

# The Bernoulli Generalized Likelihood Ratio test (BGLR) for Non-Stationary Multi-Armed Bandits

Research Seminar at PANAMA, IRISA lab, Rennes

**Lilian Besson**

PhD Student

SCEE team, IETR laboratory, CentraleSupélec in Rennes  
& SequeL team, CRIStAL laboratory, Inria in Lille

Thursday 6<sup>th</sup> of June, 2019



# Publications associated with this talk

Joint work with my advisor Émilie Kaufmann 🤖:

- *“Analyse non asymptotique d’un test séquentiel de détection de ruptures et application aux bandits non stationnaires”*  
by **Lilian Besson** & Émilie Kaufmann  
↪ presented at **GRETSI**, in Lille (France), next August 2019  
↪ [perso.crans.org/besson/articles/BK\\_\\_GRETSI\\_2019.pdf](https://perso.crans.org/besson/articles/BK__GRETSI_2019.pdf)
- *“The Generalized Likelihood Ratio Test meets klUCB: an Improved Algorithm for Piece-Wise Non-Stationary Bandits”*  
by **Lilian Besson** & Émilie Kaufmann  
Pre-print on [HAL-02006471](https://hal.archives-ouvertes.fr/hal-02006471) and [arXiv:1902.01575](https://arxiv.org/abs/1902.01575)

# Outline of the talk

- 1 (Stationary) Multi-armed bandits problems
- 2 Piece-wise stationary multi-armed bandits problems
- 3 The BGLR test and its finite time properties
- 4 The BGLR-T + klUCB algorithm
- 5 Regret analysis
- 6 Numerical simulations

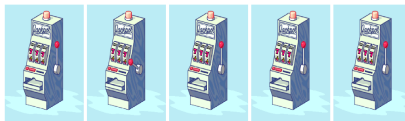
# 1. (Stationary) Multi-armed bandits problems

- ① **(Stationary) Multi-armed bandits problems**
- ② Piece-wise stationary multi-armed bandits problems
- ③ The BGLR test and its finite time properties
- ④ The BGLR-T + klUCB algorithm
- ⑤ Regret analysis
- ⑥ Numerical simulations

# Multi-armed bandits

= Sequential decision making problems in uncertain environments :

Total  
Reward 14  
Total  
Plays 24



1

	Arm 1	Arm 2	Arm 3	Arm 4	Arm 5
<b>Rewards:</b>	6	2	2	2	2
<b>Pulls:</b>	8	4	4	4	4
<b>Estimated Probs:</b>	0.750	0.500	0.500	0.500	0.500
<b>UCBs:</b>	1.641	1.761	1.761	1.761	1.761

↪ Interactive demo [perso.crans.org/besson/phd/MAB\\_interactive\\_demo/](https://perso.crans.org/besson/phd/MAB_interactive_demo/)

Ref: [Bandits Algorithms, Lattimore & Szepesvári, 2019], on [tor-lattimore.com/downloads/book/book.pdf](https://tor-lattimore.com/downloads/book/book.pdf)

# Mathematical model

- Discrete time steps  $t = 1, \dots, T$   
The *horizon*  $T$  is fixed and usually unknown
- At time  $t$ , an *agent plays the arm*  $A(t) \in \{1, \dots, K\}$ ,  
then she observes the *iid random reward*  $r(t) \sim \nu_k, r(t) \in \mathbb{R}$

# Mathematical model

- Discrete time steps  $t = 1, \dots, T$   
The *horizon*  $T$  is fixed and usually unknown
- At time  $t$ , an *agent plays the arm*  $A(t) \in \{1, \dots, K\}$ ,  
then she observes the *iid random reward*  $r(t) \sim \nu_k, r(t) \in \mathbb{R}$
- Usually, we focus on Bernoulli arms  $\nu_k = \text{Bernoulli}(\mu_k)$ , of mean  $\mu_k \in [0, 1]$ , giving binary rewards  $r(t) \in \{0, 1\}$ .

# Mathematical model

- Discrete time steps  $t = 1, \dots, T$   
The *horizon*  $T$  is fixed and usually unknown
- At time  $t$ , an *agent plays the arm*  $A(t) \in \{1, \dots, K\}$ ,  
then she observes the *iid random reward*  $r(t) \sim \nu_k, r(t) \in \mathbb{R}$
- Usually, we focus on Bernoulli arms  $\nu_k = \text{Bernoulli}(\mu_k)$ , of mean  $\mu_k \in [0, 1]$ , giving binary rewards  $r(t) \in \{0, 1\}$ .
- **Goal** : maximize the sum of rewards  $\sum_{t=1}^T r(t)$
- or maximize the sum of expected rewards  $\mathbb{E} \left[ \sum_{t=1}^T r(t) \right]$



# Mathematical model

- Discrete time steps  $t = 1, \dots, T$   
The *horizon*  $T$  is fixed and usually unknown
- At time  $t$ , an *agent plays the arm*  $A(t) \in \{1, \dots, K\}$ ,  
then she observes the *iid random reward*  $r(t) \sim \nu_k, r(t) \in \mathbb{R}$
- Usually, we focus on Bernoulli arms  $\nu_k = \text{Bernoulli}(\mu_k)$ , of mean  $\mu_k \in [0, 1]$ , giving binary rewards  $r(t) \in \{0, 1\}$ .
- **Goal** : maximize the sum of rewards  $\sum_{t=1}^T r(t)$
- or maximize the sum of expected rewards  $\mathbb{E} \left[ \sum_{t=1}^T r(t) \right]$
- Any efficient policy must balance **between exploration and exploitation**: explore all arms to discover the best one, while exploiting the arms known to be good so far.

# Two examples of bad solutions

## i) Pure exploration 😞

- Play arm  $A(t) \sim \mathcal{U}(\{1, \dots, K\})$  uniformly at random
- $\implies$  Mean expected rewards  $\frac{1}{T} \mathbb{E} \left[ \sum_{t=1}^T r(t) \right] = \frac{1}{K} \sum_{k=1}^K \mu_k \ll \max_k \mu_k$

# Two examples of bad solutions

## i) Pure exploration 😞

- Play arm  $A(t) \sim \mathcal{U}(\{1, \dots, K\})$  uniformly at random
- $\implies$  Mean expected rewards  $\frac{1}{T} \mathbb{E} \left[ \sum_{t=1}^T r(t) \right] = \frac{1}{K} \sum_{k=1}^K \mu_k \ll \max_k \mu_k$

## ii) Pure exploitation 😞

- Count the number of samples and the sum of rewards of each arm  
 $N_k(t) = \sum_{s < t} \mathbb{1}(A(s) = k)$  and  $X_k(t) = \sum_{s < t} r(s) \mathbb{1}(A(s) = k)$
- Estimate the **unknown** mean  $\mu_k$  with  $\widehat{\mu}_k(t) = X_k(t)/N_k(t)$
- Play the arm of maximum empirical mean :  $A(t) = \arg \max_k \widehat{\mu}_k(t)$
- Performance depends on the first draws, and can be very poor!

