

Idea : rely on a confidence ellispoid around $\hat{\theta}_t^\lambda$



$$\theta_\star \in \left\{ \theta \in \mathbb{R}^d : \|\theta - \hat{\theta}_t^\lambda\|_A \leq \beta_t \right\}$$

Why ? For all invertible matrix positive semi-definite matrix A,

$$\forall x \in \mathbb{R}^d, \quad \left| x^\top \theta_\star - x^\top \hat{\theta}_t^\lambda \right| \leq \|x\|_{A^{-1}} \left\| \theta_\star - \hat{\theta}_t^\lambda \right\|_A$$

$\|x\|_A = \sqrt{x^\top A x}$

How to build (tight) confidence interval on the mean rewards ?

Wanted : $\theta_\star \in \left\{\theta \in \mathbb{R}^d : \|\theta - \hat{\theta}_t^\lambda\|_A \leq \beta_t\right\}$

Example of threshold [Abbasi-Yadkori et al., 2011]

Assuming that the noise ε_t is σ^2-sub-Gaussian, and that for all t and $x \in \mathcal{X}_t$, $\|x\| \leq L$, we have

$$\mathbb{P}\left(\exists t \in \mathbb{N}^\star : \|\theta_\star - \hat{\theta}_t^\lambda\|_{B_t^\lambda} > \beta(t,\delta)\right) \leq \delta$$

with $\beta(t,\delta) = \sigma\sqrt{2\log(1/\delta) + d\log\left(1 + t\frac{L}{d\lambda}\right)} + \sqrt{\lambda}\|\theta_\star\|$.

➜ Letting

$$C_t(\delta) = \left\{\theta \in \mathbb{R}^d : \|\theta - \hat{\theta}_t^\lambda\|_{B_t^\lambda} \leq \beta(t,\delta)\right\},$$

one has $\mathbb{P}\left(\forall t \in \mathbb{N}, \theta_\star \in C_t(\delta)\right) \geq 1 - \delta$.

A Lin-UCB algorithm

Consequence :

$$\mathbb{P}\left(\forall t \in \mathbb{N}^*, \forall x \in \mathcal{X}_{t+1}, \quad \underbrace{x^\top \theta_\star}_{\substack{\text{unknown mean} \\ \text{of arm } x}} \quad \leq \underbrace{x^\top \hat{\theta}_t^\lambda + \|x\|_{(B_t^\lambda)^{-1}} \beta(t,\delta)}_{\text{Upper Confidence Bound}}\right) \geq 1 - \delta.$$

One can assign to each arm $x \in \mathcal{X}_{t+1}$

$$\mathrm{UCB}_x(t) = \underbrace{x^\top \hat{\theta}_t^\lambda}_{\substack{\text{empirical mean} \\ \text{(exploitation term)}}} + \underbrace{\|x\|_{(B_t^\lambda)^{-1}} \beta(t,\delta)}_{\text{exploration bonus}}$$

Lin-UCB

In each round $t+1$, the algorithm selects

$$x_{t+1} = \underset{x \in \mathcal{X}_{t+1}}{\mathrm{argmax}} \left[x^\top \hat{\theta}_t^\lambda + \|x\|_{(B_t^\lambda)^{-1}} \beta(t,\delta)\right]$$

(many algorithms of this style, with different choices of $\beta(t,\delta)$)

Theoretical guarantees

We want to bound the pseudo-regret

$$R_T(\text{Lin-UCB}) = \sum_{t=1}^{T} \left(\max_{x \in \mathcal{X}_t} \theta_\star^\top x - \theta_\star^\top x_t \right)$$

or its expectation, the regret $\mathcal{R}_T(\text{Lin-UCB}) = \mathbb{E}[R_T(\text{Lin-UCB})]$.

