Efficient Change-Point Detection for Tackling Piecewise-Stationary Bandits

Lilian Besson, Emilie Kaufmann, Odalric-Ambrym Maillard, Julien Seznec

Efficient Change-Point Detection for Tackling Piecewise-Stationary Bandits

Lilian Besson
ENS Rennes, IRISA, Inria Rennes, France
LILIAN.BESSON@ENS-RENNES.FR

Emilie Kaufmann
Univ. Lille, CNRS, Inria, Centrale Lille, UMR 9189 - CRIStAL, F-59000 Lille, France
EMILIE.KAUFMANN@UNIV-LILLE.FR

Odalric-Ambrym Maillard
Univ. Lille, Inria, CNRS, Centrale Lille, UMR 9189 - CRIStAL, F-59000 Lille, France
ODALRIC.MAILLARD@INRIA.FR

Julien Seznec
Inria and Lelivrescolaire.fr *Éditions*
JULIEN.SEZNEC@INRIA.FR

Editor:

Abstract

We introduce GLR-klUCB, a novel algorithm for the piecewise *i.i.d.* non-stationary bandit problem with bounded rewards. This algorithm combines an efficient bandit algorithm, klUCB, with an efficient, *parameter-free*, changepoint detector, the Bernoulli Generalized Likelihood Ratio Test, for which we provide new theoretical guarantees of independent interest. Unlike previous non-stationary bandit algorithms using a change-point detector, GLR-klUCB does not need to be calibrated based on prior knowledge on the arms' means. We prove that this algorithm can attain a $\mathcal{O}(\sqrt{TA\Upsilon_T \ln(T)})$ regret in T rounds on some "easy" instances, where A is the number of arms and Υ_T the number of change-points, *without prior knowledge of* Υ_T. In contrast with recently proposed algorithms that are agnostic to Υ_T, we perform a numerical study showing that GLR-klUCB is also very efficient in practice, beyond easy instances.

Keywords: Multi-Armed Bandits; Change Point Detection; Non-Stationary Bandits.

1. Introduction

Multi-Armed Bandit (MAB) problems form a well-studied class of sequential decision making problems, in which an agent repeatedly chooses an action $A_t \in \{1, \dots, A\}$ or "arm" among a set of A arms (Robbins, 1952; Lattimore and Szepesvári, 2019). In the most standard version of the stochastic bandit model, each arm a is associated with an *i.i.d.* sequence of rewards $(X_{a,t})$ that follow some distribution of mean μ_a. Upon selecting arm A_t, the agent receives the reward $X_{A_t,t}$. Her goal is to design a sequential arm selection strategy that maximizes the expected sum of these rewards, or, equivalently, that minimizes *regret*, defined as the difference between the total sum of rewards of an oracle strategy always selecting the arm with largest mean and that of her strategy.

Stochastic bandits were historically introduced as a simple model for clinical trials, where arms correspond to some treatments with unknown efficacy (Thompson, 1933). More recently, MAB models have

been proved useful for other applications, such as cognitive radio, where arms can model the vacancy of radio channels, or parameters of a dynamically configurable radio hardware (Maghsudi and Hossain, 2016; Bonnefoi et al., 2017; Kerkouche et al., 2018). Another application is the design of recommender systems, where arms model the popularity of different items (*e.g.*, news recommendation, Li et al. (2010)). In all these applications, the assumption that the arms distributions *do not evolve over time* is often violated: patients adapt to medical treatments, new devices can enter or leave the radio network, hence impacting the availability of radio channels, and the popularity of items is subject to trends. This aroused interest in how to take *non-stationary* aspects into account within a multi-armed bandit model.

As a possible way to cope with non-stationarity, the *piecewise stationary MAB* was introduced by Kocsis and Szepesvári (2006). In this model, the (random) reward of arm a at round t has some mean $\mu_a(t)$ and the regret is measured with respect to the *current* best arm $a_t^\star = \arg\max_a \mu_a(t)$. It is furthermore assumed that there are relatively few *breakpoints* between which the $\mu_a(t)$ remain constant for all arms a. Despite many approaches already proposed for minimizing regret under this model (see Section 2), research on this topic has been very active in the last years, notably in two different directions. The first is the design of a good combination of a bandit algorithm and a changepoint detector (CPD) supported by regret guarantees and enjoying *good empirical performance* (Liu et al., 2018; Cao et al., 2019). These algorithms share with many others the downside of having to know the number of breakpoints Υ_T to guarantee state-of-the-art regret. The second direction proposes algorithms that achieve *optimal regret without the knowledge of* Υ_T (Auer et al., 2019b; Chen et al., 2019), but without an emphasis on actual practical performance (yet).

In this paper, we propose the first algorithm based on a change-point detector that is very efficient in practice and *does not require the knowledge of* Υ_T to provably achieve optimal regret, at least on some "easy" instances, with few breakpoints of large enough magnitude. An interesting feature of our algorithm compared to other CPD-based algorithms is that it *does not require any prior knowledge on the arms means*. Like CUSUM (Liu et al., 2018) and M-UCB (Cao et al., 2019), our algorithm relies on combining a standard bandit algorithm with a changepoint detector. For the bandit component, we propose the use of the klUCB (Cappé et al., 2013) which is known to outperform UCB (Auer et al., 2002a) used in previous works. For the changepoint detector, we suggest using the Bernoulli Generalized Likelihood Ratio Test (GLRT), for which we provide new non-asymptotic properties that are of independent interest. This choice is particularly appealing because unlike the changepoint detectors used in previous works, the Bernoulli GLRT *does not require a lower bound on the minimal amount of change to detect*, which leads to a bandit algorithm which is agnostic to the arms' means. In contrast, both CUSUM and M-UCB require the knowledge of the smallest magnitude of a change in the arm's mean.

In this work we jointly investigate two versions of GLR-klUCB, one using *global restarts* (resetting the history of *all* arms once a changepoint is detected on one of them) and one using *local restarts* (resetting the history of an arm each time a changepoint is detected on that arm). We prove that GLR-klUCB based on global restart achieves a $\mathcal{O}(\sqrt{TA\Upsilon_T \ln(T)}/(\Delta^{\text{change}})^2)$ regret where Δ^{change} is the smallest magnitude of a breakpoint. If all breakpoints have a large magnitude, this $\mathcal{O}(\sqrt{TA\Upsilon_T \ln(T)})$ regret is matching the lower bound of Seznec et al. (2020) up to a $\sqrt{\ln(T)}$ factor. Following a similar analysis, we prove slightly weaker results for the version based on local restart. Numerical simulations in Section 6 reveal that these two versions are both competitive in practice with state-of-the-art algorithms.

To summarize, our contributions are the following: (1) A non-asymptotic analysis of the Bernoulli-GLR changepoint detector. (2) A new bandit algorithm for the piecewise stationary setting based on this test that needs no prior knowledge on the number of change-points and no information on the arms means

to attain near-optimal regret. (3) An extensive numerical study illustrating the good performance of two versions of GLR-klUCB compared to other algorithms with state-of-the-art regret.

Outline The paper is structured as follows. We introduce the model and review related works in Section 2. In Section 3, we present some properties of the Bernoulli-GLR changepoint detector. We introduce the two variants of GLR-klUCB in Section 4. In Section 5 we present regret upper bounds for GLR-klUCB for Global Restart and sketch our regret analysis. Numerical experiments are presented in Section 6.

2. Setup and Related Work

A *piecewise stationary bandit model* is characterized by a stream of (random) rewards $(X_{a,t})_{t\in\mathbb{N}^\star}$ associated to each arm $a \in \{1,\dots,A\}$. We assume that the rewards are bounded in a known range, and without loss of generality we assume that $X_{a,t} \in [0,1]$. We denote by $\mu_a(t) := \mathbb{E}[X_{a,t}]$ the mean reward of arm a at round t. At each round t, a decision maker has to select an arm $A_t \in \{1,\dots,A\}$, based on past observation and receives the corresponding reward $r(t) = X_{A_t,t}$. At time t, we denote by $a_t^\star$ an arm with maximal expected reward, *i.e.*, $\mu_{a_t^\star}(t) = \max_a \mu_a(t)$, called an optimal arm.