$\hookrightarrow$  Interactive demo [perso.crans.org/besson/phd/MAB\\_interactive\\_demo/](https://perso.crans.org/besson/phd/MAB_interactive_demo/)

# A first solution: “Upper Confidence Bound” algorithm

- Compute  $UCB_k(t) = X_k(t)/N_k(t) + \sqrt{\alpha \log(t)/N_k(t)}$   
= an **upper confidence bound** on the **unknown** mean  $\mu_k$
- Play the arm of maximal UCB :  $A(t) = \arg \max_k UCB_k(t)$   
 $\hookrightarrow$  Principle of “optimism under uncertainty”
- $\alpha$  balances between *exploitation* ( $\alpha \rightarrow 0$ ) and *exploration* ( $\alpha \rightarrow \infty$ )

# A first solution: “Upper Confidence Bound” algorithm

- Compute  $UCB_k(t) = X_k(t)/N_k(t) + \sqrt{\alpha \log(t)/N_k(t)}$   
= an **upper confidence bound** on the **unknown** mean  $\mu_k$
- Play the arm of maximal UCB :  $A(t) = \arg \max_k UCB_k(t)$   
 $\hookrightarrow$  Principle of “optimism under uncertainty”
- $\alpha$  balances between *exploitation* ( $\alpha \rightarrow 0$ ) and *exploration* ( $\alpha \rightarrow \infty$ )
- **UCB is efficient**: the best arm is identified correctly (with high probability) if there are enough samples (for  $T$  large enough)
- $\implies$  Expected rewards attains the maximum 😊

$$\text{For } T \rightarrow \infty, \quad \frac{1}{T} \mathbb{E} \left[ \sum_{t=1}^T r(t) \right] \rightarrow \max_k \mu_k$$

# UCB algorithm converges to the best arm

We can prove that suboptimal arms  $k$  are sampled about  $o(T)$  times

$$\implies \mathbb{E} \left[ \sum_{t=1}^T r(t) \right] \xrightarrow{T \rightarrow \infty} \mu^* \times \mathcal{O}(T) + \sum_{k: \Delta_k > 0} \mu_k \times o(T) \text{ 😊}$$

But... at which speed do we have this convergence?

## Elements of proof of convergence (for $K$ Bernoulli arms)

- Suppose the first arm is the best:  $\mu^* = \mu_1 > \mu_2 \geq \dots \geq \mu_K$
- $\text{UCB}_k(t) = X_k(t)/N_k(t) + \sqrt{\alpha \log(t)/N_k(t)}$
- Hoeffding's inequality gives  $\mathbb{P}(\text{UCB}_k(t) < \mu_k(t)) \leq \mathcal{O}(\frac{1}{t^{2\alpha}})$   
 $\implies$  the different  $\text{UCB}_k(t)$  are true “Upper Confidence Bounds” on the (unknown)  $\mu_k$  (most of the times)
- And if a suboptimal arm  $k > 1$  is sampled, it implies  $\text{UCB}_k(t) > \text{UCB}_1(t)$ , but  $\mu_k < \mu_1$ : Hoeffding's inequality also proves that any “wrong ordering” of the  $\text{UCB}_k(t)$  is unlikely

Measure the performance of algorithm  $\mathcal{A}$  by its mean regret  $R_{\mathcal{A}}(T)$

- Difference in the accumulated rewards between an “oracle” and  $\mathcal{A}$
- The “oracle” algorithm always plays **the (unknown) best arm**  
 $k^* = \arg \max_k \mu_k$  (we note the best mean  $\mu_{k^*} = \mu^*$ )
- Maximize the sum of expected rewards  $\iff$  **minimize the regret**

$$R_{\mathcal{A}}(T) = \mathbb{E} \left[ \sum_{t=1}^T r_{k^*}(t) \right] - \sum_{t=1}^T \mathbb{E} [r(t)] = T\mu^* - \sum_{t=1}^T \mathbb{E} [r(t)] .$$

Measure the performance of algorithm  $\mathcal{A}$  by its mean regret  $R_{\mathcal{A}}(T)$

- Difference in the accumulated rewards between an “oracle” and  $\mathcal{A}$
- The “oracle” algorithm always plays **the (unknown) best arm**  
 $k^* = \arg \max_k \mu_k$  (we note the best mean  $\mu_{k^*} = \mu^*$ )
- Maximize the sum of expected rewards  $\iff$  **minimize the regret**

$$R_{\mathcal{A}}(T) = \mathbb{E} \left[ \sum_{t=1}^T r_{k^*}(t) \right] - \sum_{t=1}^T \mathbb{E} [r(t)] = T\mu^* - \sum_{t=1}^T \mathbb{E} [r(t)] .$$

### Typical regime for stationary bandits (lower & upper bounds)

- No algorithm  $\mathcal{A}$  can obtain a regret better than  $R_{\mathcal{A}}(T) \geq \Omega(\log(T))$
- And an efficient algorithm  $\mathcal{A}$  obtains  $R_{\mathcal{A}}(T) \leq \mathcal{O}(\log(T))$



# Regret of UCB and kl-UCB algorithms

For any problem with  $K$  arms following Bernoulli distributions, of means  $\mu_1, \dots, \mu_K \in [0, 1]$ , and **optimal mean  $\mu^*$** , then

For the UCB algorithm

$$R_T^{\text{UCB}} \leq \sum_{\substack{k=1, \dots, K \\ \mu_k < \mu^*}} \frac{8}{(\mu_k - \mu^*)} \log(T) + o(\log(T)).$$

# Regret of UCB and kl-UCB algorithms

For any problem with  $K$  arms following Bernoulli distributions, of means  $\mu_1, \dots, \mu_K \in [0, 1]$ , and **optimal mean  $\mu^*$** , then

For the UCB algorithm

$$R_T^{\text{UCB}} \leq \sum_{\substack{k=1, \dots, K \\ \mu_k < \mu^*}} \frac{8}{(\mu_k - \mu^*)} \log(T) + o(\log(T)).$$

For the kl-UCB algorithm: a smaller regret upper-bound

$$R_T^{\text{kl-UCB}} \leq \sum_{\substack{k=1, \dots, K \\ \mu_k < \mu^*}} \frac{(\mu_k - \mu^*)}{\text{kl}(\mu^*, \mu_k)} \log(T) + o(\log(T)) = \mathcal{O}(\underbrace{C(\mu_1, \dots, \mu_K)}_{\text{Difficulty of the problem}} \log(T)).$$

If  $\text{kl}(x, y) = x \log(x/y) + (1-x) \log((1-x)/(1-y))$  is the *binary relative entropy* (ie, Kullback-Leibler divergence of two Bernoulli of means  $x$  and  $y$ )

## 2. Piece-wise stationary MAB problems

- ① (Stationary) Multi-armed bandits problems
- ② **Piece-wise stationary multi-armed bandits problems**
- ③ The BGLR test and its finite time properties
- ④ The BGLR-T + klUCB algorithm
- ⑤ Regret analysis
- ⑥ Numerical simulations

# Non stationary MAB problems

## Stationary MAB problems

Arm  $k$  gives rewards sampled from **the same distribution** for any time step:  $\forall t, r_k(t) \stackrel{\text{iid}}{\sim} \nu_k = \text{Bernoulli}(\mu_k)$ .

# Non stationary MAB problems

## Stationary MAB problems

Arm  $k$  gives rewards sampled from **the same distribution** for any time step:  $\forall t, r_k(t) \stackrel{\text{iid}}{\sim} \nu_k = \text{Bernoulli}(\mu_k)$ .

## Non stationary MAB problems?

Arm  $k$  gives rewards sampled a **(possibly) different distributions** for any time step:  $\forall t, r_k(t) \stackrel{\text{iid}}{\sim} \nu_k(t) = \text{Bernoulli}(\mu_k(t))$ .

$\implies$  😞 harder problem! And very hard if  $\mu_k(t)$  can change at any step!

# Non stationary MAB problems

## Stationary MAB problems

Arm  $k$  gives rewards sampled from **the same distribution** for any time step:  $\forall t, r_k(t) \stackrel{\text{iid}}{\sim} \nu_k = \text{Bernoulli}(\mu_k)$ .

## Non stationary MAB problems?

Arm  $k$  gives rewards sampled a **(possibly) different distributions** for any time step:  $\forall t, r_k(t) \stackrel{\text{iid}}{\sim} \nu_k(t) = \text{Bernoulli}(\mu_k(t))$ .

$\implies$  😞 harder problem! And very hard if  $\mu_k(t)$  can change at any step!