Lemma

One can prove that, with probability larger than $1 - \delta$,

$$\forall T \in \mathbb{N}^*, R_T(\text{Lin-UCB}) \leq C\beta(T, \delta)\sqrt{dT \log(T)}$$

- with the choice of $\beta(t, \delta)$ presented before, with high probability

$$R_T(\text{Lin-UCB}) = \mathcal{O}(d\sqrt{T}\log(T) + \sqrt{dT\log(T)\log(1/\delta)})$$

- choosing $\delta = 1/T$, $\mathcal{R}_T(\text{Lin-UCB}) = \mathcal{O}(d\sqrt{T}\log(T))$

Outline

1. Contextual Bandits
2. Solving Linear Bandits
   - Lin-UCB
   - Linear Thompson Sampling
3. Other variants of the classical MAB
4. Beyond maximizing rewards

A Bayesian view on Linear Regression

Bayesian model :

- likelihood : $r_t = \theta_\star^\top x_t + \varepsilon_t$
- prior : $\theta_\star \sim \mathcal{N}(0, \kappa^2 \mathrm{I}_d)$

Assuming further that the noise is Gaussian : $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$, the posterior distribution of $\theta_\star$ has a closed form :

$$\theta_\star | x_1, r_1, \dots, x_t, r_t \ \sim \ \mathcal{N}\left(\hat{\theta}_t^\lambda, \sigma^2 \left(B_t^\lambda\right)^{-1}\right)$$

with

- $B_t^\lambda = \lambda \mathrm{I}_d + \sum_{s=1}^t x_s x_s^\top$
- $\hat{\theta}_t^\lambda = \left(B_t^\lambda\right)^{-1} \left(\sum_{s=1}^t r_s x_s\right)$ is the regularized least square estimate

with a regularization parameter $\lambda = \frac{\sigma^2}{\kappa^2}$.

Thompson Sampling for Linear Bandits

Recall the Thompson Sampling principle :

> "draw a possible model from the posterior distribution and act optimally in this sampled model"

Thompson Sampling in linear bandits

In each round $t+1$,

$$\begin{aligned} \tilde{\theta}_t &\sim \mathcal{N}\left(\hat{\theta}_t^{\lambda}, \sigma^2 \left(B_t^{\lambda}\right)^{-1}\right) \\ x_{t+1} &= \underset{x \in \mathcal{X}_{t+1}}{\operatorname{argmax}} \; x^{\top} \tilde{\theta}_t \end{aligned}$$

Numerical complexity : one need to draw a sample from a multivariate Gaussian distribution, e.g.

$$\tilde{\theta}_t = \hat{\theta}_t^{\lambda} + \sigma \left(B_t^{\lambda}\right)^{-1/2} X$$

where X is a vector with d independent $\mathcal{N}(0,1)$ entries.

Theoretical guarantees

[Agrawal and Goyal, 2013] analyze a *variant* of Thompson Sampling using some "posterior inflation" :

$$\begin{aligned} \tilde{\theta}_t &\sim \mathcal{N}\left(\hat{\theta}_t^1, v^2 \left(B_t^1\right)^{-1}\right) \\ x_{t+1} &= \underset{x \in \mathcal{X}_{t+1}}{\operatorname{argmax}} \ x^\top \tilde{\theta}_t \end{aligned}$$

where $v = \sigma\sqrt{9d\ln(T/\delta)}$.

Theorem

If the noise is σ^2-sub-Gaussian, the above algorithm satisfies

$$\mathbb{P}\left(R_T(\mathrm{TS}) = \mathcal{O}\left(d^{3/2}\sqrt{T}\left[\ln(T) + \sqrt{\ln(T)\ln(1/\delta)}\right]\right)\right) \geq 1 - \delta.$$

- slightly worse than Lin-UCB... how about in practice ?
- do we need the posterior inflation ?

