Exploiting History Data for Nonstationary Multi-armed Bandit

Gerlando Re, Fabio Chiusano, Francesco Trovò (✉), Diego Carrera, Giacomo Boracchi, and Marcello Restelli

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Piazza Leonardo da Vinci, 32, Milano, Italy
{diego.carrera, francesco1.trovo, giacomo.boracchi, marcello.restelli}@polimi.it
{fabio.chiusano, gerlando.re}@mail.polimi.it

Abstract. The Multi-armed Bandit (MAB) framework has been applied successfully in many application fields. In the last years, the use of active approaches to tackle the nonstationary MAB setting, i.e., algorithms capable of detecting changes in the environment and re-configuring automatically to the change, has been widening the areas of application of MAB techniques. However, such approaches have the drawback of not reusing information in those settings where the same environment conditions recur over time. This paper presents a framework to integrate past information in the abruptly changing nonstationary setting, which allows the active MAB approaches to recover from changes quickly. The proposed framework is based on well-known *break-point prediction* methods to correctly identify the instant the environment changed in the past, and on the definition of *recurring concepts* specifically for the MAB setting to reuse information from recurring MAB states, when necessary. We show that this framework does not change the order of the regret suffered by the active approaches commonly used in the bandit field. Finally, we provide an extensive experimental analysis on both synthetic and real-world data, showing the improvement provided by our framework.

Keywords: Multi-Armed Bandit · Non-stationary MAB · Break-point Prediction · Recurring Concepts

1 Introduction

The stochastic Multi-Armed Bandit (MAB) setting has been widely used in real-world applications in sequential decision-making problems, e.g., for clinical trials [4], network routing [17], dynamic pricing [21], and internet advertising [16]. In the stochastic MAB framework, a learner selects an option – commonly referred to as *arm* – among a given finite set and observes a corresponding stochastic reward. The learning goal is to maximize the rewards collected during the entire learning process. The success of this framework is mainly due to its strong theoretical properties [7], which, in practice, turns into very effective results.

Over the past few years, researchers have targeted new strategies to increase the flexibility of the MAB framework, thus foresee new applications to more complex scenarios. One of the most interesting extensions of MAB techniques consists of handling scenarios where the distribution of rewards varies over time. This is a relatively common situation in real-world dynamic pricing [21] and online advertising problems [13], where the distributions of reward for each arm can be considered stationary only over short time intervals as they might evolve due to changes of the competitors' strategies or abrupt modification of the user behaviour. While the most general situation where reward distributions are allowed to arbitrarily change over time is not tractable by this framework, it is possible to design efficient and theoretically grounded learning algorithms under some mild assumption on change type and regularity.

One of the most studied settings, which commonly occurs in practical applications, is that of the so called *abruptly changing MAB* environments, where each arm reward expected value is a piece-wise constant function of time and is allowed to change a finite number of times. MAB algorithms operating in this setting follow two mainstream approaches to cope with nonstationarity: passive [9, 22], and active [14, 8]. Passive methods use only the most recent rewards to define the next arm to be selected. Thus, they progressively discard rewards gathered in the far past as soon as new samples are collected. Conversely, active MAB algorithms incorporate detection procedures to spot the change and adapt the decision policy only when necessary. This approaches, from now on addressed as Change Detection MABs (CD-MABs), couple a stationary MAB procedure with a Change Detection Test (CDT) [5], as for instance in [14]. Even if from a theoretical point of view the two approaches have similar guarantees, it has been shown that the active approaches are performing generally better when their empirical performances are tested [14].

In the CD-MAB framework, a CDT is used to monitor the distribution of rewards, and as soon as this gathers enough empirical evidence to state that a change has occurred, it triggers a detection and restarts from scratch the classical MAB procedure. In practice, a change detected on a specific arm triggers a reset of both the statistics of the CDT and the corresponding arm. The major limitation of this approach is that it discards the information gathered in the past by MAB, while this could be potentially used in two situations. On the one hand, samples gathered between the occurrence and the detection of the change can be used to reconfigure the MAB over the specific arm and avoid a complete restart from scratch. On the other hand, when the process presents some regularity over time, e.g., seasonal effects, it would be ideal to identify when the arm goes back in a state that was already encountered and use the information learned about that distribution to have a fast recover after the detection.

In this paper, we present the Break-point and Recurrent MAB (BR-MAB), which extends generic CD-MABs to reuse data collected before the detection and replaces the MAB cold restart with a better initialization. Most remarkably, our neat approach still makes theoretical analysis amenable in these non-stationary settings. In particular, our novel contributions are:

- we propose a technique based on *break-point prediction* [11], to reuse the most informative samples for the current distribution gathered before the change has been detected;
- we propose a technique to identify the so-called *recurrent phases* in the MAB setting, to handle cases in which seasonality effect are present;
- we integrate these techniques in a single framework, called BR-MAB, which allow their application to a generic CD-MAB;
- we show that, BR-MAB applied to CUSUM-UCB maintains the theoretical guarantees of the original active non-stationary MAB;
- we provide extensive empirical analysis to show the improvement provided by BR-MAB, when applied to a CD-MAB, comparing its performance with the state-of-the-art techniques for non-stationary MAB settings.

2 Related Works

The algorithms designed to tackle non-stationary MAB problems with a limited number of changes are divided into passive and active approaches.

From the passive approaches, we mention the D-UCB algorithm [9], which deals with nonstationarity by giving less importance to rewards collected in the near past by weighting them by a discount factor. Conversely, the SW-UCB algorithm [9] fixes a window size and feeds a UCB-like algorithm only with the most recently collected samples. They provide guarantees on the upper-bound for the pseudo-regret of order $O(\sqrt{NB_N}\log N)$ and $O(\sqrt{NB_N\log N})$, respectively, where N is the time horizon of the learning process, and B_N is the number of changes present in the environment up to time N. Another well-analyzed passive method is the SW-TS [22], which applies the sliding window approach to the Bayesian Thompson Sampling algorithm. It provides a bound on the pseudo-regret of $O(\sqrt{N}\log N)$, if the number of changes is constant w.r.t. N. We want to remark that, in general, the passive approach does not allow for incorporating information coming from past data since their intrinsic strategy consists of systematically discarding them. Therefore, they are not appealing candidates for the approach proposed here.