A policy π chooses the next arm to play based on the sequence of past plays and obtained rewards. The performance of π is measured by its (dynamic) *regret*, the difference between the expected reward obtained by an oracle policy playing an optimal arm $a_t^\star$ at time t, and that of the policy π:

$$R_T^\pi = \mathbb{E}\left[\sum_{t=1}^{T} \left(\mu_{a_t^\star}(t) - \mu_{A_t}(t)\right)\right].$$

In the piecewise *i.i.d.* model, we furthermore assume that there is a (relatively small) number of *breakpoints*, denoted by $\Upsilon_T := \sum_{t=1}^{T-1} \mathbb{1}\left(\exists a \in \{1,\dots,A\} : \mu_t(a) \neq \mu_{t+1}(a)\right)$. We define the k-th breakpoint by $\tau^{(k)} = \inf\{t > \tau^{(k-1)} : \exists a : \mu_a(t) \neq \mu_a(t+1)\}$ with $\tau^{(0)} = 1$. Hence for $t \in [\tau^{(k)}+1, \tau^{(k+1)}]$, the rewards $(X_{a,t})$ associated to all arms are *i.i.d.*, with mean denoted by $\mu_a^{(k)}$. The *magnitude* of a breakpoint k is defined as $\Delta^{c,(k)} =: \max_{a=1,\dots,A} \left|\mu_a^{(k)} - \mu_a^{(k-1)}\right|$ and we let $\Delta^{\text{change}} =: \min_{k=1,\dots,\Upsilon_T} \Delta^{c,(k)}$.

Note than when a breakpoint occurs, we do not assume that all the arms means change, but that *there exists* an arm which experiences a *changepoint*, i.e. whose mean satisfies $\mu_a(t) \neq \mu_a(t+1)$. Depending on the application, many scenarios can be meaningful: changes occurring for all arms simultaneously (due to some exogenous event), or only a few arms that experience a changepoint in each breakpoint. Letting C_T denote the total number of changepoints before horizon T, we have $C_T \in \{\Upsilon_T, \dots, A\Upsilon_T\}$.

2.1 An Adversarial View on Non-Stationary Bandits

A natural way to cope with non-stationary is to model the decision making problem as an *adversarial bandit problem* (Auer et al., 2002b), under which the rewards are arbitrarily generated. For adversarial environments, the most studied performance measure is the pseudo-regret, which compares the accumulated reward of a given strategy with that of the best fixed-arm policy. However in some changing environments it is more natural to measure regret against the best *sequence of actions*. Auer et al. (2002b) propose the Exp3.S algorithm, that achieves a regret of $\mathcal{O}(\sqrt{A\Upsilon_T T \ln(T)})$ against the best sequence of actions with $\Upsilon_T - 1$ switches. This regret rate matches the corresponding lower bound. Exp3.S is simple to implement and run with time and space complexity $\mathcal{O}(A)$ but requires the knowledge of T and Υ_T to reach near-minimax optimal regret rate.

When the piecewise i.i.d. assumption holds (with Υ_T stationary part), the best sequence of actions with $\Upsilon_T - 1$ switches corresponds to the optimal oracle policy. The minimax optimal rate against piecewise i.i.d. rewards sequences is also $\mathcal{O}(\sqrt{A\Upsilon_T T})$. It is similar to the fixed-arm case where the adversarial pseudo-regret rate and the minimax stochastic rate are the same ($\mathcal{O}(\sqrt{AT})$, Audibert and Bubeck (2010)). However, in the fixed-arm setup, the stochastic stationary assumption allows a *problem-dependent analysis*: some algorithms (e.g. UCB or Thomson Sampling) suffer $\mathcal{O}(\ln T/\Delta_i)$ regret on each arm i with a reward gap of Δ_i compared to the best arm. When Δ_i is large enough, this problem-dependent guarantee is much better than the $\mathcal{O}(\sqrt{T})$ minimax rate. Unfortunately, in the piecewise i.i.d. setup, Garivier and Moulines (2011) show that any algorithm whose regret is $R_T(\boldsymbol{\mu})$ on a stationary bandit instance $\boldsymbol{\mu}$ is such that there exists a piecewise stationary instance $\boldsymbol{\mu}'$ with at most two breakpoints such that $R_T(\boldsymbol{\mu}') \geq cT/R_T(\boldsymbol{\mu})$, for some absolute constant c. In particular, this implies that an algorithm that attains $\mathcal{O}(\sqrt{T})$ regret for *any* piecewise stationary bandit model has no hope to reach $\mathcal{O}(\ln(T))$ regret on easy instances. The intuition behind this result is that if an algorithm achieves very low regret on a specific problem then it has to pull suboptimal arms very scarcely. By doing so, it is unable to perform well on a similar problem where the identified suboptimal arms' surreptitiously increase to become optimal. Therefore, it is important to pull every arm often enough (e.g. every $\mathcal{O}(\sqrt{AT/\Upsilon_T})$ rounds) even when one is clearly underperforming.

Nevertheless, the piecewise i.i.d. bandit problem remained actively studied since the seminal paper of Auer et al. (2002b). The outcome of this line of work is threefold. First, designing strategies leveraging tools from the stochastic MAB can greatly improve the empirical performance compared to adversarial algorithms like Exp3.S. Second, we would like to build strategies that are near-optimal without the knowledge of Υ_T[1] (unlike Exp3.S). Third, it is possible to further restrain the setup to make the problem-dependent analysis possible by forbidding the aforementioned surreptitious increase of one arm. For instance, Mukherjee and Maillard (2019) consider the "global change" setup in which all the arms change significantly when a breakpoint occurs. Seznec et al. (2020) consider the rotting setup where the arms cannot increase. In both cases, the authors proved a logarithmic problem-dependent upper bound on the regret of their algorithms.

In this paper, we bring theoretical and empirical contributions to the two first points. We also discuss a possible adaptation of GLR-klUCB that may recover logarithmic regret in the easier setups of Mukherjee and Maillard (2019); Seznec et al. (2020).

2.2 Algorithms Exploiting the Stochastic Assumption

The piecewise stationary bandit model was first studied by Kocsis and Szepesvári (2006); Yu and Mannor (2009); Garivier and Moulines (2011). It is also known as *switching* (Mellor and Shapiro, 2013) or *abruptly changing stationary* (Wei and Srivastava, 2018) environment. Most approaches exploiting the stochastic assumption combine a bandit algorithm with a mechanism to *forget old rewards*. We make a distinction between *passively adaptive strategies*, which use a fixed forgetting mechanism, and *actively adaptive strategies*, for which this mechanism is also data-dependent.

Passively Adaptive Strategies A simple mechanism to forget the past consists in either discounting rewards (multiplying past reward by γ^n where n is the time elapsed since that reward was collected, for a discount factor $\gamma \in (0, 1)$), or using a sliding window (only the rewards gathered in the τ last rounds

1. In the adversarial setup, Υ_T appears in the definition of the pseudo-regret, hence it is quite natural that the learner knows this parameter. In the piecewise i.i.d. setup, the regret is against the optimal oracle policy which is defined independently of Υ_T.

are taken into account, for a window size τ). Those strategies are passively adaptive as the discount factor or the window size are *fixed*, and can be tuned as a function of T and Υ_T to achieve a certain regret bound. Discounted UCB (D-UCB) was proposed by Kocsis and Szepesvári (2006) and analyzed by Garivier and Moulines (2011), who prove a $\mathcal{O}(A\sqrt{\Upsilon_T T \ln(T)})$ regret bound, if $\gamma = 1 - \sqrt{\Upsilon_T/T}/4$. The same authors proposed the Sliding-Window UCB (SW-UCB) and prove a $\mathcal{O}(A\sqrt{\Upsilon_T T \ln(T)})$ regret bound, if $\tau = 2\sqrt{T\ln(T)/\Upsilon_T}$. More recently, Raj and Kalyani (2017) proposed the Discounted Thompson Sampling (DTS) algorithm, which performs well on the reported experiments with $\gamma = 0.75$, but no theoretical guarantees are given for this particular tuning. The REXP3 algorithm (Besbes et al., 2014) is another passively adaptive strategy that is based on (non-adaptive) restarts of the Exp3 algorithm (Auer et al., 2002b). REXP3 is analyzed in terms of a different measure of interest, the total variation budget V_T which satisfies $\Delta^{\text{change}}\Upsilon_T \leq V_T \leq \Upsilon_T$. REXP3 is proved to have a $\mathcal{O}((A \ln A)^{1/3} V_T^{1/3} T^{2/3})$ regret, which translates to a sub-optimal rate in our setting.