Beyond linear bandits

Depending on the application, other parametric models may be better suited than the simple linear model, for example the logistic model.

$$\begin{aligned} \mathbb{P}\left(r_t = 1 | x_t\right) &= \frac{1}{1 + e^{-\theta_\star^\top x_t}} \\ \mathbb{P}\left(r_t = 0 | x_t\right) &= \frac{e^{-\theta_\star^\top x_t}}{1 + e^{-\theta_\star^\top x_t}} \end{aligned}$$

e.g., clic / no-clic on an add depending on a user/add feature $x_t \in \mathbb{R}^d$

- [Filippi et al., 2010] : first UCB style algorithm for Generalized Linear Bandit models
- Thompson Sampling for logistic bandits [Dumitrascu et al., 2018]
- going further : UCB/TS for neural bandits !

Outline

Many possible structures

$\mathcal{X}$-armed bandits : $\mathcal{X}_t = \mathcal{X}$ arbitrary metric space

$$r_t = f_\star(x_t) + \varepsilon_t$$

with non-parametric assumption on $f_\star$.

Examples :

- $f_\star$ is a Lipschitz function :

  $$|f_\star(x) - f_\star(y)| \leq Ld(x, y)$$

  where d is a metric on $\mathcal{X}$.

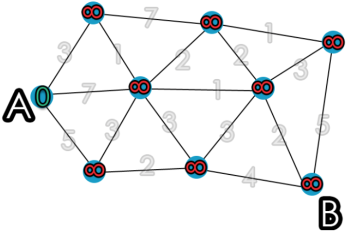
  [Bubeck et al., 2011b]
- $f_\star$ is a unimodal function
- $f_\star$ is drawn from a Gaussian process prior

  [Srinivas et al., 2010]
- ...

Beyond one arm : Combinatorial bandits

classical bandit : one arm is selected in each round
combinatorial bandit : possibility to select a group of arms (action)

e.g.,[Chen et al., 2013]



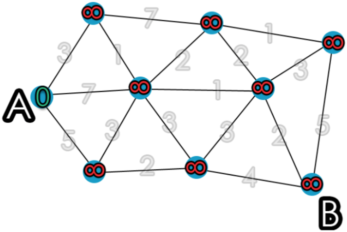
Example :

- arms : edges in a graph
- actions : paths from A to B
- reward : some function of the edges's rewards in the chosen path *(e.g. - (total travelling distance))*

Beyond one arm : Combinatorial bandits

classical bandit : one arm is selected in each round
combinatorial bandit : possibility to select a group of arms (action)

e.g.,[Chen et al., 2013]



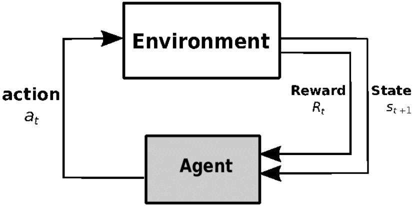
Combinatorial bandit : $\text{Actions} \subseteq \mathcal{P}(\{1, \ldots, K\})$.
In round t, the agent

- selects an action $\text{Act}_t \in \text{Actions}$
- a reward $r_{a,t}$ is generated for every arm $a \in \text{Act}_t$
- the agent receives as a reward $\sum_{a \in \text{Act}_t} r_{a,t}$ (or some other function)

Beyond one state : Reinforcement Learning

In most bandit models, the agent repeatedly faces the same set of actions (or at least the set of available actions in round does not depend on the past decisions).

➜ no longer true in **reinforcement learning**, in which an action also triggers a transition to a new state



more on this in the next lectures

Outline

Bandits without rewards ?

$\mathcal{B}(\mu_1)$ $\mathcal{B}(\mu_2)$ $\mathcal{B}(\mu_3)$ $\mathcal{B}(\mu_4)$ $\mathcal{B}(\mu_5)$

For the t-th patient in a clinical study,

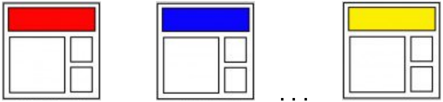
- chooses a treatment A_t
- observes a response $X_t \in \{0, 1\} : \mathbb{P}(X_t = 1) = \mu_{A_t}$

Maximize rewards $\leftrightarrow$ cure as many patients as possible

Alternative goal : identify as quickly as possible the best treatment
(without trying to cure patients during the study)

Bandits without rewards ?