For what concerns the active approaches, i.e., those algorithms using a CDT to actively detect changes in the expected values of the arms' reward distributions, the bandit literature offers a wide range of techniques [14, 8, 6, 15]. More specifically, the CUSUM-UCB method [14] uses the CUSUM CDT to detect changes and a UCB-like approach as MAB strategy. This method provides theoretical upper bound for its regret of order $O(\sqrt{NB_N\log(N/B_N)})$. The Monitored-UCB [8] is a UCB-like policy with random exploration which uses a windowed CDT to provide a regret bound of $O(\sqrt{NB_N\log(N)})$. The GLR-klUCB [6] uses a KL-UCB algorithm in combination with a Generalized Likelihood Ratio (GLR) test as a change detection algorithm to get a regret of $O(\sqrt{NB_N\log(N)})$. Notably, the approach we propose here can be applied to any of the aforementioned active approach.

Finally, other well known and efficient methods are Adapt-EvE [10], an actively adaptive policy that uses UCB1-Tuned as a sub-algorithm and employs the

Page-Hinkley test [12] to detect decreases in the mean of the optimal arm. Whenever a change-point is detected, a meta-bandit transient phase starts, whose goal is to choose between two options: reset the sub-algorithm or not. Instead, the BOCD-TS [15] uses Thompson Sampling with a Bayesian Change Point Detection algorithm. The upper-bound for these methods is unknown, hence they are accounted as heuristic algorithms.

Garivier et al. [9] showed that the problem of abruptly changing MAB has a lower bound for the the expected pseudo-regret of order $\Omega(\sqrt{N})$. We recall that, in settings in which the optimal arm expected value can change without any restriction, only trivial upper bounds for the dynamic pseudo-regret $\overline{R}_N(\mathfrak{U})$ are known [2]. Conversely, if stricter assumptions holds, e.g., the occurrence of global changes, better guarantees can be derived.

3 Problem Formulation

We model our problem as a stochastic abruptly changing MAB setting, similar to what has been defined in [9], in which the arms reward distributions are constant during sequences of rounds, and they change at specific rounds unknown to the learner. Formally, at each round n over a finite time horizon N, the learner selects an arm $a_{i(n)}$ among a finite set of K arms $\mathcal{A} := \{a_1, \dots, a_K\}$ and observes a realization of the reward $x_{i(n),n}$ from the chosen arm $a_{i(n)}$. The rewards for each arm a_i are modeled by a sequence of independent random variables $X_{i,n}$ from a distribution whose parameters are unknown to the learner. As customary in the MAB literature, here we consider Bernoulli distributed rewards, i.e., $X_{i,n} \sim Be(\mu_{i,n})$, where $\mu_{i,n}$ is the expected value of the reward for arm a_i at round n.[1] During the learning process, we denote as *breakpoints* those rounds in which the expected reward of at least one arm a_i changes. Formally, a break-point $b \in \{1, \dots, N\}$ is a round in which for at least an arm a_i we have $\mathbb{E}[X_{i,b-1}] \neq \mathbb{E}[X_{i,b}]$. In the analysed setting, we have a set of B_N breakpoints $\mathcal{B} := \{b_1, \dots, b_{B_N}\}$ that occur before round N (for sake of notation we define $b_0 = 1$), and whose location is unknown to the learner. The breakpoints determine a set of phases $\{\mathcal{F}_1, \dots, \mathcal{F}_{B_N}\}$, where each phase $\mathcal{F}_\phi$ is a sequence of rounds between two consecutive breakpoints:

$$\mathcal{F}_\phi = \{n \in \{1, \dots, N\} \mid b_{\phi-1} \leq n < b_\phi\}. \tag{1}$$

With abuse of notation, we denote with $\mu_{i,\phi} := \mathbb{E}[X_{i,n}]$, with $n \in \mathcal{F}_\phi$, the expected value of the reward of the arm a_i during the phase $\mathcal{F}_\phi$. Figure 1 illustrates an example of a specific setting with two arms a_1 and a_2 in which three phases $\mathcal{F}_1$, $\mathcal{F}_2$, and $\mathcal{F}_3$ occurs over the time horizon. Note that, differently from the classical MAB setting, a single optimal arm over the entire time horizon might not exist. Indeed, during each phase $\mathcal{F}_\phi$ we define $a^*_\phi := \arg\max_i \mu_{i,\phi}$ the arm having the largest expected reward $\mu^*_\phi := \max_i \mu_{i,\phi}$. A *policy* $\mathfrak{U}$ is a function

[1] The extension to other finite support distributions is straightforward and the theoretical results here provided are still valid.

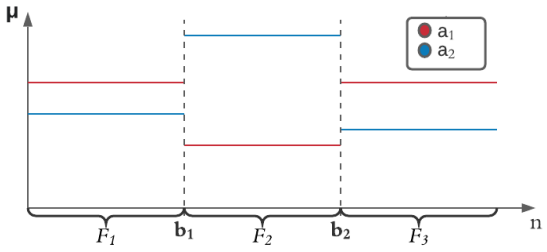


Fig. 1: Example of a nonstationary setting.

$\mathfrak{U}(h) = a_{i(n)}$ that chooses the arm $a_{i(n)}$ to play at round n according to history h, defined as the sequence of past plays and obtained rewards.

Our goal is to design a policy $\mathfrak{U}$ that minimizes the loss w.r.t. the optimal decision in terms of reward. This loss, namely the *dynamic pseudo-regret*, is:

$$\overline{R}_N(\mathfrak{U}) := \mathbb{E}\left[\sum_{n=1}^{N} \mu_n^* - \mu_{i(n),n}\right], \tag{2}$$

where $\mu_n^* := \max_{i\in\{1,\dots K\}} \mu_{i,n}$ is the optimal expected reward at round n.

In this work, we are interested in reusing the information coming from the situation in which an arm a_i has a value of the expected reward that recurs over the different phases. This models the possibility that an arm behaviour is recurring over time due to seasonality effects. Formally:

Definition 1. *A recurrent phase on arm a_i occurs when there exist two phases $\mathcal{F}_\phi, \mathcal{F}_{\phi'}$, with $\phi \neq \phi'$, s.t. $\mu_{i,\phi} = \mu_{i,\phi'}$, i.e., when the arm over the two phases has the same expected reward.*[2]

The rationale behind the above definition is that the information gathered from an arm are valid in the future, no matter how the other arms' rewards are changing, and, thus, they can be reused as long as the arm has the same reward distribution. In Figure 1, two recurrent phases are present, i.e., as $\mathcal{F}_1$ and $\mathcal{F}_3$, since the arm a_1 has $\mu_{1,1} = \mu_{1,3}$. Notice that, if a concept recurs during phase $\mathcal{F}_2$, one might reuse the samples collected during phase $\mathcal{F}_1$ to speed up learning.