Actively Adaptive Strategies The first *actively adaptive* strategy is Windowed-Mean Shift (Yu and Mannor, 2009), which combines any bandit policy with a change point detector which performs *adaptive restarts* of the bandit algorithm. However, this approach does not apply to our setting as it takes into account side observations. Another line of research on actively adaptive algorithms uses a Bayesian point of view, where the process of change point occurrences is modeled and tracked using Bayesian updates. A Bayesian Change-Point Detection (CPD) algorithm is combined with Thompson Sampling by Mellor and Shapiro (2013), and more recently in the Memory Bandit algorithm of Alami et al. (2017). Since none of these algorithms have theoretical guarantees and they are designed for a different setup, we do not include them in our experiments.

Our closest competitors rather use frequentist CPD algorithms combined with a bandit algorithm. The first algorithm of this flavor, Adapt-EVE algorithm (Hartland et al., 2006) uses a Page-Hinkley test and the UCB policy, but no theoretical guarantees are given. Exp3.R (Allesiardo and Féraud, 2015; Allesiardo et al., 2017) combines a CPD with Exp3, and the history of all arms are reset as soon as a sub-optimal arm is detected to become optimal and it achieves a $\mathcal{O}(\Upsilon_T A\sqrt{T\ln(T)})$ regret (without the knowledge of Υ_T). More recently, CUSUM-UCB (Liu et al., 2018) and Monitored UCB (M-UCB, Cao et al. (2019)) have achieved $\mathcal{O}(\sqrt{\Upsilon_T A T \ln(T)})$ regret, when Υ_T is known.

CUSUM-UCB is based on a variant of a two-sided CUSUM test, that uses the first M samples from one arm to compute an initial average, and then detects whether a drift of size larger than ε occurred from this value by checking whether a random walk based on the remaining observations crosses a threshold h. It requires the tuning of three parameters, M, ε and h. CUSUM-UCB performs *local restarts* using this test, to reset the history of *one arm* for which the test detects a change. M-UCB uses a simpler test, based on the w most recent observations from an arm: a change is detected if the absolute difference between the empirical means of the first and second halves of those w observations exceeds a threshold h. It requires the tuning of two parameters, w, and h. M-UCB performs *global restarts* using this test, to reset the history of *all arms* whenever the test detects a change on one of them.

On a stationary batch, a UCB index algorithm tends to pull each arm at a logarithmic rate asymptotically. According to the aforementioned Garivier and Moulines (2011)'s lower bound, it is not enough to shield against increases of the suboptimal arms' values. Thus, CPD-based algorithms usually rely on additional *forced exploration*: each arm is pulled regularly either according to a constant probability of uniform exploration (Liu et al., 2018) or according to a deterministic scheme (Cao et al., 2019). To avoid linear regret, the total budget dedicated to this forced exploration is tuned with the knowledge of T and Υ_T (e.g. $\mathcal{O}\left(\sqrt{A\Upsilon_T T}\right)$). Mukherjee and Maillard (2019) suggest canceling the forced exploration when all the arms change at the same rounds. Indeed, in that case, we can aim to detect the changes on any

arms' sequences and then restart all the arms' indexes. Similarly, Seznec et al. (2020) do not use forced exploration and study the Rotting Adaptive Window UCB (RAW-UCB) - a UCB index policy with an adaptive window designed for non-increasing sequences of rewards. Both these algorithms can get logarithmic regret on some problem instances and, therefore, cannot be minimax optimal for the general piecewise i.i.d bandit problem, which is our focus in this paper.

2.3 Knowledge of the Number of Breakpoints

All algorithms mentioned above for the general piecewise stationary bandit problem require some tuning that should depend on Υ_T to attain state-of-the-art $\mathcal{O}(\sqrt{\Upsilon_T AT \ln(T)})$ regret. Two algorithms achieving this regret *without the knowledge of* Υ_T were recently proposed: Ada-ILTCB$^+$(Chen et al., 2019) and AdSwitch (Auer et al., 2019b), that also rely on detecting non-stationarities (Auer et al., 2019a). While the former is tailored for the more general adversarial and contextual setting, the latter is specifically proposed for the piecewise i.i.d. model.

AdSwitch is an elimination strategy based on confidence interval (like Improved UCB (Auer and Ortner, 2010)) with global restarts when a change-point is detected on one arm. AdSwitch performs an *adaptive forced exploration* scheme on the eliminated arms which adds two main components to the aforementioned uniform random exploration. First, AdSwitch uses a counter l (initialized at 1) for the number of detected changes by the CPD subroutine as a proxy for Υ_T to tune the random exploration probability on each eliminated arms at $\mathcal{O}\left(\sqrt{l/KT}\right)$. Second, AdSwitch also selects at random a change size Δ on a geometric grid, with a probability proportional to Δ. The arm is then pulled $\mathcal{O}\left(1/\Delta^2\right)$ consecutive pulls to check if there is a change of size Δ. The consecutive sampling is particularly helpful theoretically to analyze the algorithm when the CPD misses some breakpoints.

However, unlike in our work, the underlying changepoint detectors used in AdSwitch have not been optimized for efficiency or tractability[2]. Neither (Chen et al., 2019) nor (Auer et al., 2019b) report simulation to assess the empirical or numerical efficiency of their algorithms. In this paper, we include (a tweaked, tractable version of) AdSwitch in our experiments for short horizons.

An alternative idea to adapt to Υ_T is the "Bandit over Bandit" approach of Cheung et al. (2019), which uses an exponential weights algorithm for expert aggregation on top of several copies of Sliding-Windows UCB with different (fixed) window size. Yet this approach does not yield optimal regret.

3. The Bernoulli GLR Change Point Detector

Sequential changepoint detection has been extensively studied in the statistical community (see, *e.g.*, Basseville et al. (1993); Jie and Gupta (2000); Wu (2007))). In this article, we are interested in detecting changes on the mean of a probability distribution with bounded support. Assume that we collect independent samples $X_1, X_2, \ldots$ all from some distribution supported in $[0, 1]$. We want to discriminate between two possible scenarios: all the samples come from distributions that have a common mean μ_0, or there exists a *changepoint* $\tau \in \mathbb{N}^\star$ such that $X_1, \ldots, X_\tau$ have some mean μ_0 and $X_{\tau+1}, X_{\tau+2}, \ldots$ have a different mean $\mu_1 \neq \mu_0$. A sequential changepoint detector is a stopping time $\hat{\tau}$ with respect to the filtration $\mathcal{F}_t = \sigma(X_1, \ldots, X_t)$ such that $(\hat{\tau} < \infty)$ means that we reject the hypothesis $\mathcal{H}_0 : (\exists \mu_0 \in [0, 1] : \forall i \in \mathbb{N}, \mathbb{E}[X_i] = \mu_0)$.

2. Indeed, at each time step t, the test employed by AdSwitch requires $\Theta(At^3)$ operations, resulting in a very expensive $\Theta(AT^4)$ time complexity when compared to $\Theta(AT)$ for simple algorithms like UCB and $\Theta(AT^2)$ for other adaptive approaches based on scan statistics like GLR-klUCB.

Generalized Likelihood Ratio tests have been used for a very long time (see, e.g. Wilks (1938)) and were for instance studied for changepoint detection by Siegmund and Venkatraman (1995). Exploiting the fact that bounded distributions are $(1/4)$-sub-Gaussian (*i.e.*, have a moment generating function dominated by that of a Gaussian with the same mean and variance $1/4$), the (Gaussian) GLRT, recently studied in depth by Maillard (2019), can be used for our problem. We propose instead to exploit the fact that bounded distributions are also dominated by Bernoulli distributions. We call a *sub-Bernoulli distribution* any distribution ν that satisfies $\ln \mathbb{E}_{X\sim\nu}\left[e^{\lambda X}\right] \leq \phi_\mu(\lambda)$ with $\mu = \mathbb{E}_{X\sim\nu}[X]$ and $\phi_\mu(\lambda) = \ln(1-\mu+\mu e^\lambda)$ is the log moment generating function of a Bernoulli distribution with mean μ. Lemma 1 of Cappé et al. (2013) establishes that any bounded distribution supported in $[0,1]$ is a sub-Bernoulli distribution.