Probability that some version of a website generates a conversion :



μ_1 μ_2 μ_K

Best version : $a_\star = \underset{a=1,\dots,K}{\mathrm{argmax}}\ \mu_a$

Sequential protocol : for the t-th visitor :

- display version A_t
- observe conversion indicator $X_t \sim \mathcal{B}(\mu_{A_t})$.

Maximize rewards $\leftrightarrow$ maximize the number of conversions

Alternative goal : identify the best version
(without trying to maximize conversions during the test)

A Pure Exploration Problem

Goal : identify an arm with mean close to $\mu_\star$ as quickly and accurately as possible $\simeq$ identify

$$a_\star = \underset{a=1,\dots,K}{\mathrm{argmax}} \ \mu_a.$$

Algorithm : made of three components :

- ➜ sampling rule : A_t (arm to explore)
- ➜ recommendation rule : B_t (current guess for the best arm)
- ➜ stopping rule τ (when do we stop exploring ?)

Probability of error

The probability of error after n rounds is

$$p_\nu(T) = \mathbb{P}_\nu\left(B_T \neq a_\star\right).$$

A Pure Exploration Problem

Goal : identify an arm with mean close to $\mu_\star$ as quickly and accurately as possible $\simeq$ identify

$$a_\star = \underset{a=1,\dots,K}{\mathrm{argmax}} \ \mu_a.$$

Algorithm : made of three components :

- ➜ sampling rule : A_t (arm to explore)
- ➜ recommendation rule : B_t (current guess for the best arm)
- ➜ stopping rule τ (when do we stop exploring ?)

Simple regret [Bubeck et al., 2011a]

The simple regret after n rounds is

$$r_\nu(n) = \mu_\star - \mu_{B_n}.$$

A Pure Exploration Problem

Goal : identify an arm with mean close to $\mu_\star$ as quickly and accurately as possible $\simeq$ identify

$$a_\star = \underset{a=1,\dots,K}{\operatorname{argmax}} \ \mu_a.$$

Algorithm : made of three components :

- ➜ sampling rule : A_t (arm to explore)
- ➜ recommendation rule : B_t (current guess for the best arm)
- ➜ stopping rule τ (when do we stop exploring ?)

Simple regret [Bubeck et al., 2011a]

The simple regret after n rounds is

$$r_\nu(n) = \mu_\star - \mu_{B_n}.$$

$$\Delta_{\min} p_\nu(T) \leq \mathbb{E}_\nu[r_\nu(T)] \leq \Delta_{\max} p_\nu(T)$$

Several objectives

Algorithm : made of three components :

➜ sampling rule : A_t (arm to explore)

➜ recommendation rule : B_t (current guess for the best arm)

➜ stopping rule τ (when do we stop exploring ?)

- **Objectives studied in the literature :**

Fixed-budget setting	Fixed-confidence setting
input : budget T	input : risk parameter δ (tolerance parameter ϵ)
$\tau = T$ minimize $\mathbb{P}(B_T \neq a_\star)$ or $\mathbb{E}[r_T(\nu)]$	minimize $\mathbb{E}[\tau]$ $\mathbb{P}(B_\tau \neq a_\star) \leq \delta$ or $\mathbb{P}(r_\nu(\tau) > \epsilon) \leq \delta$
[Bubeck et al., 2011a] [Audibert et al., 2010]	[Even-Dar et al., 2006]

Can we use UCB ?

Context : bounded rewards (ν_a supported in $[0,1]$)

We know good algorithms to maximize rewards, for example UCB(α)

$$A_{t+1} = \underset{a=1,\dots,K}{\text{argmax}} \ \hat{\mu}_a(t) + \sqrt{\frac{\alpha \ln(t)}{N_a(t)}}$$

- How good is it for best arm identification ?