Finally, it is common in the CD-MAB literature to require two assumptions [8, 14]. At first, we require a minimum magnitude for the change s.t. it is possible to detect it:

Assumption 1 *$\exists\ \varepsilon \in (0, 1]$, known to the learner, such that for each arm a_i whose expected reward changes between consecutive phases ϕ and $\phi+1$, we have:*

$$|\mu_{i,\phi} - \mu_{i,\phi+1}| \geq \varepsilon. \tag{3}$$

[2] Since we are considering Bernoulli reward, having the same expected value also implies to have the same distribution. This definition can be easily generalized to handle other distributions, requiring that the distribution repeats over different phases.

The second assumption prevents two consecutive breakpoints from being too close in terms of rounds:

Assumption 2 *There exist a number M, known to the learner, such that:*

$$\min_{\phi \in \{1,\dots,B_N\}} (b_\phi - b_{\phi-1}) \geq KM. \tag{4}$$

With reference to Figure 1, the two assumptions are stating that the two breakpoints b_1, and b_2 must be such that $(b_2 - b_1) > KM$, and that $|\mu_{0,\phi} - \mu_{0,\phi+1}| > \varepsilon$, and $|\mu_{1,\phi} - \mu_{1,\phi}| > \varepsilon$ for each $\phi \in \{1, 2\}$. These two assumptions are natural in MAB algorithms adopting CDT as tools to detect changes, e.g., [14, 8, 15] since they state that the changes are detectable by the CDT in a limited amount of rounds (Assumption 1) and allow to set the CDT at the beginning of the learning process and after each change is detected (Assumption 2). Therefore, the knowledge of ϵ and M is customary when designing algorithms following the active framework and allows them to outperform passive ones in terms of empirical performance significantly.

4 The BR-MAB Algorithm

In what follows, we present the BR-MAB algorithm, which can be seen as a generalization of the CD-MAB framework presented in [14] that learns from historical information after each detected change. The BR-MAB algorithm builds upon the definition of a concept $\mathcal{C}_i$ as follows:

Definition 2. *A* concept $\mathcal{C}_i = \{x_1, \dots, x_C\}$ *is a set of rewards collected over time for the arm a_i, which are deemed to belong to the same phase.*

This definition is used in BR-MAB to store information about past phases and identify recurrent phases. In this case, we refer to *recurrent concepts*.

The pseudo-code of the BR-MAB algorithm is presented in Algorithm 1, and takes as input any nonstationary active CD-MAB policy (namely both a change-detection test to be used on each arm and an arm-selection policy), a break-point prediction procedure $\mathfrak{B}$, and a test $\mathcal{E}$ to evaluate when two concepts can be conveniently aggregated. At first, the algorithm initializes all the parameters for the selected CD-MAB and, for each arm a_i, the set of tracked recurrent concepts $\mathfrak{C}_i$, the actual concept being observed $\mathcal{C}_i^{now}$, and a binary variable cf_i to check if a concept had been used in the past for that arm (Line 1). Then, at each round $n \in \{1, \dots, N\}$, the algorithm selects an arm $a_{i(n)}$ accordingly to the CD-MAB policy (Line 3), uses the reward to update the CD-MAB (Line 4), and updates the concept currently in use $C_{i(n)}^{now}$ for the selected arm $a_{i(n)}$, i.e., adds the currently collected reward $x_{i(n),n}$ to the set $C_{i(n)}^{now}$ (Line 5). Subsequently, the CDT of the CD-MAB is being executed and when this detects a change in the currently selected arm $a_{i(n)}$, the break-point procedure B, detailed in Section 4.1, is activated to estimate the break-point r (Line 7). As a result, the rewards collected during rounds $\{r, \dots, n\}$ corresponding to the arm $a_{i(n)}$ are used to

update the information of the arm $a_{i(n)}$ in the CD-MAB (Line 8). Moreover, the algorithm removes the rewards selected by the break-point procedure from the current concept $\mathcal{C}^{now}_{i(n)}$, adds the current concept $\mathcal{C}^{now}_{i(n)}$ to the set of available concepts $\mathfrak{C}_i$ (Line 10), and resets it using the reward of arm $a_{i(n)}$ collected after the break-point (Line 11). Finally, BR-MAB sets $cf_i = 0$, to state that the arm a_i is eligible of using one of the concept in $\mathfrak{C}_{i(n)}$ if it is recurring (Line 12). After the change detection phase occurred, the algorithm tries to detect if a concept in $\mathfrak{C}_{i(n)}$ is recurrent. More specifically, if no concept has been already used for the arm $a_{i(n)}$ ($cf_{i(n)} = 0$), for each concept $\mathcal{C}$ present in $\mathfrak{C}_{i(n)}$, it checks if it can be considered equivalent to the current concept $\mathcal{C}^{now}_{i(n)}$ using the test $\mathcal{E}$ (Line 16), detailed in Section 4.2. If the test $\mathcal{E}$ passes, the current concept $\mathcal{C}^{now}_{i(n)}$ is updated with the rewards contained into the concept $\mathcal{C}$ (Line 18), and $\mathcal{C}$ is removed from $\mathfrak{C}_{i(n)}$ (Line 10). Finally, the CD-MAB procedure is updated using the reward present in the recurrent concept $\mathcal{C}$.

4.1 Break-point Prediction Procedure

In this section, we present the break-point prediction procedure $\mathcal{B}$ that identifies the position of the break-point after the CDT provides a detection. This problem is commonly addressed in the statistical literature by the change-point formulation [11]. These tests perform a retrospective and offline analysis over a sequence of observations that presumably contains a change and determine whether there is enough statistical evidence to confirm the sequence contains a change and case its location. Change-point formulation has also been extended to detect changes in streaming data from a Bernoulli [19] or arbitrary [18] distributions. In this case, the change-point formulation provides change-detection capabilities, and the break-point estimate is automatically provided after each detection.