## Piece-wise stationary problems!

$\hookrightarrow$  we focus on the easier case when there are at most  $o(\sqrt{T})$  intervals on which the means are all stationary (= **sequence**)

# Break-points and stationary sequences

Define

- The number of break-points

$$\Upsilon_T = \sum_{t=1}^{T-1} \mathbb{1}(\exists k \in \{1, \dots, K\} : \mu_k(t) \neq \mu_k(t+1))$$

- The  $i$ -th break-point

$$\tau^i = \inf\{t > \tau^{i-1} : \exists k : \mu_k(t) \neq \mu_k(t+1)\} \quad (\text{with } \tau^0 = 0)$$

# Break-points and stationary sequences

## Define

- The number of break-points

$$\Upsilon_T = \sum_{t=1}^{T-1} \mathbb{1}(\exists k \in \{1, \dots, K\} : \mu_k(t) \neq \mu_k(t+1))$$

- The  $i$ -th break-point

$$\tau^i = \inf\{t > \tau^{i-1} : \exists k : \mu_k(t) \neq \mu_k(t+1)\} \quad (\text{with } \tau^0 = 0)$$

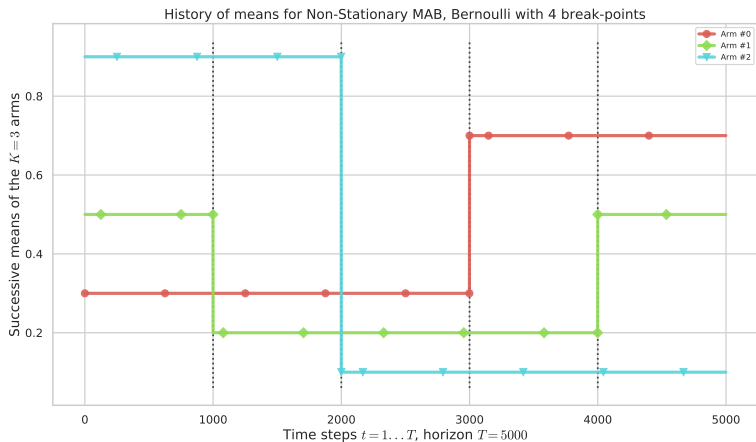
## Hypotheses on piece-wise stationary problems

- The rewards  $r_k(t)$  generated by each arm  $k$  are *iid on each interval*  $[\tau^i + 1, \tau^{i+1}]$  (the  $i$ -th sequence)
- There are  $\Upsilon_T = o(\sqrt{T})$  break-points
- And  $\Upsilon_T$  can be known before-hand
- All sequences are “long enough”



# Example of a piece-wise stationary MAB problem

We plot the means  $\mu_1(t)$ ,  $\mu_2(t)$ ,  $\mu_3(t)$  of  $K = 3$  arms. There are  $\Upsilon_T = 4$  break-points and 5 sequences between  $t = 1$  and  $t = T = 5000$ :



# Regret for piece-wise stationary bandits?

The “oracle” algorithm know plays the (unknown) best arm  
 $k^*(t) = \arg \max \mu_k(t)$  (which changes between stationary sequences)

$$R_{\mathcal{A}}(T) = \mathbb{E} \left[ \sum_{t=1}^T r_{k^*(t)}(t) \right] - \sum_{t=1}^T \mathbb{E} [r(t)] = \left( \sum_{t=1}^T \max_k \mu_k(t) \right) - \sum_{t=1}^T \mathbb{E} [r(t)] .$$

# Regret for piece-wise stationary bandits?

The “oracle” algorithm know plays the (unknown) best arm  
 $k^*(t) = \arg \max_k \mu_k(t)$  (which changes between stationary sequences)

$$R_{\mathcal{A}}(T) = \mathbb{E} \left[ \sum_{t=1}^T r_{k^*(t)}(t) \right] - \sum_{t=1}^T \mathbb{E} [r(t)] = \left( \sum_{t=1}^T \max_k \mu_k(t) \right) - \sum_{t=1}^T \mathbb{E} [r(t)] .$$

## Typical regimes for piece-wise stationary bandits

- The lower-bound is  $R_{\mathcal{A}}(T) \geq \Omega(\sqrt{KT\Upsilon_T})$
- Currently, state-of-the-art algorithms  $\mathcal{A}$  obtain
  - $R_{\mathcal{A}}(T) \leq \mathcal{O}(K\sqrt{T\Upsilon_T \log(T)})$  if  $T$  and  $\Upsilon_T$  are known
  - $R_{\mathcal{A}}(T) \leq \mathcal{O}(K\Upsilon_T\sqrt{T \log(T)})$  if  $T$  and  $\Upsilon_T$  are unknown

## 3. The BGLR test and its finite time properties

- 1 (Stationary) Multi-armed bandits problems
- 2 Piece-wise stationary multi-armed bandits problems
- 3 **The BGLR test and its finite time properties**
- 4 The BGLR-T + klUCB algorithm
- 5 Regret analysis
- 6 Numerical simulations

# The break-point detection problem

Imagine the following problem...

- You observe data  $X_1, X_2, \dots, X_t, \dots \in [0, 1]$  sequentially...
- You know that  $X_t$  is generated by a certain **unknown** distribution...

# The break-point detection problem

Imagine the following problem...

- You observe data  $X_1, X_2, \dots, X_t, \dots \in [0, 1]$  sequentially...
- You know that  $X_t$  is generated by a certain **unknown** distribution...
- **Your goal** is to distinguish between two hypotheses:
  - $\mathcal{H}_0$  The distributions **all have the same mean** (“no break-point”)
 
$$\exists \mu_0, \mathbb{E}[X_1] = \mathbb{E}[X_2] = \dots = \mathbb{E}[X_t] = \mu_0$$
  - $\mathcal{H}_1$  The distributions have **changed mean at a break-point at time  $\tau$** 

$$\exists \mu_0, \mu_1, \tau, \mathbb{E}[X_1] = \dots = \mathbb{E}[X_\tau] = \mu_0, \mu_0 \neq \mu_1, \mathbb{E}[X_{\tau+1}] = \mathbb{E}[X_{\tau+2}] = \dots = \mu_1$$
- You stop at time  $\hat{\tau}$ , as soon as you detect a change

# The break-point detection problem

Imagine the following problem...

- You observe data  $X_1, X_2, \dots, X_t, \dots \in [0, 1]$  sequentially...
- You know that  $X_t$  is generated by a certain **unknown** distribution...
- **Your goal** is to distinguish between two hypotheses:
  - $\mathcal{H}_0$  The distributions **all have the same mean** (“no break-point”)
 
$$\exists \mu_0, \mathbb{E}[X_1] = \mathbb{E}[X_2] = \dots = \mathbb{E}[X_t] = \mu_0$$
  - $\mathcal{H}_1$  The distributions have **changed mean at a break-point at time  $\tau$** 

$$\exists \mu_0, \mu_1, \tau, \mathbb{E}[X_1] = \dots = \mathbb{E}[X_\tau] = \mu_0, \mu_0 \neq \mu_1, \mathbb{E}[X_{\tau+1}] = \mathbb{E}[X_{\tau+2}] = \dots = \mu_1$$
- You stop at time  $\hat{\tau}$ , as soon as you detect a change

A **sequential break-point detection** is a **stopping time**  $\hat{\tau}$ , measurable for  $\mathcal{F}_t = \sigma(X_1, \dots, X_t)$ , which rejects hypothesis  $\mathcal{H}_0$  when  $\hat{\tau} < \infty$ .