3.1 Presentation of the test

If the samples (X_t) were all drawn from a Bernoulli distribution, our changepoint detection problem would reduce to a parametric sequential test of $\mathcal{H}_0 : (\exists \mu_0 : \forall i \in \mathbb{N}, X_i \overset{\text{i.i.d.}}{\sim} \mathcal{B}(\mu_0))$ against the alternative $\mathcal{H}_1 : (\exists \mu_0 \neq \mu_1, \tau \in \mathbb{N}^\star : \ X_1, \dots, X_\tau \overset{\text{i.i.d.}}{\sim} \mathcal{B}(\mu_0)$ and $X_{\tau+1}, X_{\tau+2}, \dots \overset{\text{i.i.d.}}{\sim} \mathcal{B}(\mu_1))$. The (log)-Generalized Likelihood Ratio statistic for this test is defined by

$$\mathrm{GLR}(n) := \ln \frac{\sup\limits_{\mu_0,\mu_1,\tau<t} \ell(X_1,\dots,X_n;\mu_0,\mu_1,\tau)}{\sup\limits_{\mu_0} \ell(X_1,\dots,X_n;\mu_0)},$$

where $\ell(X_1,\dots,X_n;\mu_0)$ and $\ell(X_1,\dots,X_n;\mu_0,\mu_1,\tau)$ denote the likelihood of the first n observations under a model in $\mathcal{H}_0$ and $\mathcal{H}_1$. High values of this statistic tend to indicate rejection of $\mathcal{H}_0$. Using the form of the likelihood for Bernoulli distribution, this statistic can be written with the binary relative entropy kl,

$$\mathrm{kl}(x,y) := x \ln\left(\frac{x}{y}\right) + (1-x)\ln\left(\frac{1-x}{1-y}\right). \tag{1}$$

Indeed, one can show that $\mathrm{GLR}(n) = \sup_{s\in[1,n]} Z_{s,n}$ where $Z_{s,n} = s \times \mathrm{kl}\left(\hat{\mu}_{1:s}, \hat{\mu}_{1:n}\right) + (n-s) \times \mathrm{kl}\left(\hat{\mu}_{s+1:n}, \hat{\mu}_{1:n}\right)$ and for $k \leq k'$, $\hat{\mu}_{k:k'}$ denotes the average of the observations collected between the instants k and k'. This motivates the definition of the Bernoulli GLR change point detector.

Definition 1 *The Bernoulli GLR change point detector with threshold function $\beta(n,\delta)$ is*

$$\hat{\tau}_\delta := \inf\left\{ n \in \mathbb{N}^\star : \sup_{s\in[1,n]} \left[s \times \mathrm{kl}\left(\hat{\mu}_{1:s}, \hat{\mu}_{1:n}\right) + (n-s) \times \mathrm{kl}\left(\hat{\mu}_{s+1:n}, \hat{\mu}_{1:n}\right) \right] \geq \beta(n,\delta) \right\}. \tag{2}$$

Asymptotic properties of the GLR for changepoint detection have been studied by Lai and Xing (2010) for Bernoulli distributions and more generally for one-parameter exponential families, for which the GLR test is defined as in (2) but with $\mathrm{kl}(x,y)$ replaced by the Kullback-Leibler divergence between two elements in that exponential family that have mean x and y. For example, the Gaussian GLR studied by Maillard (2019) corresponds to (2) with $\mathrm{kl}(x,y) = 2(x-y)^2$ when the variance is set to $\sigma^2 = 1/4$, and non-asymptotic properties of this test are given for any $(1/4)$-sub-Gaussian samples.

In the next section, we provide new non-asymptotic results about the Bernoulli GLR test under the assumption that the samples (X_t) come from a sub-Bernoulli distribution, which holds for any distribution supported in $[0,1]$. Note that Pinsker's inequality gives that $\mathrm{kl}(x,y) \geq 2(x-y)^2$, hence the Bernoulli GLR may stop earlier that the Gaussian GLR based on the quadratic divergence $2(x-y)^2$.

GLR versus confidence-based CPD An alternative to the GLR also based on scan statistics, used by Mukherjee and Maillard (2019) consists in building individual confidence intervals for the mean in each segment, of the form

$$\left[\hat{\mu}_{1:s} \pm \sqrt{\frac{\tilde{\beta}(s,\delta)}{2s}}\right] \quad \text{and} \quad \left[\hat{\mu}_{s+1:n} \pm \sqrt{\frac{\tilde{\beta}(n-s,\delta)}{2(n-s)}}\right]$$

and report that there is a change point if there exists s such that these confidence interval are disjoint, i.e.

$$\hat{\tau}'_\delta \;=\; \inf\left\{ n \in \mathbb{N}^\star : \exists s \in [1,n], \left|\hat{\mu}_{1:s} - \hat{\mu}_{s+1:n}\right| > \sqrt{\frac{\tilde{\beta}(s,\delta)}{2s}} + \sqrt{\frac{\tilde{\beta}(n-s,\delta)}{2(n-s)}} \right\}.$$

By measuring distances with the appropriate KL divergence function, the Bernoulli GLR test better exploits the geometry of (sub-)Bernoulli distributions.

3.2 Properties of the Bernoulli GLR

In Lemma 2 below, we propose a choice of the threshold function $\beta(n,\delta)$ under which the probability that there exists a *false alarm* under *i.i.d.* data is small. To define β, we introduce the function $\mathcal{T}$, originally introduced by Kaufmann and Koolen (2018),

$$\mathcal{T}(x) \;:=\; 2\tilde{h}\left(\frac{h^{-1}(1+x) + \ln(2\zeta(2))}{2}\right) \tag{3}$$

where for $u \geq 1$ we define $h(u) = u - \ln(u)$ and its inverse $h^{-1}(u)$. And for any $x \geq 0$, $\tilde{h}(x) = e^{1/h^{-1}(x)} h^{-1}(x)$ if $x \geq h^{-1}(1/\ln(3/2))$ and $\tilde{h}(x) = (3/2)(x - \ln(\ln(3/2)))$ otherwise. The function $\mathcal{T}$ is easy to compute numerically. Its use for the construction of concentration inequalities that are uniform in time is detailed in Kaufmann and Koolen (2018), where tight upper bounds on the function $\mathcal{T}$ are also given: $\mathcal{T}(x) \simeq x + 4\ln\left(1 + x + \sqrt{2x}\right)$ for $x \geq 5$ and $\mathcal{T}(x) \sim x$ when x is large. The proof of Lemma 2 is given in Appendix B.1.

Lemma 2 *Assume that there exists $\mu_0 \in [0,1]$ such that $\mathbb{E}[X_t] = \mu_0$ and that $X_i \in [0,1]$ for all i. Then the Bernoulli GLR test satisfies $\mathbb{P}_{\mu_0}(\hat{\tau}_\delta < \infty) \leq \delta$ with the threshold function*

$$\beta(n,\delta) = 2\mathcal{T}\left(\frac{\ln(3n\sqrt{n}/\delta)}{2}\right) + 6\ln(1+\ln(n)). \tag{4}$$

Another key feature of a changepoint detector is its *detection delay* under a model in which a change from μ_0 to μ_1 occurs at time τ. We already observed that from Pinsker's inequality, the Bernoulli GLR stops earlier than a Gaussian GLR. Hence, one can leverage some techniques from Maillard (2019) to upper bound the detection delay of the Bernoulli GLR. Letting $\Delta = |\mu_0 - \mu_1|$, one can essentially establish that for τ larger than $(1/\Delta^2)\ln(1/\delta)$ (*i.e.*, enough samples before the change), the delay can be of the same magnitude (*i.e.*, enough samples after the change). In our bandit analysis to follow, the detection delay will be crucially used to control the probability of the "good event" that all the changepoints are detected within a reasonable delay (Lemma 8 and 15).

3.3 Practical considerations

Lemma 2 provides the first control of false alarm for the Bernoulli GLR employed for bounded distributions. However, the threshold (4) is not fully explicit as the function $\mathcal{T}(x)$ can only be computed numerically. Note that for sub-Gaussian distributions, results from Maillard (2019) show that the smaller and more explicit threshold $\beta(n,\delta) = \left(1+\frac{1}{n}\right)\ln\left(\frac{3n\sqrt{n}}{\delta}\right)$, can be used to prove an upper bound of δ for the false alarm probability of the GLR, with quadratic divergence $\mathrm{kl}(x,y) = 2(x-y)^2$. For the Bernoulli GLR, numerical simulations suggest that the threshold (4) is a bit conservative, and in practice we recommend to keep only the leading term and use $\beta(n,\delta) = \ln(n\sqrt{n}/\delta)$.