The CUSUM test [5] is a popular option for the CDT used for monitoring the stream of rewards in the CD-MAB is the CUSUM test. In this case, the test already provide after each detection a break-point estimate. Let t' be the time when a change has been detected on the arm a_i (or possibly $t' = 0$), and let $\{x_{i,t(1)}, \dots x_{i,t(M)}\}$ be the sequence of last M rewards collected from arm a_i from the current phase at rounds $\{t(1), \dots, t(M)\}$. The CUSUM test uses such rewards to estimate the expected values of the reward of a_i, namely $\hat{m}_i := \sum_{h=1}^{M} \frac{x_{i,t(h)}}{M}$. When monitoring the next rounds $h \in \{t' + M + 1, \dots\}$, the CUSUM test computes the following statistics to detect an increase/decrease in the expected reward $\mu_{i,\phi}$:

$$g^+_{i,h} = \begin{cases} \max\{0, g^+_{i,h-1} + x_{i,h} - \hat{m}_i - \varepsilon\} & \text{if } i(h) = i \\ g^+_{i,h-1} & \text{otherwise} \end{cases}, \tag{5}$$

$$g^-_{i,h} = \begin{cases} \max\{0, g^-_{i,h-1} + \hat{m}_i - x_{i,h} - \varepsilon\} & \text{if } i(h) = i \\ g^+_{i,h-1} & \text{otherwise} \end{cases}, \tag{6}$$

where the quantities has been initialized as $g^+_{i,t'+M} = 0$ and $g^-_{i,t'+M} = 0$, and ε is defined in Assumption 1. Changes are detected as soon as one of these statistics

Algorithm 1 BR-MAB

Require: non-stationary algorithm CD-MAB, break-point prediction procedure $\mathcal{B}$, recurrent concept equivalence test $\mathcal{E}$

$\mathfrak{C}_i \leftarrow \emptyset,\ \mathcal{C}_i^{now} \leftarrow \emptyset,\ cf_i \leftarrow 0\ \forall i \in \{1, \dots, K\}$
for $n \in \{1, \dots, N\}$ **do**
  Play $a_{i(n)}$ according to CD-MAB
  Collect reward $x_{i(n),n}$ and update the CD-MAB accordingly
  Update the concept $\mathcal{C}_{i(n)}^{now} \leftarrow \mathcal{C}_{i(n)}^{now} \cup \{x_{i(n),n}\}$
  if a change has been detected by the CD-MAB **then** ▷ change detection
    Run $\mathcal{B}$ to identify the change round r ▷ break-point prediction
    Update arm $a_{i(n)}$ in the CD-MAB using rewards from rounds $\{r, \dots, n\}$
    Remove rewards collected from $a_{i(n)}$ from rounds $\{r, \dots, n\}$ from $\mathcal{C}_{i(n)}^{now}$
    $\mathfrak{C}_{i(n)} \leftarrow \mathfrak{C}_{i(n)} \cup \{\mathcal{C}_{i(n)}^{now}\}$
    Initialize $\mathcal{C}_{i(n)}^{now}$ with the rewards of arm $a_{i(n)}$ collected at rounds $\{r, \dots, n\}$
    $cf_{i(n)} \leftarrow 0$
  end if
  if $cf_{i(n)} = 0$ **then**
    for $\mathcal{C} \in \mathfrak{C}_{i(n)}$ **do**
      if $\mathcal{E}(\mathcal{C}, \mathcal{C}_{i(n)}^{now})$ **then** ▷ recurrent concept test
        $cf_{i(n)} \leftarrow 1$
        $\mathcal{C}_{i(n)}^{now} \leftarrow \mathcal{C} \cup \mathcal{C}_{i(n)}^{now}$ ▷ concept merge
        $\mathfrak{C}_{i(n)} \leftarrow \mathfrak{C}_{i(n)} \setminus \mathcal{C}$
        Update arm $a_{i(n)}$ in the CD-MAB using the rewards in $\mathcal{C}$
      end if
    end for
  end if
end for

exceed a suitable threshold. Let us assume that this occurs at time t'', the round corresponding to the break-point is then identified as:

$$r = \arg \min_{h \in \{t', \dots, t''\}} g_{i,h}^{+}, \quad \text{or} \quad r = \arg \min_{h \in \{t', \dots, t''\}} g_{i,h}^{-}, \tag{7}$$

depending on whether the detection comes from monitoring $g_{i,h}^{+}$ or $g_{i,h}^{-}$, respectively. If there are multiple values attaining the minimum in Equation (7), we set r as the most recent value. Once the break-point prediction occurred, we initialize the CUSUM as described above and reset the two statistics $g_{i,h}^{+}$ and $g_{i,h}^{-}$ before restarting monitoring.

4.2 Recurrent Concepts Equivalence Test

After a change has been detected, we need to assess whether the currently expected reward of an arm a_i, represented in the concept $\mathcal{C}_i^{now}$, corresponds to any of the previously encountered phases using the concepts stored in $\mathfrak{C}_i$. Inspired by [1], we solve this problem by an equivalence test $\mathcal{E}(\cdot, \cdot)$ that consists in a Two One Sided Test (TOST) [20]. More specifically, let $\mathcal{C}_i^{now}$ be the current concept

associated to the arm a_i, and let $\mathcal{C}$ be any concept from the collection of previously seen concepts $\mathcal{C} \in \mathfrak{C}_i$. The TOST determines whether there is enough statistical evidence to claim that the expected rewards in the two concepts $\mathcal{C}_i^{now}$ and $\mathcal{C}$ differ less than a given threshold.

The TOST formulates the following statistical tests over the expected values μ' and μ'' of the rewards in $\mathcal{C}_i^{now}$ and $\mathcal{C}$, respectively:

$$\text{Test 1} \quad H_0 : \mu' - \mu'' \leq -d \qquad \text{vs.} \qquad H_1 : \mu' - \mu'' > -d, \tag{8}$$

$$\text{Test 2} \quad H_0 : \mu' - \mu'' \geq d \qquad \text{vs.} \qquad H_1 : \mu' - \mu'' < d, \tag{9}$$

where $d > 0$ is the equivalence bound, indicating a difference between rewards that is deemed as negligible when identifying recurrent phases. When the TOST rejects both the null hypothesis, we argue that there is enough statistical evidence that the difference $|\mu' - \mu''|$ lies within $(-d, d)$ Therefore, the test $\mathcal{E}(\mathcal{C}_i^{now}, \mathcal{C})$ asserts that the two concept are recurrent, and they are merged into a single concept in the BR-MAB algorithm.