# Bernoulli likelihood ratio test

**Hypothesis:** all distributions are Bernoulli ( $\nu_k = \mathcal{B}(\mu_k)$ )

The problem boils down to distinguishing

$\mathcal{H}_0$ :  $(\exists \mu_0 : \forall i \in \mathbb{N}^*, X_i \stackrel{\text{i.i.d.}}{\sim} \mathcal{B}(\mu_0))$ , against the alternative

$\mathcal{H}_1$ :  $(\exists \mu_0 \neq \mu_1, \tau > 1 : X_1, \dots, X_\tau \stackrel{\text{i.i.d.}}{\sim} \mathcal{B}(\mu_0) \text{ et } X_{\tau+1}, \dots \stackrel{\text{i.i.d.}}{\sim} \mathcal{B}(\mu_1))$ .



# Bernoulli likelihood ratio test

**Hypothesis:** all distributions are Bernoulli ( $\nu_k = \mathcal{B}(\mu_k)$ )

The problem boils down to distinguishing

$\mathcal{H}_0$ : ( $\exists \mu_0 : \forall i \in \mathbb{N}^*, X_i \stackrel{\text{i.i.d.}}{\sim} \mathcal{B}(\mu_0)$ ), against the alternative

$\mathcal{H}_1$ : ( $\exists \mu_0 \neq \mu_1, \tau > 1 : X_1, \dots, X_\tau \stackrel{\text{i.i.d.}}{\sim} \mathcal{B}(\mu_0)$  et  $X_{\tau+1}, \dots \stackrel{\text{i.i.d.}}{\sim} \mathcal{B}(\mu_1)$ ).

The **Likelihood Ratio statistic** for this hypothesis test, after observing  $X_1, \dots, X_n$ , is

$$\mathcal{L}(n) = \frac{\sup_{\mu_0, \mu_1, \tau < n} \ell(X_1, \dots, X_n; \mu_0, \mu_1, \tau)}{\sup_{\mu_0} \ell(X_1, \dots, X_n; \mu_0)},$$

where  $\ell(X_1, \dots, X_n; \mu_0)$  (resp.  $\ell(X_1, \dots, X_n; \mu_0, \mu_1, \tau)$ ) is the likelihood of the observations under a model in  $\mathcal{H}_0$  (resp.  $\mathcal{H}_1$ ).

# Bernoulli likelihood ratio test

**Hypothesis:** all distributions are Bernoulli ( $\nu_k = \mathcal{B}(\mu_k)$ )

The problem boils down to distinguishing

$\mathcal{H}_0$ : ( $\exists \mu_0 : \forall i \in \mathbb{N}^*, X_i \stackrel{\text{i.i.d.}}{\sim} \mathcal{B}(\mu_0)$ ), against the alternative

$\mathcal{H}_1$ : ( $\exists \mu_0 \neq \mu_1, \tau > 1 : X_1, \dots, X_\tau \stackrel{\text{i.i.d.}}{\sim} \mathcal{B}(\mu_0)$  et  $X_{\tau+1}, \dots \stackrel{\text{i.i.d.}}{\sim} \mathcal{B}(\mu_1)$ ).

The **Likelihood Ratio statistic** for this hypothesis test, after observing  $X_1, \dots, X_n$ , is

$$\mathcal{L}(n) = \frac{\sup_{\mu_0, \mu_1, \tau < n} \ell(X_1, \dots, X_n; \mu_0, \mu_1, \tau)}{\sup_{\mu_0} \ell(X_1, \dots, X_n; \mu_0)},$$

where  $\ell(X_1, \dots, X_n; \mu_0)$  (resp.  $\ell(X_1, \dots, X_n; \mu_0, \mu_1, \tau)$ ) is the likelihood of the observations under a model in  $\mathcal{H}_0$  (resp.  $\mathcal{H}_1$ ).

$\Leftrightarrow$  High values of this statistic  $\mathcal{L}(n)$  tends to reject  $\mathcal{H}_0$  over  $\mathcal{H}_1$ .

# Expression of the (log) Bernoulli Likelihood ratio

We can rewrite this statistic  $\mathcal{L}(n) = \frac{\sup_{\mu_0, \mu_1, \tau < n} \ell(X_1, \dots, X_n; \mu_0, \mu_1, \tau)}{\sup_{\mu_0} \ell(X_1, \dots, X_n; \mu_0)}$ , by using

Bernoulli likelihood, and shifting means  $\hat{\mu}_{k:k'} = \frac{1}{k' - k + 1} \sum_{s=k}^{k'} X_s :$

$$\log \mathcal{L}(n) = \max_{s \in \{2, \dots, n-1\}} \left[ s \times \text{kl} \left( \underbrace{\hat{\mu}_{1:s}}_{\text{before change}}, \underbrace{\hat{\mu}_{1:n}}_{\text{all data}} \right) + (n - s) \times \text{kl} \left( \underbrace{\hat{\mu}_{s+1:n}}_{\text{after change}}, \underbrace{\hat{\mu}_{1:n}}_{\text{all data}} \right) \right].$$

Where  $\text{kl}(x, y) = x \ln\left(\frac{x}{y}\right) + (1 - x) \ln\left(\frac{1-x}{1-y}\right)$  is the binary relative entropy

# The Bernoulli Generalized likelihood ratio test (BGLR)

- We can extend the Bernoulli likelihood ratio test if the observations are **sub-Bernoulli**.
- And any bounded distributions on  $[0, 1]$  is sub-Bernoulli !
- $\implies$  the BGLR test can be applied for any bounded observations 😊

# The Bernoulli Generalized likelihood ratio test (BGLR)

- We can extend the Bernoulli likelihood ratio test if the observations are **sub-Bernoulli**.
- And any bounded distributions on  $[0, 1]$  is sub-Bernoulli !
- $\implies$  the BGLR test can be applied for any bounded observations 😊

## The BGRL-T sequential break-point detection test

The **BGLR-T** is the stopping time defined by

$$\hat{\tau}_\delta = \inf \left\{ n \in \mathbb{N}^* : \max_{s \in \{2, \dots, n-1\}} \left[ s \text{kl}(\hat{\mu}_{1:s}, \hat{\mu}_{1:n}) + (n-s) \text{kl}(\hat{\mu}_{s+1:n}, \hat{\mu}_{1:n}) \right] \geq \beta(n, \delta) \right\}$$

- with a **threshold function**  $\beta(n, \delta)$  specified later,
- $n$  is the number of observations,
- $\delta$  is the confidence level.

# Probability of false alarm

A good test should not detect any break-point if there is no break-point to detect. . .

# Probability of false alarm

A good test should not detect any break-point if there is no break-point to detect. . .

## Definition: False alarm

The stopping time is  $\hat{\tau}_\delta$ , and a break-point is detected if  $\hat{\tau}_\delta < \infty$ .

Let  $\mathbb{P}_{\mu_0}$  be a probability model under which the observations are  $\forall t, X_t \in [0, 1]$  and  $\forall t, \mathbb{E}[X_t] = \mu_0$ .

The **false alarm probability** is  $\mathbb{P}_{\mu_0}(\hat{\tau}_\delta < \infty)$ .

$\implies$  **Goal: controlling the false alarm event!** (in high probability)

# First result for the BGLR test 😊

## Controlling the false alarm probability

For any **confidence level**  $0 < \delta < 1$ , the BGLR test satisfies

$$\mathbb{P}_{\mu_0}(\hat{\tau}_\delta < \infty) \leq \delta$$

with the threshold function

$$\beta(n, \delta) = 2 \mathcal{T} \left( \frac{\ln(3n\sqrt{n}/\delta)}{2} \right) + 6 \ln(1 + \ln(n)) \simeq \ln \left( \frac{3n\sqrt{n}}{\delta} \right) = \mathcal{O} \left( \log \left( \frac{n}{\delta} \right) \right).$$

Where  $\mathcal{T}(x)$  verifies  $\mathcal{T}(x) \simeq x + \ln(x)$  for  $x$  large enough



# First result for the BGLR test 😊

## Controlling the false alarm probability

For any **confidence level**  $0 < \delta < 1$ , the BGLR test satisfies

$$\mathbb{P}_{\mu_0}(\hat{\tau}_\delta < \infty) \leq \delta$$

with the threshold function

$$\beta(n, \delta) = 2 \mathcal{T} \left( \frac{\ln(3n\sqrt{n}/\delta)}{2} \right) + 6 \ln(1 + \ln(n)) \simeq \ln \left( \frac{3n\sqrt{n}}{\delta} \right) = \mathcal{O} \left( \log \left( \frac{n}{\delta} \right) \right).$$

Where  $\mathcal{T}(x)$  verifies  $\mathcal{T}(x) \simeq x + \ln(x)$  for  $x$  large enough

## Proof ?

Hard to explain in a short time...