Can we use UCB ?

Context : bounded rewards (ν_a supported in $[0,1]$)

We know good algorithms to maximize rewards, for example UCB(α)

$$A_{t+1} = \underset{a=1,\dots,K}{\operatorname{argmax}} \ \hat{\mu}_a(t) + \sqrt{\frac{\alpha \ln(t)}{N_a(t)}}$$

- How good is it for best arm identification ?

Possible recommendation rules :

Empirical Best Arm (EBA)	$B_t = \operatorname{argmax}_a \ \hat{\mu}_a(t)$
Most Played Arm (MPA)	$B_t = \operatorname{argmax}_a \ N_a(t)$
Empirical Distribution of Plays (EDP)	$B_t \sim p_t$, where $p_t = \left(\frac{N_1(t)}{t}, \dots, \frac{N_K(t)}{t}\right)$

[Bubeck et al., 2011a]

Can we use UCB ?

Context : bounded rewards (ν_a supported in $[0, 1]$)

We know good algorithms to maximize rewards, for example UCB(α)

$$A_{t+1} = \underset{a=1,\dots,K}{\operatorname{argmax}} \ \hat{\mu}_a(t) + \sqrt{\frac{\alpha \ln(t)}{N_a(t)}}$$

- How good is it for best arm identification ?

Possible recommendation rules :

Empirical Best Arm (EBA)	$B_t = \operatorname{argmax}_a \ \hat{\mu}_a(t)$
Most Played Arm (MPA)	$B_t = \operatorname{argmax}_a \ N_a(t)$
Empirical Distribution of Plays (EDP)	$B_t \sim p_t$, where $p_t = \left(\frac{N_1(t)}{t}, \dots, \frac{N_K(t)}{t}\right)$

[Bubeck et al., 2011a]

Can we use UCB ?

- **UCB + Empirical Distribution of Plays**

$$\begin{aligned}
\mathbb{E}\left[r_\nu(T)\right] &= \mathbb{E}\left[\mu_\star - \mu_{B_T}\right] = \mathbb{E}\left[\sum_{b=1}^{K}(\mu_\star - \mu_b)\mathbb{1}_{(B_T=b)}\right] \\
&= \mathbb{E}\left[\sum_{b=1}^{K}(\mu_\star - \mu_b)\mathbb{P}(B_T = b|\mathcal{F}_T)\right] \\
&= \mathbb{E}\left[\sum_{b=1}^{K}(\mu_\star - \mu_b)\frac{N_b(T)}{T}\right] \\
&= \frac{1}{T}\sum_{b=1}^{K}(\mu_\star - \mu_b)\mathbb{E}[N_b(T)] \\
&= \frac{\mathcal{R}_\nu(T)}{T}.
\end{aligned}$$

➜ a conversion from cumulative regret to simple regret !

Can we use UCB ?

- **UCB + Empirical Distribution of Plays**

$$\mathbb{E}\left[r_\nu\left(\texttt{UCB}(\alpha), T\right)\right] \leq \frac{\mathcal{R}_\nu(\texttt{UCB}(\alpha), T)}{T} \leq \frac{C(\nu)\ln(T)}{T}$$

Can we use UCB ?

- **UCB + Empirical Distribution of Plays**

$$\mathbb{E}\left[r_\nu\left(\texttt{UCB}(\alpha), T\right)\right] \leq \frac{\mathcal{R}_\nu(\texttt{UCB}(\alpha), T)}{T} \leq C\sqrt{\frac{K\ln(T)}{T}}$$

Can we use UCB ?