Also note that, as any test based on scan-statistics, the GLR can be costly to implement: at every time step, it considers all previous time steps as a possible position for a changepoint. Thus, in practice the following adaptation may be interesting, based on down-sampling the possible time steps:

$$\tilde{\tau}_\delta = \inf\left\{ n \in \mathcal{N} : \sup_{s\in\mathcal{S}_n} Z_{s,n} \geq \beta(n,\delta) \right\}, \tag{5}$$

for any strict subsets $\mathcal{N} \subseteq \mathbb{N}$ and $\mathcal{S}_n \subset \{1,\dots,n\}$. Following the proof of Lemma 2, we can easily see that this variant enjoys the exact same false-alarm control. However, the detection delay may be slightly increased. Our experiments reveal that the price in terms of regret of the speed-up is negligible.

4. The GLR-klUCB Algorithm

GLR-klUCB (Algorithm 1) combines the klUCB algorithm (Cappé et al., 2013), known to be optimal for Bernoulli bandits, with the Bernoulli GLR changepoint detector introduced in Section 3. It also needs a third ingredient: some extra exploration to ensure each arm is sampled enough and changes can also be detected on arms currently under-sampled by klUCB. This forced exploration is parameterized by a sequence $\alpha = (\alpha_k)_{k\in\mathbb{N}}$ of exploration frequencies $\alpha_k \in (0,1)$. GLR-klUCB can be used in any bandit model with bounded rewards, and is expected to be very efficient for Bernoulli distributions.

The GLR-klUCB algorithm can be viewed as a klUCB algorithm allowing for some *restarts* on the different arms. A restart happens when the Bernoulli GLR changepoint detector detects a change on the arm that has been played (line 9). To be fully specific, $\mathrm{GLR}_\delta(Y_1,\dots,Y_n) = \mathrm{True}$ if and only if the GLR statistic associated to those n samples,

$$\sup_{1\leq s\leq n} \left[s \times \mathrm{kl}(\hat{Y}_{1:s}, \hat{Y}_{1:n}) + (n-s) \times \mathrm{kl}(\hat{Y}_{s+1:n}, \hat{Y}_{1:n}) \right],$$

is larger than the threshold $\beta(n,\delta)$ defined in (4), or $\beta(n,\delta) = \ln(n^{3/2}/\delta)$, as recommended in Section 3.3. Each restart (on any arm) triggers a new episode and we denote by k_t the number of episodes started after t samples (i.e. the index of the on-going episode at time t).

Letting $\tau_a(t)$ denote the instant of the last restart that happened for arm a before time t, $n_a(t) = \sum_{s=\tau_a(t)+1}^{t} \mathbb{1}(A_s = a)$ the number of selections of arm a and $\hat{\mu}_a(t) = (1/n_a(t)) \sum_{s=\tau_a(t)+1}^{t} X_{a,s}\mathbb{1}(A_s = a)$ the empirical mean (if $n_a(t) \neq 0$), the index used by the algorithm is defined as

$$\mathrm{UCB}_a(t) := \max\left\{ q : n_a(t) \times \mathrm{kl}\left(\hat{\mu}_a(t), q\right) \leq f(t-\tau_a(t)) \right\}. \tag{6}$$

Algorithm 1 presents two variants of GLR-klUCB, one using *local restarts* (line 11), and one using *global restarts* (line 13). Under local restarts, in the general case the times $\tau_a(t)$ are not equal for all arms, hence the index policy associated to (6) is *not* a standard UCB algorithm, as each index uses a *different exploration rate*. One can highlight that in the CUSUM-UCB algorithm, which is the only existing

Algorithm 1: GLR-klUCB (**Local** or **Global** restarts)

Input: $(\alpha_k)_{k\in\mathbb{N}^\star}$ (sequence of exploration probabilities),
$\delta \in (0,1)$ (maximum error probability for the test);
Input: *Option*: **Local** or **Global** restart;
Initialization: $\forall a \in \{1,\dots,A\}$, $\tau_a \leftarrow 0$ (last restart) and $n_a \leftarrow 0$ (number of selections since last restart)
$k \leftarrow 1$ (number of episodes)
for $t = 1, 2, \dots, T$ **do**
  if $t \mod \left\lfloor \frac{A}{\alpha_k} \right\rfloor \in \{1,\dots,A\}$ **then**
    $A_t \leftarrow t \mod \left\lfloor \frac{A}{\alpha_k} \right\rfloor$
  else
    $A_t \leftarrow \arg\max_{a\in\{1,\dots,A\}} \mathrm{UCB}_a(t)$ as defined in (6)
  end
  Play arm A_t and receive the reward $X_{A_t,t}$: $n_{A_t} \leftarrow n_{A_t}+1$; $Y_{A_t,n_{A_t}} \leftarrow X_{A_t,t}$
  if $\mathrm{GLR}_\delta(Y_{A_t,1},\dots,Y_{A_t,n_{A_t}}) = \mathrm{True}$ **then**
    if Local *restart* **then**
      $\tau_{A_t} \leftarrow t$ and $n_{A_t} \leftarrow 0$ and $k \leftarrow k+1$
    else
      $\forall a \in \{1,\dots,A\}, \tau_a \leftarrow t$ and $n_a \leftarrow 0$ and $k \leftarrow k+1$
    end
end

algorithm based on local restarts, the UCB index are defined differently[3]: $f(t-\tau_a(t))$ is replaced by $f(n_t)$ with $n_t = \sum_{a=1}^{A} n_a(t)$.

The forced exploration scheme used in GLR-klUCB (lines 3-5) generalizes the deterministic exploration scheme proposed for M-UCB by (Cao et al., 2019), whereas CUSUM-UCB performs randomized exploration. A consequence of this forced exploration is given in Proposition 3 (proved in Appendix A).

Proposition 3 *Let s, t be two time instants between two consecutive restarts on arm a (i.e. $\tau_a(t) < s < t$). Then it holds that $n_a(t) - n_a(s) \geq \left\lfloor \frac{\alpha_{k_t}}{A}(t-s) \right\rfloor$, with k_t the number of episodes before round t.*

5. Regret Analysis

In this section, we prove regret bounds for GLR-klUCB using Global Restart. Our results for GLR-klUCB with Local Restart are a bit weaker and are deferred to Appendix D.

5.1 Regret Upper Bounds

Recall that $\tau^{(k)}$ denotes the position of the k-th breakpoint and let $\mu_a^{(k)}$ be the mean of arm a on the segment $\{\tau^{(k-1)}+1,\dots,\tau^{(k)}\}$. We also introduce $k^\star = \arg\max_a \mu_a^{(k)}$, the sub-optimality gap $\Delta_a^{(k)} = \mu_{k^\star}^{(k)} - \mu_a^{(k)}$ and the recall that the magnitude of breakpoint k is $\Delta^{c,(k)} = \max_{a=1,\dots,A} |\mu_a^{(k)} - \mu_a^{(k-1)}| > 0$.

3. This choice is currently not fully supported by theory, as we found mistakes in the analysis of CUSUM-UCB: Hoeffding's inequality is wrongly used with a *random* number of observations and a *random* threshold to obtain Eq. (31)-(32).

We first introduce an assumption, which is easy to interpret and standard in non-stationary bandits. It requires that the distance between two consecutive breakpoints is large enough: how large depends on the magnitude of the largest change that happens at those two breakpoints.

Assumption 4 *Define the delay* $d^{(k)} = d^{(k)}(\alpha, \delta)$ *as*

$$d^{(k)}(\alpha,\delta) = \left\lceil \frac{4A}{\alpha_k \left(\Delta^{c,(k)}\right)^2} \beta\left(\tfrac{3}{2}(\tau^{(k)} - \tau^{(k-1)}), \delta\right) + \frac{A}{\alpha_k} \right\rceil,$$

we assume that $\forall k \leq \Upsilon_T, \tau^{(k)} - \tau^{(k-1)} \geq 2(d^{(k)} \vee d^{(k-1)})$.

Under Assumption 4, we provide in Theorem 5 a finite time problem-dependent regret upper bound. It features the parameters α and δ, the gaps $\Delta_a^{(k)}$ and KL-divergence terms $\mathrm{kl}(\mu_a^{(k)}, \mu_{k^\star}^{(k)})$ expressing the hardness of the stationary MAB problem between two breakpoints, and the detection delays $d^{(k)}(\alpha, \delta)$, which feature the gap $\Delta^{c,(k)}$ and express the hardness of the detection of each breakpoint.