↪ see the article, on [HAL-02006471](#) and [arXiv:1902.01575](#)

# Delay of detection

A good test should detect a break-point “fast enough” if there is a break-point to detect, with enough samples before the break-point. . .

# Delay of detection

A good test should detect a break-point “fast enough” if there is a break-point to detect, with enough samples before the break-point. . .

## Definition: Delay of detection

Let  $\mathbb{P}_{\mu_0, \mu_1, \tau}$  be a probability model under which  $\forall t, X_t \in [0, 1]$  and  $\forall t \leq \tau, \mathbb{E}[X_t] = \mu_0$  and  $\forall t \geq \tau + 1, \mathbb{E}[X_t] = \mu_1$ , with  $\mu_0 \neq \mu_1$ .

The **gap** of this break-point is  $\Delta = |\mu_0 - \mu_1|$ .

The **delay of detection** is  $u = \hat{\tau}_\delta - \tau \in \mathbb{N}$ .

$\implies$  **Goal: controlling the delay of detection!** (in high probability)

## Second result for the BGLR test 😊

### Controlling the delay of detection

On a break-point of amplitude  $\Delta = |\mu_1 - \mu_0|$ , the BGLRT test satisfies

$$\begin{aligned} \mathbb{P}_{\mu_0, \mu_1, \tau}(\hat{\tau}_\delta \geq \tau + u) &\leq \exp \left( -\frac{2\tau u}{\tau + u} \left( \max \left[ 0, \Delta - \sqrt{\frac{\tau + u}{2\tau u}} \beta(\tau + u, \delta) \right] \right)^2 \right) \\ &= \mathcal{O}(\text{decreasing exponential of } u) = \mathcal{O}(\exp \searrow (u)). \end{aligned}$$

with the same threshold function  $\beta(n, \delta) \simeq \ln(3n\sqrt{n}/\delta)$ .

### Consequence

In high probability, **the delay  $\hat{\tau}_\delta$  of BGLR is bounded** by  $\mathcal{O}(\Delta^{-2} \ln(1/\delta))$  **if enough samples are observed before the break-point at time  $\tau$ .**

# BGLR is an efficient break-point detection test 😊 !

- We just saw that by choosing
  - a confidence level  $\delta$ ,
  - and a good threshold function  $\beta(n, \delta) \simeq \ln(3n\sqrt{n}/\delta) = \mathcal{O}(\log(n/\delta))$

# BGLR is an efficient break-point detection test 😊 !

- We just saw that by choosing
  - a confidence level  $\delta$ ,
  - and a good threshold function  $\beta(n, \delta) \simeq \ln(3n\sqrt{n}/\delta) = \mathcal{O}(\log(n/\delta))$
- we can control the two properties of the BGLR test:
  - its **false alarm probability**:  $\mathbb{P}_{\mu_0}(\hat{\tau}_\delta < \infty) \leq \delta$
  - its **detection delay**:  $\mathbb{P}_{\mu_0, \mu_1, \tau}(\hat{\tau}_\delta \geq \tau + u)$  decreases exponentially fast wrt  $u$  (if there are enough samples before and after the break-point)
- $\implies$  The BGLR is an efficient break-point detection test 😊

# BGLR is an efficient break-point detection test 😊 !

- We just saw that by choosing
  - a confidence level  $\delta$ ,
  - and a good threshold function  $\beta(n, \delta) \simeq \ln(3n\sqrt{n}/\delta) = \mathcal{O}(\log(n/\delta))$
- we can control the two properties of the BGLR test:
  - its **false alarm probability**:  $\mathbb{P}_{\mu_0}(\hat{\tau}_\delta < \infty) \leq \delta$
  - its **detection delay**:  $\mathbb{P}_{\mu_0, \mu_1, \tau}(\hat{\tau}_\delta \geq \tau + u)$  decreases exponentially fast wrt  $u$  (if there are enough samples before and after the break-point)
- $\implies$  The BGLR is an efficient break-point detection test 😊

## Finite time guarantees 😊

[Maillard, ALT, 2019] [Lai & Xing, Sequential Analysis, 2010]

Such **finite time** (non asymptotic) guarantees are recent results!

## 4. The BGLR-T + klUCB algorithm

- 1 (Stationary) Multi-armed bandits problems
- 2 Piece-wise stationary multi-armed bandits problems
- 3 The BGLR test and its finite time properties
- 4 **The BGLR-T + klUCB algorithm**
- 5 Regret analysis
- 6 Numerical simulations



# Our algorithm combines BGRL test + kl-UCB index

## Main ideas

- We compute a UCB index on each arm  $k$
- Most of the times, we select  $A(t) = \arg \max_{k \in \{1, \dots, K\}} \text{kl-UCB}_k(t)$
- We use a BGRL test **to detect changes on the played arm  $A(t)$**
- If a break-point is detected, **we reset the memories of *all arms***

# Our algorithm combines BGRL test + kl-UCB index

## Main ideas

- We compute a UCB index on each arm  $k$
- Most of the times, we select  $A(t) = \arg \max_{k \in \{1, \dots, K\}} \text{kl-UCB}_k(t)$
- We use a BGRL test **to detect changes on the played arm  $A(t)$**
- If a break-point is detected, **we reset the memories of *all arms***

## The kl-UCB indexes

- $\tau_k(t)$  is the time of last reset of arm  $k$  before time  $t$ ,
- $n_k(t)$  counts the selections and  $\hat{\mu}_k(t)$  is the empirical means of observations of arm  $k$  since  $\tau_k(t)$ ,
- Let  $\text{kl-UCB}_k(t) = \max\{q \in [0, 1] : n_k(t) \times \text{kl}(\hat{\mu}_k(t), q) \leq f(t - \tau_k(t))\}$
- $f(t) = \ln(t) + 3 \ln(\ln(t))$  controls the width of the UCB.

## Two details of our algorithm

### *i)* How do we use the BGLR test? (parameter $\delta$ )

From observations  $Z_1, \dots, Z_n$  we use the BGLR test to detect a breakpoint with confidence level  $\delta$  when

$$\sup_{2 \leq s \leq n-1} \left[ s \times \text{kl} \left( \hat{Z}_{1:s}, \hat{Z}_{1:n} \right) + (n - s) \times \text{kl} \left( \hat{Z}_{s+1:n}, \hat{Z}_{1:n} \right) \right] \geq \beta(n, \delta)$$

# Two details of our algorithm

## i) How do we use the BGLR test? (parameter $\delta$ )

From observations  $Z_1, \dots, Z_n$  we use the BGLR test to detect a break-point with confidence level  $\delta$  when

$$\sup_{2 \leq s \leq n-1} \left[ s \times \text{kl} \left( \hat{Z}_{1:s}, \hat{Z}_{1:n} \right) + (n - s) \times \text{kl} \left( \hat{Z}_{s+1:n}, \hat{Z}_{1:n} \right) \right] \geq \beta(n, \delta)$$