In particular, it uses two two-sample z-test to compare proportions, formally it requires to compute the following test statistics:

$$z_{-d} = \frac{(\hat{\mu}' - \hat{\mu}'') + d}{\sqrt{\frac{\hat{\mu}'(1-\hat{\mu}')}{n'} + \frac{\hat{\mu}''(1-\hat{\mu}'')}{n''}}}, \text{ and } z_d = \frac{(\hat{\mu}' - \hat{\mu}'') - d}{\sqrt{\frac{\hat{\mu}'(1-\hat{\mu}')}{n'} + \frac{\hat{\mu}''(1-\hat{\mu}'')}{n''}}}, \tag{10}$$

where $\hat{\mu}'$ and $\hat{\mu}''$ are the empirical means of the reward stored in the concepts $\mathcal{C}_i^{now}$ and $\mathcal{C}$, respectively, and $n' := |\mathcal{C}_i^{now}|$ and $n'' := |\mathcal{C}|$ are their cardinality. In this test, we fix a significance level α_z, and we reject both null hypothesis when the test statistic z_{-d} is above the $1 - \alpha_z$ quantiles of a normal distribution and z_d is below the α_z quantiles of a normal distribution.

Even though representing in each concept $\mathcal{C}$ the set of rewards is not very efficient in terms of memory requirements, in our case, a much more compact representation is possible. In fact, in the case of Bernoulli rewards, the TOST requires only the mean of the rewards collected in the concept $\mathcal{C}$ and the concept cardinality, which can be updated incrementally and stored in just two values.

4.3 Regret Analysis for Generic CD-MABs

At first we consider the CD-MAB setup, where there is no break-point prediction nor the recurrent concept identification. Assume to have a stationary stochastic MAB policy $\mathcal{P}$ ensuring an upper bound on the expected pseudo-regret of $C_1(\log N) + C_2$ over a time horizon of N for the stochastic stationary MAB problem (being $C_1, C_2 \in \mathbb{R}^+$ suitable constants), and a CDT procedure $\mathcal{D}$ ensuring an expected detection delay of $\mathbb{E}[D]$ and an expected number of false positive of $\mathbb{E}[F]$. We prove the following:

Theorem 3. *The expected pseudo-regret of a CD-MAB algorithm, where the arm selection is performed using $\mathcal{P}$ with probability $1-\alpha$ and randomly selecting*

an arm with probability α and that uses $\mathcal{D}$ on a generic abruptly changing MAB setting, is upper bounded by:

$$\overline{R}_N(\textit{CD-MAB}) \leq (1 + B_N + \mathbb{E}[F])KM + (B_N + \mathbb{E}[F]) \left(C_1 \log \frac{N}{B_N} + C_2 \right) + \frac{KB_N \mathbb{E}[D]}{\alpha} + \alpha N, \tag{11}$$

where we assume that the CDT requires M samples for each arm to be initialized.

Proof. Due to space limitations, the proof is deferred to Appendix A.

The contribution to the regret in the right-hand side of Equation (11) is composed by the following components (from left to right): *i*) the samples required for the initialization of the CDT at the beginning of the learning procedure and each time a change is detected, *ii*) the regret of the stationary MAB procedure repeated every time a change is detected, *iii*) the loss due to the detection delay, and *iv*) the loss due to random sampling performed over the time horizon N.

This result generalizes that in [14], in which the authors provide an upper bound to the expected pseudo-regret of the same order for an algorithm using as stationary MAB procedure the UCB1 algorithm [3]. In the same work, the authors also present theoretical results for the specific choice of UCB1 as stationary MAB and CUSUM as CDT and provide a bound of the order of $O(\sqrt{B_N N \log \frac{N}{B_N}})$, when the values of the threshold of the CUSUM h and the exploration parameter α are adequately set. Notably, Theorem 3 provides the same order of pseudo-regret of the CUSUM-UCB when substituting in Equation (11) the guarantees provided by CUSUM and those of UCB1.

4.4 Regret Analysis for the Break-point Prediction Procedure

Here, we analyse the theoretical guarantees provided by a specific instance of the BR-MAB algorithm, using CUSUM-UCB as CD-MAB procedure and using a generic break-point prediction procedure $\mathcal{B}$. Indeed, updating CUSUM-UCB after each detection, exploiting a bounded number of reward values recovered by the break-point prediction procedure $\mathcal{B}$, allow us to provide theoretical guarantees on the performance of BR-MAB. We show that:

Theorem 4. *Consider the BR-MAB algorithm with the CUSUM-UCB as CD-MAB procedure and a break-point procedure $\mathcal{B}$, s.t. number of rewards selected by this procedure are less than $\frac{\xi}{4} \log N_t$. Using such an algorithm on the abruptly changing MAB setting provides an upper bounded on the pseudo-regret of:*

$$R_N(\mathfrak{U}) \leq O\left(\sqrt{N B_N \log N / B_N} \right), \tag{12}$$

where ξ is the parameter used in the UCB bound for the CUSUM-UCB algorithm, $N_t := \sum_i N_{i,t}$ is the number of samples collected from the instant a change has been detected.

Proof. Due to space limitations, the proof is deferred to Appendix A.

We remark that any break-point procedure $\mathcal{B}$ can be adapted to satisfy the constraint in 4, by using $\max\{r, t - \frac{\xi}{4} \log N_t\}$, where r is the round at which $\mathcal{B}$ predicted the break-point and t is the current time instant. Notice that the limitation in terms of samples is required to avoid that the estimated expected value for an arm, used in the CUSUM-UCB to take decisions, is biased significantly by the presence of samples coming from the previous phase.

5 Experiments

In what follows, we conduct experiments to evaluate the empirical improvement provided by the proposed BR-MAB approach on generic CD-MAB algorithms. At first, we present a toy example to show the effect of using the BR-MAB approach on a CD-MAB algorithm. After that, we evaluate the proposed algorithm on synthetically generated data, and a real-world problem of online ads selection.