Theorem 5 *For* $\alpha = (\alpha_1, \alpha_2, \dots)$ *an increasing exploration sequence and* δ *for which Assumption 4 is satisfied, the regret of* GLR-klUCB *with parameters* α *and* δ *based on* ***Global*** *Restart satisfies*

$$R_T \leq (A+1)\Upsilon_T \delta T + A\delta T + \alpha_{\Upsilon_T+1} T$$
$$+ \sum_{k=0}^{\Upsilon_T} \sum_{a:\Delta_a^{(k)}>0} \min\left\{ \Delta_a^{(k)}\left(\tau^{(k+1)} - \tau^{(k)}\right); \Delta_a^{(k)}\left[d^{(k)}(\alpha,\delta) + \frac{\ln\left(\tau^{(k+1)} - \tau^{(k)}\right)}{\mathrm{kl}\left(\mu_a^{(k)}, \mu_{k^\star}^{(k)}\right)} \right] + \mathcal{O}\left(\sqrt{\ln\left(\tau^{(k+1)} - \tau^{(k)}\right)}\right) \right\}.$$

We express below the scaling of this regret bound when the exploration sequence α and the parameter δ are carefully tuned using the knowledge of the horizon T, but *without the knowledge of the number of breakpoints* Υ_T. We express this scaling as a function of the smallest value of a sub-optimality gap on one of the stationary segments and the gap of the hardest breakpoint to detect, respectively defined as $\Delta^{\text{opt}} := \min_{k=1,\dots,\Upsilon_T} \min_{\{a:\Delta_a^{(k)}>0\}} \Delta_a^{(k)}$, and $\Delta^{\text{change}} := \min_{k=1,\dots,\Upsilon_T} \Delta^{c,(k)} = \min_{k=1,\dots,\Upsilon_T} \max_{a=1,\dots,A} |\mu_a^{(k)} - \mu_a^{(k-1)}|$.

Corollary 6 *For any* $\alpha_0 \in \mathbb{R}^+$ *and* $\gamma \in (1/2, 1]$, *choosing*

$$\alpha_k = \alpha_0 \sqrt{\frac{kA\ln(T)}{T}} \quad \text{and} \quad \delta = \frac{1}{T^\gamma},$$

on problem instances satisfying the corresponding Assumption 4, the regret of GLR-klUCB *satisfies*

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

and $R_T = \mathcal{O}\left(\frac{\sqrt{\Upsilon_T AT\ln(T)}}{\left(\Delta^{\text{change}}\right)^2} \right)$.

If Δ^{change} is viewed as a constant, our regret upper bound is matching the lower bound of Seznec et al. (2020) up to a $\sqrt{\ln(T)}$ factor. Hence we propose a tuning of GLR-klUCB which attains near-optimal regret without the knowledge of the number of breakpoints, on "easy" problems such that two consecutive breakpoints are separated by more than $\sqrt{TA\ln(T)}/(\Delta^{\text{change}})^2$ time steps. As shown is Section 6, this doesn't prevent GLR-klUCB from performing well on more realistic instances, which was similarly

observed by Cao et al. (2019) for M-UCB. The dependency in $\left(\Delta^{\text{change}}\right)^{-2}$ is also present in the regret bound for other algorithms combining UCB-style algorithms and changepoint detectors Liu et al. (2018); Cao et al. (2019). It may come from a limitation of the current analysis of such algorithms, which require every breakpoint to be detected.

Compared to other algorithms based on stationary bandit strategies combined with change-point detectors, GLR-klUCB is the only one that doesn't require a tuning based on Υ_T to attain the best possible regret. Indeed, it uses an increasing exploration sequence instead of a constant sequence, which allows the trick (8) in the proof of Corollary 6. If Υ_T is known, observe that one can also run GLR-klUCB with the constant exploration sequence $\alpha_k = \alpha = \alpha_0\sqrt{kA\ln(T)/T}$ and obtain the same regret as in Corollary 6. We tried the two alternatives in our experiments, and got similar performances. Hence, the use of an exploration sequence that is agnostic to Υ_T does not hinder the practical performance of GLR-klUCB.

Finally, there exist algorithms which attain near-optimal regret without the knowledge of Υ_T and with no $\left(\Delta^{\text{change}}\right)^{-2}$ multiplicative factor (Auer et al., 2019b; Chen et al., 2019). However, these algorithms are very conservative in order to make their analysis possible. For instance, AdSwitch is an elimination policy, which is often a poor choice in practice for regret minimization. In Section 6, we indeed show that GLR-klUCB greatly outperforms AdSwitch.

5.2 Proof of Corollary 6

With the choice $\delta = T^{-\gamma}$ and $\alpha_k = \alpha_0\sqrt{k\ln(T)A/T}$, Theorem 5 yields the following upper bound on the regret of GLR-klUCB:

$$(A+1)\Upsilon_T T^{1-\gamma} + AT^{-\gamma} + \alpha_0\sqrt{(\Upsilon_T+1)AT\ln(T)} + \sum_{k=1}^{\Upsilon_T}\frac{4A}{\alpha_k\left(\Delta^{c,(k)}\right)^2}\beta(\tfrac{3}{2}T, T^{-\gamma})$$
$$+\sum_{k=1}^{\Upsilon_T}\sum_{a:\mu_a^{(k)}\neq\mu_{k^\star}^{(k)}}\frac{\left(\mu_{k^\star}^{(k)}-\mu_a^{(k)}\right)}{\mathrm{kl}\left(\mu_a^{(k)},\mu_{k^\star}^{(k)}\right)}\ln(T) + \mathcal{O}\Big(\sqrt{\ln(T)}\Big)\ . \tag{7}$$

For $\gamma > 1/2$, the leading term in this expression is

$$\alpha_0\sqrt{(\Upsilon_T+1)AT\ln(T)} + \sum_{k=1}^{\Upsilon_T}\frac{4A}{\alpha_k\left(\Delta^{c,(k)}\right)^2}\beta(\tfrac{3}{2}T, T^{-\gamma}) + \sum_{k=1}^{\Upsilon_T}\sum_{a:\mu_a^{(k)}\neq\mu_{k^\star}^{(k)}}\frac{\left(\mu_{k^\star}^{(k)}-\mu_a^{(k)}\right)}{\mathrm{kl}\left(\mu_a^{(k)},\mu_{k^\star}^{(k)}\right)}\ln(T)\ .$$

Using that $\beta(n,\delta) \leq C\ln(n/\delta)$ for some absolute constant C together with the fact that

$$\sum_{k=1}^{\Upsilon_T}\frac{1}{\alpha_k} = \frac{1}{\alpha_0}\sqrt{\frac{T}{A\ln(T)}}\sum_{k=1}^{\Upsilon_T}\frac{1}{\sqrt{k}} \leq \frac{1}{\alpha_0}\sqrt{\frac{\Upsilon_T T}{A\ln(T)}} \tag{8}$$

yields the following control on the expected regret

$$R_T = \mathcal{O}\left((1+\gamma)\frac{\sqrt{\Upsilon_T AT\ln(T)}}{\left(\min_{k=1}^{\Upsilon_T}\Delta^{c,(k)}\right)^2} + \sum_{k=1}^{\Upsilon_T}\sum_{a:\mu_a^{(k)}\neq\mu_{k^\star}^{(k)}}\frac{\left(\mu_{k^\star}^{(k)}-\mu_a^{(k)}\right)}{\mathrm{kl}\left(\mu_a^{(k)},\mu_{k^\star}^{(k)}\right)}\ln(T)\right)\ .$$

The conclusion follows from Pinsker's inequality: $\mathrm{kl}(\mu_a^{(k)}, \mu_{k^\star}^{(k)}) \geq 2\left(\Delta_a^{(k)}\right)^2$ and from lower bounding all sub-optimality gaps by Δ^{opt}.

Rather than using the problem-dependent complexity of the MAB problem on each stationary segment, using Theorem 5 and standard techniques one can also obtain the following "worse-case" upper bound:

$$(A+1)\Upsilon_T T^{1-\gamma} + AT^{-\gamma} + \alpha_0\sqrt{(\Upsilon_T+1)AT\ln(T)} + \sum_{k=1}^{\Upsilon_T}\frac{4A}{\alpha_k\left(\Delta^{c,(k)}\right)^2}\beta(\tfrac{3}{2}T, T^{-\gamma}) + \sum_{k=1}^{\Upsilon_T}\sqrt{A\left(\tau^{(k+1)}-\tau^{(k)}\right)\ln(T)}\,.$$

Using the Cauchy-Schwarz inequality, the last term in this sum is upper bounded by $\sqrt{\Upsilon_T AT\ln(T)}$. Following the same steps as before, we get a scaling in the regret that no longer depends on Δ^{opt}.