## ii) Forced exploration (parameter $\alpha$ )

- We use a forced exploration uniformly on all arms...  
*ie*, in average, arm  $k$  is forced to be sampled at least  $T \times \alpha / K$  times
- $\implies$  so we can detect break-points on all the arms
- and not only on the arm played by the kl-UCB indexes

# The BGLR + kl-UCB algorithm

- 1 **Data:** *Parameters of the problem* :  $T \in \mathbb{N}^*, K \in \mathbb{N}^*$
- 2 **Data:** *Parameters of the algorithm* :  $\alpha \in (0, 1), \delta > 0$  // can use  $T$  and  $\Upsilon_T$
- 3 **Initialisation** :  $\forall k \in \{1, \dots, K\}, \tau_k = 0$  and  $n_k = 0$
- 4 **for**  $t = 1, 2, \dots, T$  **do**

# The BGLR + kl-UCB algorithm

```

1 Data: Parameters of the problem :  $T \in \mathbb{N}^*, K \in \mathbb{N}^*$ 
2 Data: Parameters of the algorithm :  $\alpha \in (0, 1), \delta > 0$  // can use  $T$  and  $\Upsilon_T$ 
3 Initialisation :  $\forall k \in \{1, \dots, K\}, \tau_k = 0$  and  $n_k = 0$ 
4 for  $t = 1, 2, \dots, T$  do
5     if  $t \bmod \lfloor \frac{K}{\alpha} \rfloor \in \{1, \dots, K\}$  then
6          $A(t) = t \bmod \lfloor \frac{K}{\alpha} \rfloor$  // forced exploration
7     else
8          $A(t) = \arg \max_{k \in \{1, \dots, K\}} \text{kl-UCB}_k(t)$  // highest UCB index

```

# The BGLR + kl-UCB algorithm

```

1 Data: Parameters of the problem :  $T \in \mathbb{N}^*, K \in \mathbb{N}^*$ 
2 Data: Parameters of the algorithm :  $\alpha \in (0, 1), \delta > 0$  // can use  $T$  and  $\Upsilon_T$ 
3 Initialisation :  $\forall k \in \{1, \dots, K\}, \tau_k = 0$  and  $n_k = 0$ 
4 for  $t = 1, 2, \dots, T$  do
5     if  $t \bmod \lfloor \frac{K}{\alpha} \rfloor \in \{1, \dots, K\}$  then
6          $A(t) = t \bmod \lfloor \frac{K}{\alpha} \rfloor$  // forced exploration
7     else
8          $A(t) = \arg \max_{k \in \{1, \dots, K\}} \text{kl-UCB}_k(t)$  // highest UCB index
9     Play arm  $k = A(t)$ , and update play count  $n_{A(t)} = n_{A(t)} + 1$ 
10    Observe a reward  $X_{A(t),t}$ , and store it  $Z_{A(t),n_{A(t)}} = X_{A(t),t}$ 

```

# The BGLR + kl-UCB algorithm

```

1 Data: Parameters of the problem :  $T \in \mathbb{N}^*, K \in \mathbb{N}^*$ 
2 Data: Parameters of the algorithm :  $\alpha \in (0, 1), \delta > 0$  // can use  $T$  and  $\Upsilon_T$ 
3 Initialisation :  $\forall k \in \{1, \dots, K\}, \tau_k = 0$  and  $n_k = 0$ 
4 for  $t = 1, 2, \dots, T$  do
5     if  $t \bmod \lfloor \frac{K}{\alpha} \rfloor \in \{1, \dots, K\}$  then
6          $A(t) = t \bmod \lfloor \frac{K}{\alpha} \rfloor$  // forced exploration
7     else
8          $A(t) = \arg \max_{k \in \{1, \dots, K\}} \text{kl-UCB}_k(t)$  // highest UCB index
9     Play arm  $k = A(t)$ , and update play count  $n_{A(t)} = n_{A(t)} + 1$ 
10    Observe a reward  $X_{A(t),t}$ , and store it  $Z_{A(t),n_{A(t)}} = X_{A(t),t}$ 
11    if  $\text{BGLRT}_\delta(Z_{A(t),1}, \dots, Z_{A(t),n_{A(t)}}) = \text{True}$  then
12         $\forall k, \tau_k = t$  and  $n_k = 0$  // reset memories of all arms
13 end

```



## 5. Regret analysis

- 1 (Stationary) Multi-armed bandits problems
- 2 Piece-wise stationary multi-armed bandits problems
- 3 The BGLR test and its finite time properties
- 4 The BGLR-T + klUCB algorithm
- 5 **Regret analysis**
- 6 Numerical simulations

# Hypotheses of our theoretical analysis

- Denote  $\tau^i$  the position of break-point  $i$  ( $\tau^0 = 0$ )
- and  $\mu_k^i$  the mean of arm  $k$  on the segment  $[\tau^i, \tau^{i+1}]$
- and  $b(i) \in \arg \max_k \mu_k^i$  (one of) the best arm(s) on the  $i$ -th segment
- and the largest gap at break-point  $i$  is  $\Delta^i = \max_{k=1, \dots, K} |\mu_k^i - \mu_k^{i-1}| > 0$

# Hypotheses of our theoretical analysis

- Denote  $\tau^i$  the position of break-point  $i$  ( $\tau^0 = 0$ )
- and  $\mu_k^i$  the mean of arm  $k$  on the segment  $[\tau^i, \tau^{i+1}]$
- and  $b(i) \in \arg \max_k \mu_k^i$  (one of) the best arm(s) on the  $i$ -th segment
- and the largest gap at break-point  $i$  is  $\Delta^i = \max_{k=1, \dots, K} |\mu_k^i - \mu_k^{i-1}| > 0$

## Assumption

Fix the parameters  $\alpha$  and  $\delta$ , and let  $d^i = d^i(\alpha, \delta) = \lceil \frac{4K}{\alpha(\Delta^i)^2} \beta(T, \delta) + \frac{K}{\alpha} \rceil$ .

**We assume that all sequences are “long enough”:**

$$\forall i \in \{1, \dots, \Upsilon_T\}, \quad \tau^i - \tau^{i-1} \geq 2 \max(d^i, d^{i-1}).$$

# Hypotheses of our theoretical analysis

- Denote  $\tau^i$  the position of break-point  $i$  ( $\tau^0 = 0$ )
- and  $\mu_k^i$  the mean of arm  $k$  on the segment  $[\tau^i, \tau^{i+1}]$
- and  $b(i) \in \arg \max_k \mu_k^i$  (one of) the best arm(s) on the  $i$ -th segment
- and the largest gap at break-point  $i$  is  $\Delta^i = \max_{k=1, \dots, K} |\mu_k^i - \mu_k^{i-1}| > 0$

## Assumption

Fix the parameters  $\alpha$  and  $\delta$ , and let  $d^i = d^i(\alpha, \delta) = \lceil \frac{4K}{\alpha(\Delta^i)^2} \beta(T, \delta) + \frac{K}{\alpha} \rceil$ .

**We assume that all sequences are “long enough”:**

$$\forall i \in \{1, \dots, \Upsilon_T\}, \quad \tau^i - \tau^{i-1} \geq 2 \max(d^i, d^{i-1}).$$

$\hookrightarrow$  The **minimum length of sequence  $i$**  depends on the amplitude of the changes at **the beginning** and **the end** of the sequence ( $\Delta^{i-1}$  and  $\Delta^i$ ).

# Theoretical result

Under this hypothesis, we obtained a *finite time* upper-bound on the regret  $R_T$ , with explicit dependency from the problem difficulty.

The exact bound uses:

- the divergences  $\text{kl}(\mu_k^i, \mu_{b(i)}^i)$  account for the difficulty of the stationary problem on sequence  $i$ ,
- the gaps  $\Delta^i$  account for the difficulty of detecting break-point  $i$ ,

as well as the two parameters

- $\alpha$  the probability of forced exploration,
- and  $\delta$  the confidence level of the break-point detection test.