In the experiments, we evaluated two flavours of our BR-MAB algorithm applied to the CUSUM-UCB algorithm: the former exploiting only the break-point prediction procedure $\mathcal{B}$, denoted from now on with BR-CUSUM-UCB($\mathcal{B}$,/), and the latter using both the break-point prediction procedure $\mathcal{B}$ and the recurrent concept equivalence test $\mathcal{E}$, denoted by BR-CUSUM-UCB($\mathcal{B}$,$\mathcal{E}$). This allows us to separately evaluate the improvements provided solely by the break-point prediction in BR-MAB. We compare our method against: *i*) the UCB1 algorithm [3], an algorithm designed for stationary stochastic bandits, *ii*) D-UCB and *iii*) SW-UCB [9], which are algorithms for non-stationary MAB adopting the passive approach to deal with changes in the environment, *iv*) CUSUM-UCB [14], the version of the CD-MAB algorithm without using our framework. We set the parameters required by each one of the tested algorithms as suggested by the corresponding papers. A summary of the parameters is provided by Table 2 provided in Appendix C. We evaluate the different algorithms in terms of empirical pseudo-regret $R_n(\mathfrak{U})$ over the time horizon. The experiments have been repeated for 200 independent simulations. The code used for the experiments is available at https://github.com/gerlaxrex/BR-MAB.

5.1 Toy Example

The aim of this experiment is to compare the behaviour over time of the upper confidence bounds of the CUSUM-UCB algorithm, BR-CUSUM-UCB($\mathcal{B}$,/), and BR-CUSUM-UCB($\mathcal{B}$,$\mathcal{E}$). In this experiment, we model $K = 2$ arms over a time horizon of $N = 10^5$ with $B_N = 4$ break-points. We tested the three algorithms on an abruptly changing scenario where the expected rewards $\mu_{i,\phi}$ varies over time as depicted in Figure 2a.

In Figures 2b, 2c, and 2d we provide the estimated expected value (solid line) and the confidence bounds (shaded areas) used for the arm selection by the CUSUM-UCB, BR-CUSUM-UCB($\mathcal{B}$,/), and BR-CUSUM-UCB($\mathcal{B}$,$\mathcal{E}$) algorithm,

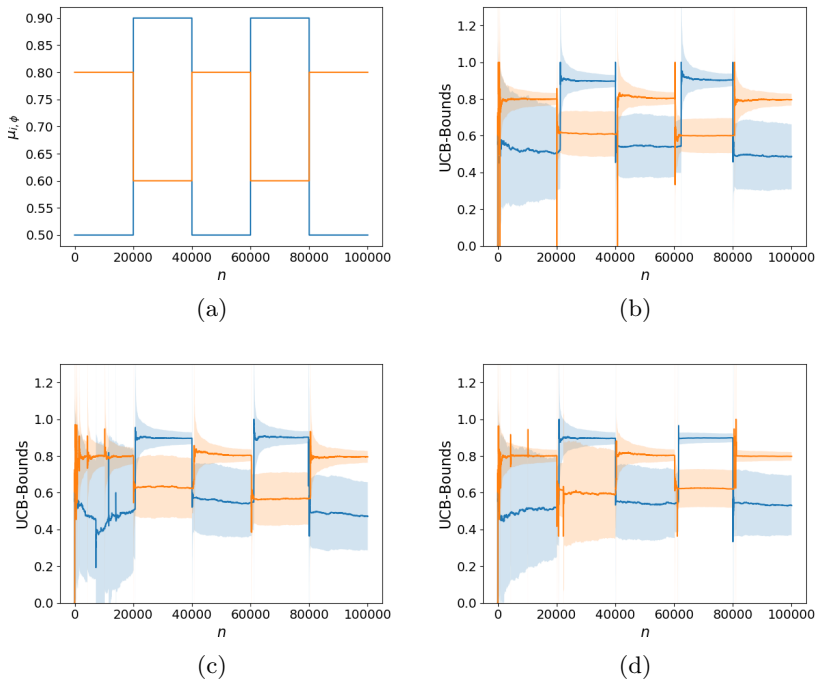
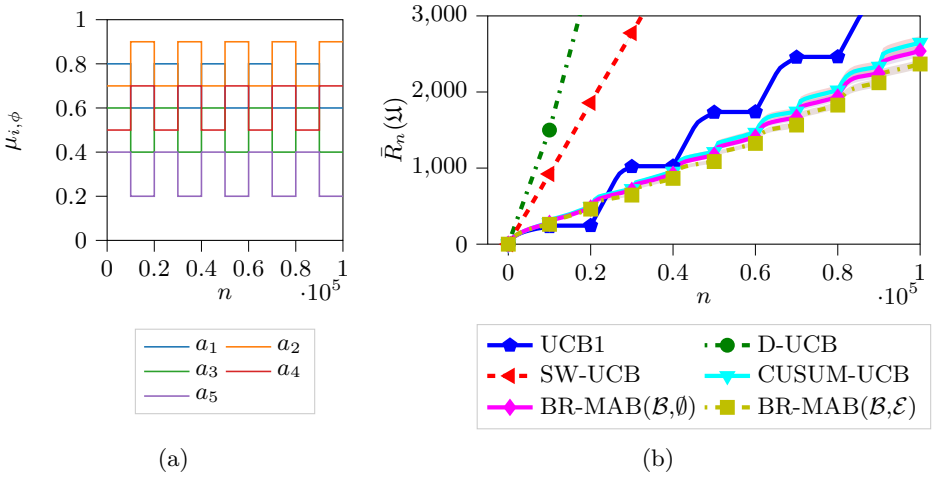


Fig. 2: Toy example: (a) expected rewards for the arms, upper confidence bounds for (b) CUSUM-UCB, (c) BR-CUSUM-UCB($\mathcal{B}$,/), (d) BR-CUSUM-UCB($\mathcal{B}$,$\mathcal{E}$).

respectively. The sole introduction of the $\mathcal{B}$ procedure improves the estimate of the mean value at the beginning of the phases, since the mean values are initialized using the samples collected before the detection of the change. This is evident at times $n = 20,000$ and $n = 40,000$ where the CUSUM-UCB algorithm features downward spikes, while ours take advantage of the samples collected before the detection to reinitialize the empirical expected value of the reward and reduce the variance in reward' estimates.