- **UCB + Empirical Distribution of Plays**

$$\mathbb{E}\left[r_\nu\left(\texttt{UCB}(\alpha), T\right)\right] \leq \frac{\mathcal{R}_\nu(\texttt{UCB}(\alpha), T)}{T} \leq C\sqrt{\frac{K\ln(T)}{T}}$$

- vs. **Uniform Sampling**

The simple regret or the uniform strategy decays exponentially :

$$\mathbb{E}_\nu\left[r_\nu\left(\texttt{Unif}, T\right)\right] \leq (K-1)\Delta_{\max}\exp\left(-\frac{1}{2}\frac{T}{K}\Delta_{\min}^2\right)$$

➜ UCB does not provably outperform uniform sampling...

Fixed Budget : Sequential Halving

Input : total number of plays T

Idea : split the budget in $\log_2(K)$ phases of equal length, eliminate the worst half of the remaining arms after each phase.

Initialisation : $S_0 = \{1, \dots, K\}$;
For $r = 0$ **to** $\lceil \ln_2(K) \rceil - 1$, **do**
sample each arm $a \in S_r$ $t_r = \left\lfloor \frac{T}{|S_r| \lceil \log_2(K) \rceil} \right\rfloor$ times ;
let $\hat{\mu}_a^r$ be the empirical mean of arm a ;
let S_{r+1} be the set of $\lceil |S_r|/2 \rceil$ arms with largest $\hat{\mu}_a^r$
Output : B_T the unique arm in $S_{\lceil \log_2(K) \rceil}$

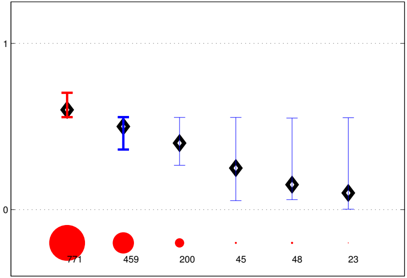
Theorem [Karnin et al., 2013]

Letting $H_2(\nu) = \max_{a \neq a_\star} a\Delta_{[a]}^{-2}$, for any bounded bandit instance,

$$\mathbb{P}_\nu\left(B_T \neq a_\star\right) \leq 3 \log_2(K) \exp\left(-\frac{T}{8 \log_2(K) H_2(\nu)}\right).$$

Fixed Budget : LUCB

$$\mathcal{I}_a(t) = [\mathrm{LCB}_a(t), \mathrm{UCB}_a(t)].$$



- At round t, draw

$$B_t = \underset{b}{\mathrm{argmax}} \ \hat{\mu}_b(t)$$
$$C_t = \underset{c \neq B_t}{\mathrm{argmax}} \ \mathrm{UCB}_c(t)$$

- Stop at round t if

$$\mathrm{LCB}_{B_t}(t) > \mathrm{UCB}_{C_t}(t) - \epsilon$$

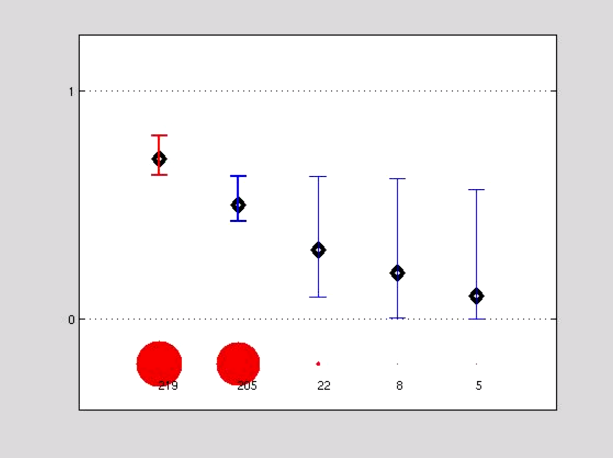
Theorem [Kalyanakrishnan et al., 2012]

For well-chosen confidence intervals, $\mathbb{P}_\nu(\mu_{B_\tau} > \mu_\star - \epsilon) \geq 1 - \delta$ and

$$\mathbb{E}[\tau_\delta] = O\left(\left[\frac{1}{\Delta_2^2 \vee \epsilon^2} + \sum_{a=2}^{K} \frac{1}{\Delta_a^2 \vee \epsilon^2}\right] \ln\left(\frac{1}{\delta}\right)\right)$$