# Simplified form of the result for BGLR + kl-UCB

## Regret upper bound for BGLR + kl-UCB 😊

- On a problem satisfying our assumption...
- let  $\alpha = \sqrt{\Upsilon_T \ln(T)/T}$  and  $\delta = 1/\sqrt{T\Upsilon_T}$  (if  $T$  and  $\Upsilon_T$  are known),

# Simplified form of the result for BGLR + kl-UCB

## Regret upper bound for BGLR + kl-UCB 🧐

- On a problem satisfying our assumption...
- let  $\alpha = \sqrt{\Upsilon_T \ln(T)/T}$  and  $\delta = 1/\sqrt{T\Upsilon_T}$  (if  $T$  and  $\Upsilon_T$  are known),
- then if BGLR + kl-UCB uses parameters  $\alpha$  and  $\delta$ , its regret satisfies

$$R_T = \mathcal{O} \left( \frac{K}{(\Delta^{\text{change}})^2} \sqrt{T\Upsilon_T \ln(T)} + \frac{(K-1)}{\Delta^{\text{opt}}} \Upsilon_T \ln(T) \right),$$

- with  $\Delta^{\text{change}}$  = the smallest detection gap between two stationary segments = Difficulty of the break-point detection problems!
- and  $\Delta^{\text{opt}}$  = the smallest value of sub-optimality gap on a stationary segment = Difficulty of the stationary bandit problems!

# Simplified form of the result for BGLR + kl-UCB

## Regret upper bound for BGLR + kl-UCB 😊

- On a problem satisfying our assumption...
- let  $\alpha = \sqrt{\Upsilon_T \ln(T)/T}$  and  $\delta = 1/\sqrt{T\Upsilon_T}$  (if  $T$  and  $\Upsilon_T$  are known),
- then if BGLR + kl-UCB uses parameters  $\alpha$  and  $\delta$ , its regret satisfies

$$R_T = \mathcal{O} \left( \frac{K}{(\Delta^{\text{change}})^2} \sqrt{T\Upsilon_T \ln(T)} + \frac{(K-1)}{\Delta^{\text{opt}}} \Upsilon_T \ln(T) \right),$$

- with  $\Delta^{\text{change}}$  = the smallest detection gap between two stationary segments = Difficulty of the break-point detection problems!
- and  $\Delta^{\text{opt}}$  = the smallest value of sub-optimality gap on a stationary segment = Difficulty of the stationary bandit problems!

$\implies R_T = \mathcal{O}(K\sqrt{T\Upsilon_T \log(T)})$  if we hide the dependency on the gaps.



# Comparison with other state-of-the-art approaches

## Our algorithm (BGLR + kl-UCB)

- Hypotheses: bounded rewards, known  $T$ , known  $\Upsilon_T = o(\sqrt{T})$ , and “long enough” stationary sequences
- We obtain  $R_T = \mathcal{O}(K\sqrt{T\Upsilon_T \log(T)})$

# Comparison with other state-of-the-art approaches

## Our algorithm (BGLR + kl-UCB)

- Hypotheses: bounded rewards, known  $T$ , known  $\Upsilon_T = o(\sqrt{T})$ , and “long enough” stationary sequences
- We obtain  $R_T = \mathcal{O}(K\sqrt{T\Upsilon_T \log(T)})$

Two recent competitors use a similar assumption **but they both require prior knowledge of a lower-bound on the gaps**

## CUSUM-UCB

[Liu & Lee & Shroff, AAAI 2018]

- They obtain  $R_T = \mathcal{O}(K\sqrt{T\Upsilon_T \log(T/\Upsilon_T)})$

## M-UCB

[Cao & Zhen & Kveton & Xie, AISTATS 2019]

- They obtain  $R_T = \mathcal{O}(K\sqrt{T\Upsilon_T \log(T)})$

## 6. Numerical simulations

- ① (Stationary) Multi-armed bandits problems
- ② Piece-wise stationary multi-armed bandits problems
- ③ The BGLR test and its finite time properties
- ④ The BGLR-T + klUCB algorithm
- ⑤ Regret analysis
- ⑥ **Numerical simulations**

# Numerical simulations

We consider three problems with

- $K = 3$  arms, Bernoulli distributed
- $T = 5000$  time steps (fixed horizon)
- $\Upsilon_T = 4$  break-points (= 5 stationary sequences)  
Algorithms can use this prior knowledge of  $T$  and  $\Upsilon_T$
- 1000 independent runs, we plot the average regret

# Numerical simulations

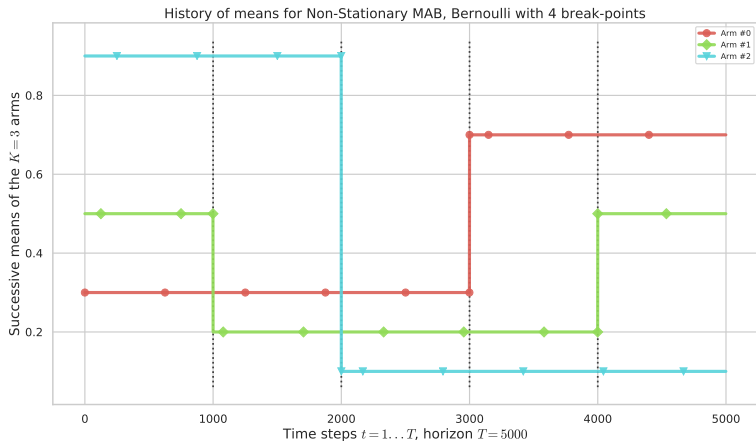
We consider three problems with

- $K = 3$  arms, Bernoulli distributed
- $T = 5000$  time steps (fixed horizon)
- $\Upsilon_T = 4$  break-points (= 5 stationary sequences)  
Algorithms can use this prior knowledge of  $T$  and  $\Upsilon_T$
- 1000 independent runs, we plot the average regret

## Reference

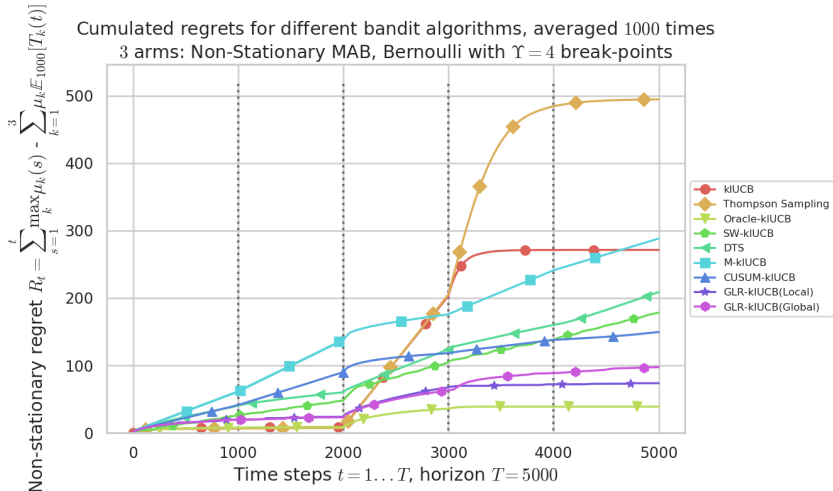
- We used my open-source Python library for simulations of multi-armed bandits problems, **SMPyBandits**  
↪ Published online at [SMPyBandits.GitHub.io](https://github.com/lilianbesson/smpybandits)
- More experiments are included in the long version of the paper!  
↪ pre-print on [HAL-02006471](https://arxiv.org/abs/1902.01575) and [arXiv:1902.01575](https://arxiv.org/abs/1902.01575)

# Problem 1: only local changes



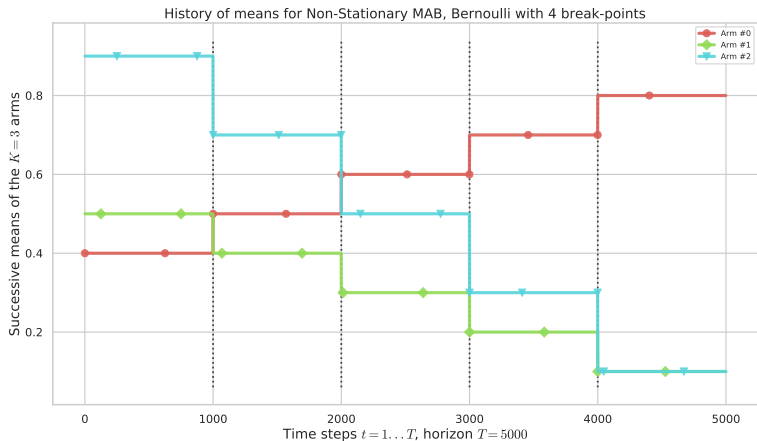
We plots the means:  $\mu_1(t)$ ,  $\mu_2(t)$ ,  $\mu_3(t)$ .