Comparing Figures 2b and 2d in the interval $60,000 \leq t \leq 100,000$ of, we observe that the test $\mathcal{E}$ to identify recurrent concepts makes the upper confidence bounds tighter, especially those corresponding to the optimal arm in each phase. This means that the amount of exploratory pulls required by BR-CUSUM-UCB($\mathcal{B}$,$\mathcal{E}$) to identify the optimal arm are greatly reduced, which also reduces the regret suffered.

Moreover, the management of recurring concepts also mitigate the impact of false positive detection. This is evident in Figures 2c and 2d, when two false positive detections occurring at $t \approx 7,000$ and $t \approx 12,000$ (small spikes in the figure on the orange arm statistics). While the BR-CUSUM-UCB($\mathcal{B}$,/) algorithm recovers slowly from these false detections, the BR-CUSUM-UCB($\mathcal{B}$,$\mathcal{E}$) algorithm experiences only a slight spike in the mean, while the upper confidence bounds



(a) (b)

Fig. 3: Synthetic setting: (a) reward expected value, (b) empirical pseudo-regret over the learning process. The shaded areas represent the 95% confidence intervals for the mean.

continues to decrease monotonically. This suggests that the reuse of information provided by recurrent concept is also useful to recover promptly to a false positive detection of the CDT adopted in the CD-MAB.

5.2 Synthetic Setting

The first experiment was carried out in a setting with $K = 5$ arms, on a time horizon of $N = 10^5$ rounds, with $B_N = 9$ break-points, evenly distributed over time. The expected reward of the arms over time is depicted in Figure 3a.

Results Figure 3b shows the empirical pseudo-regret $R_n(\mathfrak{U})$ over time of the different algorithms. In this specific setting, the two passive approaches D-UCB and SW-UCB are those providing the worst performances, since the value of the regret gets larger than $3,500$ after $t \approx 30,000$. UCB1, which in principle should not be able to adapt after changes, is performing better than passive approaches. This is due to the fact that the arm originally optimal in the first phase $\mathcal{F}_1$ is also optimal in the phases $\mathcal{F}_3$, $\mathcal{F}_5$, $\mathcal{F}_7$, and $\mathcal{F}_9$, therefore, the information gathered in the past are helping in the selection performed by UCB1. Conversely, in the even index phases, where a different arm is optimal, the UCB1 algorithm experience an almost linear increase of the regret, due to the fact that it focus on the arm optimal in the initial phase, overall providing evidence that it is not suited for such a scenario. After $t = 25,000$ rounds the CUSUM-UCB keeps its regret below all the above-mentioned algorithms, showing the superiority of the active approaches. Even using this approach, we have that the increase of the regret is accentuated as soon as a change occurred. This effect is mitigated by BR-CUSUM-UCB($\mathcal{B}$,/) thanks to the samples recovered by the $\mathcal{B}$ procedure. Indeed,

Table 1: Regret $R_N(\mathfrak{U})$ at the end of the time horizon N.

Algorithm	Synthetic Setting	Yahoo! Setting
UCB1	$3,193 \pm 17$	908 ± 5
D-UCB	$17,758 \pm 8$	$1,653 \pm 1$
SW-UCB	$9,307 \pm 15$	$1,599 \pm 1$
CUSUM-UCB	$2,719 \pm 84$	831 ± 35
BR-CUSUM-UCB($\mathcal{B}$,/)	$2,619 \pm 80$	805 ± 34
BR-CUSUM-UCB($\mathcal{B}, \mathcal{E}$)	$\mathbf{2,273 \pm 61}$	$\mathbf{682 \pm 21}$

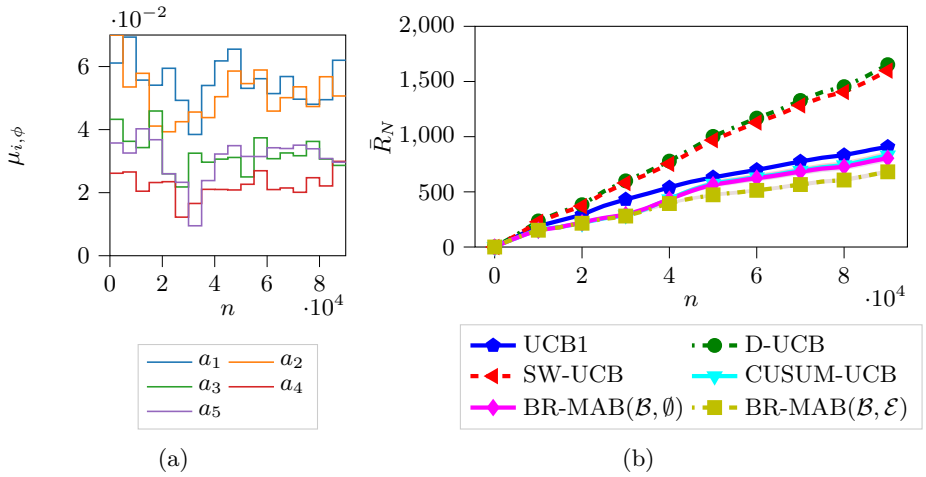
on average BR-CUSUM-UCB($\mathcal{B}$,/) is performing better than CUSUM-UCB but no statistical evidence for its superior performance is provided, even at the end of the learning period (the shaded areas are overlapping). Conversely, the BR-CUSUM-UCB($\mathcal{B}$, $\mathcal{E}$) is getting a significant advantage in terms of pseudo-regret, by exploiting the fact that all the even phases are recurrent, as well as all the odd ones. The proposed approach is able to incrementally gain advantage over the other algorithms as the number of recurring phases increases.

The regret at the end of the time horizon N is presented in Table 1, second column. Even if there is no significance that the BR-CUSUM-UCB($\mathcal{B}$,/) algorithm performs better than CUSUM-UCB, on average it decreases the pseudo-regret of $\approx 4\%$ in the synthetic setting. Instead, the BR-CUSUM-UCB($\mathcal{B}$, $\mathcal{E}$) provides a significant improvement of $\approx 15\%$ over CUSUM-UCB. This suggests that the information provided by previous phases, in a setting where the environment presents recurrent phases multiple times, might provide a large improvement to nonstationary MAB algorithms.

5.3 Yahoo! Setting

The second experiment used a dataset of click percentage of online articles, more specifically the ones corresponding to the first day ($T = 90,000$) of the Yahoo! Dataset [23]. In this setting the use of a CDT-MAB approach is appropriate since the user behaviour is known to vary over time, and the recommender system wants to maximize the visualization of the most interesting article at each time over the day. We selected $K = 5$ article at random from the available ones, and a phase $\mathcal{F}_\phi$ is defined computing their average click-through rate each $5,000$ seconds and keeping the arms expected reward constant over this period.