5.3 Proof of Theorem 5

We first introduce some notation for the proof. Recall that $\tau^{(k)}$ denote the k-th breakpoint, we add the convention that $\tau^{(\Upsilon_T+1)} = T$. We denote by $\hat{\tau}^{(k)}$ the k-th breakpoint detected by GLR-klUCB.

Distinguishing the exploration steps and the steps in which GLR-klUCB uses the UCBs to select the next arm to play, one can upper bound the regret as

$$R_T \leq \mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\left(t\left\lceil\frac{A}{\alpha_{k_t}}\right\rceil \in \{1,\dots,A\}\right) + \sum_{t=1}^{T}(\mu_{a_t^\star}(t) - \mu_{A_t}(t))\mathbb{1}\left(\mathrm{UCB}_{A_t}(t-1) \geq \mathrm{UCB}_{a_t^\star}(t-1)\right)\right] \tag{9}$$

We now introduce some high-probability event in which all the breakpoints are detected within a reasonable delay. With $d^{(k)} = d^{(k)}(\alpha,\delta)$ in Assumption 4, we define

$$\mathcal{E}_T = \mathcal{E}_T(\alpha,\delta) := \left(\forall k \in \{1,\dots,\Upsilon_T\}, \hat{\tau}^{(k)} \in \left[\tau^{(k)}+1, \tau^{(k)}+d^{(k)}\right], \hat{\tau}^{(\Upsilon_T+1)} > T\right). \tag{10}$$

Note that from Assumption 4, as the period between two changes are long enough, if $\mathcal{E}_T$ holds, then for all change k, one has $\tau^{(k)} \leq \hat{\tau}^{(k)} \leq \tau^{(k+1)}$ for all $k \in \{1,\dots,\Upsilon_T\}$. Also, when $\mathcal{E}_T$ holds, GLR-klUCB experiences exactly Υ_T restarts which permits to upper bound the exploration term in (9), using the convention that $\hat{\tau}^{(\Upsilon_T+1)} = T$:

$$\begin{aligned}
\sum_{t=1}^{T}\mathbb{1}\left(t\left\lceil\frac{A}{\alpha_{k_t}}\right\rceil \in \{1,\dots,A\}\right) &\leq \sum_{k=0}^{\Upsilon_T}\sum_{t=\hat{\tau}^{(k)}+1}^{\hat{\tau}^{(k+1)}}\mathbb{1}\left(t\left\lceil\frac{A}{\alpha_{k+1}}\right\rceil \in \{1,\dots,A\}\right) \\
&\leq \sum_{k=0}^{\Upsilon_T}\alpha_{k+1}\left(\hat{\tau}^{(k+1)}-\hat{\tau}^{(k)}\right) \leq \alpha_{\Upsilon_T+1}\sum_{k=0}^{\Upsilon_T}\left(\hat{\tau}^{(k+1)}-\hat{\tau}^{(k)}\right) \\
&= \alpha_{\Upsilon_T+1}T.
\end{aligned}$$

On $\mathcal{E}_T$, the second term in (9) can also be decomposed along the Υ_T+1 episodes experienced by the algorithm. Recalling that $k^\star$ denotes the optimal arm for $t \in \left[\tau^{(k)}+1, \tau^{(k+1)}\right]$, one can write

$$R_T \leq T\mathbb{P}\left(\mathcal{E}_T^c\right) + \alpha_{\Upsilon_T+1}T + \sum_{k=0}^{\Upsilon_T}\mathbb{E}\left[\mathbb{1}(\mathcal{E}_T)\sum_{t=\tau^{(k)}+1}^{\tau^{(k+1)}}\left(\mu_{k^\star}^{(k)} - \mu_{A_t}^{(k)}\right)\mathbb{1}(\mathrm{UCB}_{A_t}(t-1) \geq \mathrm{UCB}_{k^\star}(t-1))\right]. \tag{11}$$

The conclusion follows from the two lemmas stated below, whose proofs are given in Appendix C. The first one hinges on some elements of the analysis of the klUCB algorithm proposed by Cappé et al. (2013) whereas the second exploits the changepoint detection mechanism.

Lemma 7 *With $\Delta_a^{(k)} = \mu_{k^\star}^{(k)} - \mu_a^{(k)}$, the following upper bound holds:*

$$(11) \leq \sum_{k=0}^{\Upsilon_T} \sum_{a:\Delta_a^{(k)}>0} \min\left\{ \Delta_a^{(k)} \left(\tau^{(k+1)} - \tau^{(k)}\right) ; d^{(k)} + \frac{\Delta_a^{(k)} \ln\left(\tau^{(k+1)} - \tau^{(k)}\right)}{\mathrm{kl}\left(\mu_a^{(k)}, \mu_{k^\star}^{(k)}\right)} + \mathcal{O}\left(\sqrt{\ln\left(\tau^{(k+1)} - \tau^{(k)}\right)}\right) \right\} .$$

Lemma 8 *Under Assumption 4, it holds that $\mathbb{P}(\mathcal{E}_T^c(\alpha,\delta)) \leq \delta(A+1)\Upsilon_T + A\delta$.*

The tricky part in the analysis is the proof of Lemma 8, which crucially exploits Assumption 4, that we briefly sketch here (with a detailed proof in Appendix C.2). Introducing the event $\mathcal{C}^{(k)} = \left\{\forall \ell \leq k, \hat{\tau}^{(\ell)} \in \left[\tau^{(\ell)}+1, \tau^{(\ell)}+d^{(\ell)}\right]\right\}$ that all the changes up to the k-th have been detected and using the convention $\tau^{(\Upsilon_T+1)} = T$, a union bound permits to upper bound $\mathbb{P}(\mathcal{E}_T^c)$ by the sum of two terms:

$$\sum_{k=1}^{\Upsilon_T+1} \underbrace{\mathbb{P}\left(\hat{\tau}^{(k)} \leq \tau^{(k)} | \mathcal{C}^{(k-1)}\right)}_{(a)} + \sum_{k=1}^{\Upsilon_T} \underbrace{\mathbb{P}\left(\hat{\tau}^{(k)} \geq \tau^{(k)} + d^{(k)} \mid \mathcal{C}^{(k-1)}\right)}_{(b)} .$$

The event in (a) implies that the change point detector associated with some arm a experiences a false alarm. The probability of such an event is upper bounded by Lemma 2 for a changepoint detector run in isolation. Under the bandit algorithm, arm a's change point detector is based on less than $t - \tau_a(t)$ samples, which makes a false alarm even less likely. We finally show that $(a) \leq A\delta$ (with union bound over the A arms).

Term (b) is related to the control of the detection delay, which is more involved under the GLR-klUCB adaptive sampling scheme, when compared to a result like Theorem 6 in Maillard (2019) for the changepoint detector run in isolation. More precisely, we need to leverage the forced exploration (Proposition 3) to be sure we have enough samples for detection. This explains why the detection delay for the k-th breakpoint defined in Assumption 4 is scaled by α_k. Using some elementary calculus and a concentration inequality given in Lemma 10, we can finally prove that $(b) \leq \delta$.

6. Experimental Results

In this section, we report numerical simulations performed on synthetic data to compare the performance of GLR-klUCB against other state-of-the-art approaches. Experiments were performed with a library written in the Julia language which is available online.[4]

Algorithms and Parameters Tuning We include two baselines: the klUCB algorithm (not designed for the non-stationary setting) and an algorithm that we call Oracle-klUCB, which knows the exact locations of the breakpoints, and restarts klUCB for all arms at those locations. Then, we include algorithms with state-of-the-art regret for the piecewise stationary MAB presented in Section 2. For a fair comparison, all algorithms that use UCB as a sub-routine were adapted to use klUCB instead, which yields better performance[5]. For all the algorithms, we used the tuning recommended in the corresponding paper, using in particular the knowledge of the number of breakpoints Υ_T and the horizon T when needed. Only two algorithms do not require the knowledge of Υ_T: AdSwitch and GLR-klUCB.

We experiment with Exp3.S (with theoretically optimal tuning in Corollary 8.3 of Auer et al. (2002b)), and the two passively adaptive algorithms Discounted klUCB (D-klUCB) with discount factor $\gamma =$

4. https://github.com/EmilieKaufmann/PiecewiseStationaryBandits
5. Liu et al. (2018); Cao et al. (2019) both mention that extending their analysis to the use of klUCB should not be too difficult.