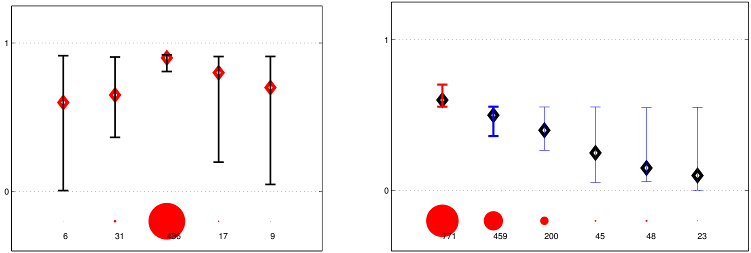
(kl)-LUCB in action

$$\begin{aligned}\mathrm{UCB}_a(t) &= \max\left\{q \in [0,1] : N_a(t)\mathrm{kl}(\hat{\mu}_a(t), q) \leq \log(Ct^2/\delta)\right\} \\ \mathrm{LCB}_a(t) &= \min\left\{q \in [0,1] : N_a(t)\mathrm{kl}(\hat{\mu}_a(t), q) \leq \log(Ct^2/\delta)\right\}\end{aligned}$$



A comparison with UCB

Regret minimizing algorithms and Best Arm Identification algorithms behave quite differently



Number of selections and confidence intervals for KL-UCB (left) and KL-LUCB (right)

Abbasi-Yadkori, Y., D.Pál, and C.Szepesvári (2011).
Improved Algorithms for Linear Stochastic Bandits.
In *Advances in Neural Information Processing Systems*.

Agrawal, S. and Goyal, N. (2013).
Thompson Sampling for Contextual Bandits with Linear Payoffs.
In *International Conference on Machine Learning (ICML)*.

Audibert, J.-Y., Bubeck, S., and Munos, R. (2010).
Best Arm Identification in Multi-armed Bandits.
In *Proceedings of the 23rd Conference on Learning Theory*.

Bubeck, S., Munos, R., and Stoltz, G. (2011a).
Pure Exploration in Finitely Armed and Continuous Armed Bandits.
Theoretical Computer Science 412, 1832-1852, 412 :1832–1852.

Bubeck, S., Munos, R., Stoltz, G., and Szepesvári, C. (2011b).
X-armed bandits.
Journal of Machine Learning Research, 12 :1587–1627.

Chen, W., Wang, Y., and Yuan, Y. (2013).
Combinatorial multi-armed bandit : General framework and applications.
In *International Conference on Machine Learning*.

Dumitrascu, B., Feng, K., and Engelhardt, B. E. (2018).
PG-TS : improved thompson sampling for logistic contextual bandits.
In *Advances in Neural Information Processing Systems (NeurIPS)*.

Even-Dar, E., Mannor, S., and Mansour, Y. (2006).
Action Elimination and Stopping Conditions for the Multi-Armed Bandit and Reinforcement Learning Problems.
Journal of Machine Learning Research, 7 :1079–1105.

Filippi, S., Cappé, O., Garivier, A., and Szepesvári, C. (2010).
Parametric Bandits : The Generalized Linear case.
In *Advances in Neural Information Processing Systems*.

Kalyanakrishnan, S., Tewari, A., Auer, P., and Stone, P. (2012).
PAC subset selection in stochastic multi-armed bandits.
In *International Conference on Machine Learning (ICML)*.

Karnin, Z., Koren, T., and Somekh, O. (2013).
Almost optimal Exploration in multi-armed bandits.
In *International Conference on Machine Learning (ICML)*.

Srinivas, N., Krause, A., Kakade, S., and Seeger, M. (2010).
Gaussian Process Optimization in the Bandit Setting : No Regret and Experimental Design.
In *Proceedings of the International Conference on Machine Learning*.