# Results on problem 1



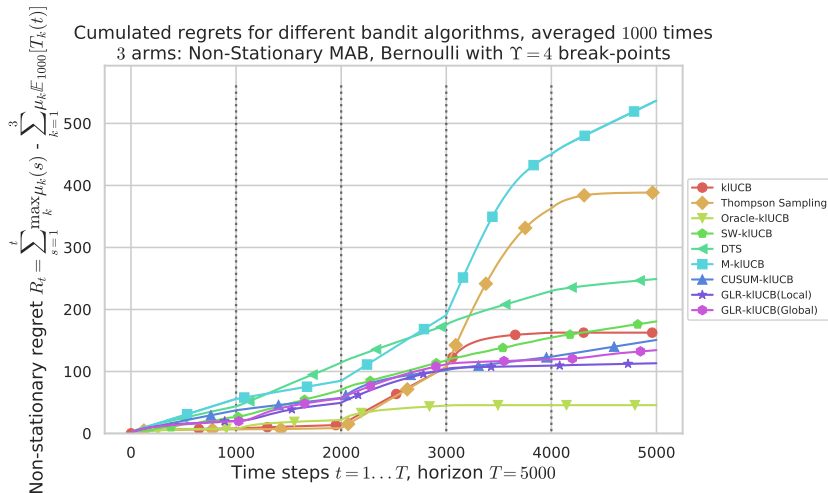
$\Rightarrow$  BGLR achieves the best performance among non-oracle algorithms 😎!

## Problem 2: only global changes



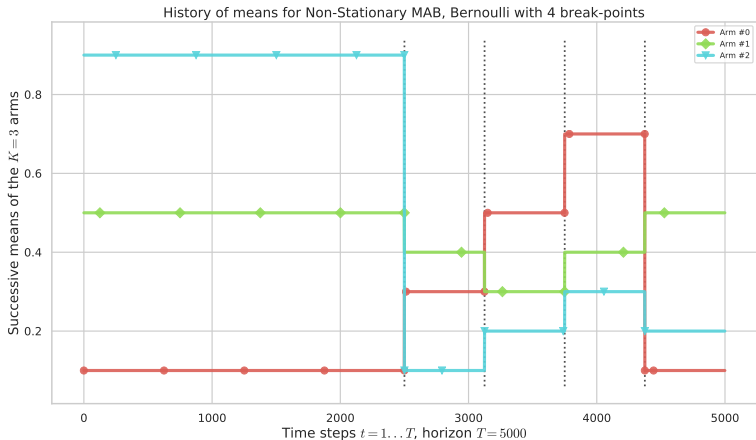


# Results on problem 2

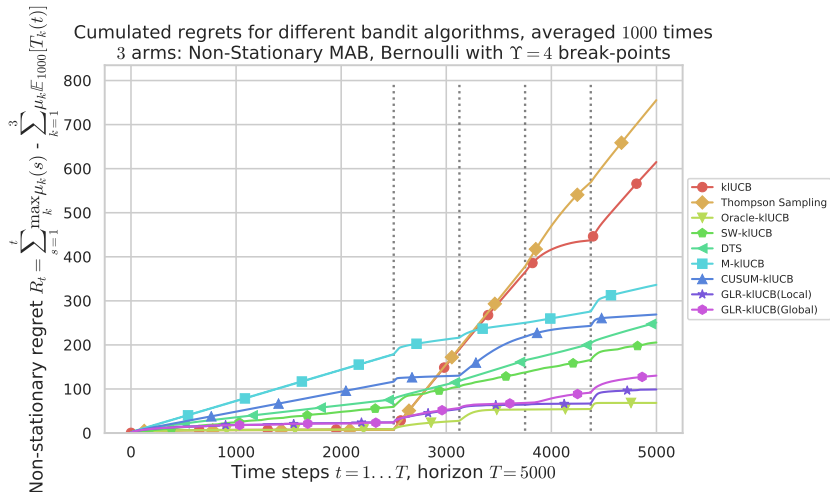


$\Rightarrow$  BGLR again achieves the best performance 🤖 !

# Pb 3: non-uniform lengths of stationary sequences



# Results on problem 3



$\Rightarrow$  BGLR achieves the best performance among non-oracle algorithms 😊 !

# Interpretation of the simulations (1/2)

## Conclusions in terms of regret

- Empirically we can check that the **BGLR test is efficient** 😊 :
  - it has a **low false alarm probability**,
  - it has a **small delay** if the stationary sequences are long enough.

And this is true even outside of the hypotheses of our analysis

- Using the kl-UCB indexes policy gives good performance 😊

⇒ Our algorithm (BGLR test + kl-UCB) is efficient

⇒ We verified that it obtains state-of-the-art performance!

## Interpretation of the simulations (2/2)

What about the efficiency in terms of **memory** and **time** complexity?

Memory: efficient 😊

Our algorithm is as efficient as other state-of-the-art strategies!

Memory cost =  $\mathcal{O}(K d_{\max})$  for  $K$  arms.

$(d_{\max} = \max_i \tau^i - \tau^{i+1} = \text{duration of the longer stationary sequence})$

## Interpretation of the simulations (2/2)

What about the efficiency in terms of **memory** and **time** complexity?

Memory: efficient 😊

Our algorithm is as efficient as other state-of-the-art strategies!

Memory cost =  $\mathcal{O}(Kd_{\max})$  for  $K$  arms.

Time: slow 😞 !

But it is too slow! Time cost =  $\mathcal{O}(Kd_{\max} \times t)$  at every time step  $t$ , so  $\mathcal{O}(Kd_{\max}T^2)$  in total.

↪ we proposed two numerical tweaks to speed it up

⇒ BGLR test + kl-UCB can be as fast as M-UCB or CUSUM-UCB

$(d_{\max} = \max_i \tau^i - \tau^{i+1} = \text{duration of the longer stationary sequence})$

# Summary

What we just presented...

- Stationary or **piece-wise stationary** Multi-Armed Bandits problems
- The efficient Bernoulli Generalized Likelihood Ratio test 😊
  - to detect break-points with **no false alarm** and **low delay**
  - for Bernoulli data, and can also be used for sub-Bernoulli data (any bounded distributions),
  - and does *not* need to know the amplitude of the break-point
- We can combine it with an efficient MAB policy: **BGLR + kl-UCB** 😊
- Its regret bound is  $R_T = \mathcal{O}(K\sqrt{T\Upsilon_T \log(T)})$  😊 (state-of-the-art)
- Our algorithm outperforms other efficient policies on numerical simulations 😊  
and BGLR + kl-UCB can be as fast as its best competitors.

# Conclusion

Thanks for your attention. 😊

Questions & Discussion ?