$1-\sqrt{\Upsilon_T/T}/4$ and Sliding-Window klUCB (SW-klUCB) with window-size $\tau = \lceil 2\sqrt{T\ln(T)/\Upsilon_T}\rceil$. As for *actively adaptive algorithms*, we experiment with AdSwitch (Auer et al., 2019b) and three algorithms combining a change-point detector with klUCB: CUSUM-klUCB, M-klUCB and GLR-klUCB. These three algorithms share the use of an exploration parameter that we call α (or an exploration sequence α_k for GLR-klUCB). Liu et al. (2018) and Cao et al. (2019) recommend two slightly different tuning for CUSUM-klUCB and M-klUCB respectively, that both scale in $\sqrt{\Upsilon_T \ln(T)/T}$. This is also the order of magnitude of α_{Υ_T} given by Corollary 6 for GLR-klUCB. Hence, in order to compare algorithm that adds a similar amount of exploration, we set $\alpha = \sqrt{\Upsilon_T A \ln(T)/T}$ for all algorithms using a constant exploration probability and $\alpha_k = \sqrt{kA\ln(T)/T}$ for the exploration sequence of GLR-klUCB.

Regarding the parameters of the change-point detectors, we use a threshold $h = \ln(T/\Upsilon_T)$ for CUSUM-klUCB, as recommended by Liu et al. (2018), and experience with different values of (M, ϵ) that have to be tuned using some prior knowledge of the problem. For M-klUCB, we experience with different values of the windows parameter w (often choosing the tuning $w = 800$ that was found to be robust in the experiments of Cao et al. (2019)) and use the recommended threshold $b = \sqrt{w\ln(2AT^2)/2}$. For the Bernoulli-GLR test, we use the threshold function $\beta(n, \delta) = \ln(n^{3/2}/\delta)$ and set $\delta = 1/\sqrt{T}$, which is the largest value licensed by Corollary 6.

For GLR-klUCB and AdSwitch, which are computationally more demanding due to the use of tests based on scan-statistics, we use some implementation tweaks. For GLR-klUCB, we use some down-sampling as discussed in Section 3.3, performing the test only every $\Delta t = 10$ time steps and scanning every $\Delta s = 5$ observations for a possible change-point. To be able to implement AdSwitch up to a horizon $T = 5000$, we used $\Delta t = 50$. The computational bottleneck in AdSwitch is the checks on good arms that compare the empirical mean between s and t to that between s_1 and s_2 for all possible $s < t$ and $s_1 \leq s_2 < t$. We only test values of s_1, s_2 and s satisfying $s' \mod \Delta s = 0$, for $\Delta s = 20$. This reduces the time complexity by $(\Delta s)^3$, which is a significant speed-up in practice. Finally, the parameter C_1 that governs the elimination of good arms and should be chosen large enough was set to $C_1 = 1$.

Results on two simple benchmarks We design two simple piecewise stationary bandit problems with $A = 3$ arms and $\Upsilon_T = 4$ breakpoints. These breakpoints are evenly spaced up to the horizon, for which we investigate 4 values for each problem: $T = 5000$, $T = 10000$, $T = 20000$ and $T = 100000$. In **Problem 1**, a single arm changes in each breakpoint ($\Upsilon_T = C_T = 4$) and $\Delta^{\text{change}} = 0.3$, whereas in **Problem 2**, all arms means change at every breakpoint ($\Upsilon_T = 4, C_T = 16$) and $\Delta^{\text{change}} = 0.2$. For each problem, we display the reward functions of each arm in the top left corner of Figures 1 and 2.

For the different values of the horizon, the reward functions are simply expanded: the size Δ^{change} remains the same and the breakpoints are still evenly spaced. Hence, it is a way to vary the difficulty of the underlying change-point detection problems. Indeed, when T goes larger, the distance between two consecutive breakpoints increases, and Assumption 4 is closer to be satisfied. On this simple problems with equally spaced breakpoints ($\tau^{(k)} - \tau^{(k-1)} = T/5$), with our choice of α_k and δ, Assumption 4 amounts to

$$\sqrt{T} \geq \frac{80\sqrt{A}}{\min_{k=1}^{\Upsilon_T} \sqrt{k}(\Delta^{c,(k)})^2}\sqrt{\ln(T)} + 1$$

which is only satisfied for T much larger than 100000 for both Problem 1 and Problem 2. Therefore, in this experiment, we investigate the performance of GLR-klUCB for difficult problems on which it does not have theoretical guarantees.

In Figure 1 (respectively Figure 2), we display the results for Problem 1 (respectively Problem 2). We display the regret of each algorithm as a function of the rounds for one horizon (top right corner); and we also tabulate the regret at the horizon and the number of restarts for all the algorithms and all the horizons.

In Problem 1, we observe that the regret of GLR-klUCB with Global and Local restart is competitive with that of the SW-klUCB which performs best for the different time horizons T. However, recall that this algorithm is tuned using the knowledge of Υ_T, unlike GLR-klUCB. In Problem 2, the regret of GLR-klUCB is the smallest for large horizons ($T = 20000, 100000$) whereas for shorter horizons ($T = 5000, 10000$) klUCB and SW-klUCB have (slightly) smaller regret. Regarding other passively adaptive approaches, we see that D-klUCB is competitive with (sometimes even better than) actively adaptive algorithms, whereas Exp3.S only manages to outperform klUCB for large horizons. GLR-klUCB largely outperforms AdSwitch for $T = 5000$, which is the largest horizon for which we could implement this algorithm. We now turn our attention to CPD-based algorithms.

The tests used by CUSUM-UCB and M-UCB depend on two sets of parameters that should in principle be chosen according to some prior knowledge of the problem, and we tried for each algorithm two different tunings of these parameters. For CUSUM-UCB, the two sets of parameters yield similar regret on Problem 2, but one is much better than the other on Problem 1. For M-UCB, the two sets of parameters yield similar regret on Problem 1, but one is much better than the other on Problem 2. This sheds light on the fact that tuning these parameters may be difficult. On the contrary, the tuning of the Bernoulli GLR test used in GLR-klUCB only requires to specify the error probability δ, and setting it to $\delta = 1/\sqrt{T}$ as suggested by Corollary 6 yield good performance on both Problem 1 and Problem 2.

To understand the behavior of the CPD-based algorithms, we analyze their average number of restarts, reported in the tables in Figure 1 and 2. In an asymptotic regime (i.e. for T such that Assumption 4 or Assumption 11 is satisfied), GLR-klUCB should detect all breakpoints with Global restart and all change-points with Local restart. As can be seen, the asymptotic regime is not met in our experiments, except for $T \geq 20000$ on Problem 2 in which GLR-klUCB with Global restart performs exactly $\Upsilon_T = 4$ restarts. Besides this case, GLR-klUCB typically detects fewer changes than expected, for example between 2 and 3 on Problem 1. Note that M-UCB tends to detect fewer changes than GLR-klUCB, whereas CUSUM-UCB tend to detect more. Especially, when the parameter ε (giving the minimal amount of change the CUSUM test should detect) is $\varepsilon = 0.05$, we observe that CUSUM-UCB experiences false-alarms, especially for large horizons (yet this does not prevent the algorithm from having a regret smaller than that of klUCB). Overall, we remark that GLR-klUCB is among the best algorithms on both problems for all the horizon values, including the smallest ones: it shows that GLR-klUCB is competitive in practice even when the Assumptions 4 and 11 are violated.

In these experiments, we tried four variants of GLR-klUCB: we investigate the use Global and Local restarts and the use of two exploration sequences: a constant exploration probability $\alpha = \sqrt{\Upsilon_T A \ln(T)/T}$ and the exploration sequence $\alpha_k = \sqrt{kA \ln(T)/T}$ that does not require to know the number of breakpoints. We observe that the two types of restarts yield comparable performance (with a slight advantage for Global restarts), and thus investigate the two variants further on a wider benchmark. As for the exploration sequences, we observe that the time-varying one (agnostic to Υ_T) always performs best. The reason is that it performs less forced exploration in the first episodes, and as we shall see in our next experiments, scaling down the exploration probability (or exploration sequence) for CPD-based algorithms can lead to better empirical performance. Still, GLR-klUCB with a constant exploration probability α also outperforms most of the time other CPD-based algorithms using the exact same α.

Robustness on more diverse benchmarks We now investigate further the performance of the best algorithms for Problem 1 and Problem 2 on a large number of randomly generated piecewise stationary bandit models, with $T = 20000$. To generate a random instance, we specify the number of arms A, the maximal number of breakpoints Υ, a change-point probability p, a minimal distance $d_{\min}$, a minimal and maximal amount of change, $\Delta_{\min}$ and $\Delta_{\max}$. Then, we sample the breakpoints uniformly at random