Results The results corresponding to the empirical pseudo-regret are presented in Figure 4. Also in this scenario, the two passive approaches, D-UCB and SW-UCB, are providing the worst performance, with a regret at the end of the time horizon of almost twice the value of the other considered algorithms. UCB1 is performing worse than CUSUM-UCB, which means that in this specific setting, the active approach is a valid solution to tackle this problem. The adoption of the break-point prediction procedure $\mathcal{B}$ used by BR-CUSUM-UCB($\mathcal{B}$,/) is not achieving a significant improvement, even when looking at Table 1, third



(a) (b)

Fig. 4: Yahoo! setting: (a) reward expected value, (b) empirical pseudo-regret over the learning process. The shaded areas are the 95% confidence intervals.

column, where we have a slightly smaller regret a the end of the time horizon of $\approx 2.5\%$ on average. Conversely, when adopting also a technique to integrate the samples coming from recurrent concepts, we have a significant improvement in terms of regret for $t > 40,000$ w.r.t. the one of CUSUM-UCB, which leads to a improvement of $\approx 15\%$ at the end of the time horizon. This strengthens the idea that the presented BR-MAB framework outperforms standard active techniques.

6 Conclusion and Future Works

We propose BR-MAB, a general framework extending CD-MAB algorithms to better handle non-stationary MAB setting. The rationale behind BR-MAB consists in gathering, after having detected a change, all the possible information that is consistent with the current state of the arm. More specifically, BR-MAB adopts a break-point prediction technique to recover rewards acquired in between the detection and the unknown change-time instant, and a procedure to identify recurrent phases of the arm. Our analysis demonstrates that including information collected by the break-point prediction procedure preserves the guarantees on the pseudo-regret in the CUSUM-UCB case. Moreover, experiments indicate that identifying recurrent concepts is beneficial in terms of accumulated regret, also thanks to a better recovery after false positive detections. Ongoing work concerns a further investigation to achieve tighter theoretical guarantees on specific settings, like the case of changes affecting all the arms simultaneously.

References

1. Alippi, C., Boracchi, G., Roveri, M.: Just-in-time classifiers for recurrent concepts. IEEE transactions on neural networks and learning systems **24**(4), 620–634 (2013)

2. Auer, P.: Using confidence bounds for exploitation-exploration trade-offs. Journal of Machine Learning Research **3**(Nov), 397–422 (2002)
3. Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite-time analysis of the multiarmed bandit problem. Machine learning **47**(2-3), 235–256 (2002)
4. Aziz, M., Kaufmann, E., Riviere, M.K.: On multi-armed bandit designs for dose-finding clinical trials. Journal of Machine Learning Research **22**, 1–38 (2021)
5. Basseville, M., Nikiforov, I.V.: Detection of Abrupt Changes - Theory and Application. Prentice Hall (1993)
6. Besson, L., Kaufmann, E.: The generalized likelihood ratio test meets klucb: an improved algorithm for piece-wise non-stationary bandits. arXiv preprint arXiv:1902.01575 (2019)
7. Bubeck, S., Cesa-Bianchi, N.: Regret analysis of stochastic and nonstochastic multi-armed bandit problems. CoRR **abs/1204.5721** (2012)
8. Cao, Y., Wen, Z., Kveton, B., Xie, Y.: Nearly optimal adaptive procedure with change detection for piecewise-stationary bandit. In: AISTATS. pp. 418–427 (2019)
9. Garivier, A., Moulines, E.: On upper-confidence bound policies for switching bandit problems. In: ALT. pp. 174–188 (2011)
10. Hartland, C., Gelly, S., Baskiotis, N., Teytaud, O., Sebag, M.: Multi-armed bandit, dynamic environments and meta-bandits (Nov 2006), https://hal.archives-ouvertes.fr/hal-00113668/file/MetaEve.pdf, working paper
11. Hawkins, D.M., Qiu, P., Kang, C.W.: The changepoint model for statistical process control. Journal of quality technology **35**(4), 355–366 (2003)
12. Hinkley, D.: Inference about the change-point from cumulative sum tests. Biometrika **58** (12 1971)
13. Italia, E., Nuara, A., Trovò, F., Restelli, M., Gatti, N., Dellavalle, E.: Internet advertising for non-stationary environments. In: AMEC. pp. 1–15 (2017)
14. Liu, F., Lee, J., Shroff, N.B.: A change-detection based framework for piecewise-stationary multi-armed bandit problem. In: AAAI. (2018)
15. Mellor, J.C., Shapiro, J.L.: Thompson Sampling in switching environments with Bayesian online change point detection. CoRR **abs/1302.3721** (2013)
16. Nuara, A., Trovo, F., Gatti, N., Restelli, M.: A combinatorial-bandit algorithm for the online joint bid/budget optimization of pay-per-click advertising campaigns. In: AAAI. vol. 32 (2018)
17. Parvin, M., Meybodi, M.R.: Mabrp: A multi-armed bandit problem-based energy-aware routing protocol for wireless sensor network. In: AISP. pp. 464–468. IEEE (2012)
18. Ross, G.J., Adams, N.M.: Two nonparametric control charts for detecting arbitrary distribution changes. Journal of Quality Technology **44**(2), 102–116 (2012)
19. Ross, G.J., Tasoulis, D.K., Adams, N.M.: Sequential monitoring of a bernoulli sequence when the pre-change parameter is unknown. Computational Statistics **28**(2), 463–479 (2013)
20. Schuirmann, D.J.: A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. Journal of pharmacokinetics and biopharmaceutics **15**(6), 657–680 (1987)
21. Trovò, F., Paladino, S., Restelli, M., Gatti, N.: Improving multi-armed bandit algorithms in online pricing settings. International Journal of Approximate Reasoning **98**, 196–235 (2018)
22. Trovò, F., Paladino, S., Restelli, M., Gatti, N.: Sliding-window Thompson Sampling for non-stationary settings. Journal of Artificial Intelligence Research **68**, 311–364 (2020)
23. Yahoo!: R6b - yahoo! front page today module user click log dataset, ver. 2.0 (2011)