

Multi-Armed Bandits with Non-Stationary Means

Ben Jones, Jennifer Brennan, Yiqun Chen, Jakub Filipek

May 2021

1 Introduction

For all the algorithms we have studied in this class so far, we have evaluated regret relative to the best arm in hindsight. In some situations trying to pick just one arm is an inappropriate model. As an example, let's consider a music recommendation system. Spotify should not recommend the same song all year long; we might expect that love songs are popular in February, party music in the summer, and holiday music in the winter. This reflects changing preferences over time. A more appropriate regret model in this cases is relative to the best arm at each time step. Let's formalize this notion: at each time $t = 1, \dots, T$, each of the arms $a = 1, \dots, K$ has some true (unknown to the learner) mean $\mu_t(a)$. The learner's notion of regret compares to the *best arm at every time step*,

$$R_T := \sum_{t=1}^T \max_{a \in [K]} \mu_t(a) - \mathbb{E} \left[\sum_{t=1}^T \mu_t(a_t) \right] \quad (1)$$

where a_t is the arm selected by the algorithm at time t . We will measure regret in terms of the *number of switches* L , where a switch occurs whenever some arm's mean changes (even if that doesn't change the identity of the best arm). The regret in (1) is known as the *dynamic regret*, in contrast to the static regret in stationary MAB problems.

Do our traditional multi-armed bandit algorithms work in the context of this regret? The answer is no. As an example, let's take a look at what happens if we try standard UCB on a switching instance, shown in Figure 1. In this instance, all reward is on arm 1 in the first half, and all on arm 0 in the second half. We can see that UCB quickly updates its estimate of arm 0's mean after the switch to align well with the true mean, but it does not make a corresponding shift for arm 1. This is reflected in climbing regret after the halfway point, and we can see the fraction of pulls allocated to each arm converges to be equal by the end of the run. The core problem here is that traditional multi-armed bandit approaches do not *forget* old information — we'll need new approaches that are able to adapt to switches. In order to evaluate any new approaches, we should first understand the limits of how well any algorithm could hope to do in the switching setting.

2 Lower Bounds - Stochastic is as hard as adversarial

In this section, we will consider the following simplified version stochastic switching bandit model,

$$X_t(a) = \mu_t(a) + \eta_t, \eta_t \stackrel{i.i.d.}{\sim} \mathcal{N}(0, 1), \quad (2)$$

where $\mu_t(a)$ is the mean reward of arm $a \in 1, \dots, K$ at time $t = 1, \dots, T$, and the player observes the reward $X_t(a)$. Given a list of actions a_t and reward vector μ , we will measure the performance of algorithms using the dynamic or non-stationary regret

$$R_T(\mu) = \sum_{t=1}^T \max_{i \in [k]} \mu_t(i) - \mathbb{E} \left[\sum_{t=1}^T \mu_t(a_t) \right], \quad (3)$$

where the expectation is taken over randomness of the noise η_t in (2) and the action sequence $a_t, t = 1, \dots, T$.

Theorem 1 (Theorem 31.2 in Lattimore and Szepesvári (2020)). *Let $K = 2$, and fix $\Delta \in (0, 1)$ and a policy π . Consider the two-arm stochastic bandit problem with the following mean reward vector μ : $\mu_t(i) = \mu_i$, $i = 1, 2$ is constant for both arms, and $\Delta = \mu_1 - \mu_2 > 0$. If the expected regret $R_T(\mu)$ of π on bandit μ satisfies $R_T(\mu) = o(T)$, then for sufficiently large T , there exists a non-stationary bandit μ' with at most two change points and $\min_{t \in [T]} |\mu'_t(1) - \mu'_t(2)| \geq \Delta$ such that $R_T(\mu') \geq T/(22R_T(\mu))$.*

Before proving Theorem 1, we remark that it is quite insightful despite the simple set-up. In particular, Theorem 1 implies that if a policy enjoys $\log T$ regret (e.g., UCB in the stochastic MAB setting), then its worst case regret against non-stationary bandits with at most two changes in the mean reward is lower bounded by $\Omega(n/\log n)$. Similarly, if a policy enjoys $R_T(\mu) = \mathcal{O}(\sqrt{T})$, then its minimax regret is at least $\Omega(\sqrt{T})$ on some non-stationary bandit. In particular, this implies that even in the asymptotic sense, algorithms like Exp3.S (Auer et al., 2002) is optimal. An informal intuition for the pessimistic result is that any algorithm that anticipates the possibility of an abrupt change in the optimal arm must frequently explore all sub-optimal arms to ensure that no change has occurred.

We need the following Lemmas to prove Theorem 1.

Lemma 2 (Theorem 14.2 in Lattimore and Szepesvári (2020)). *Let P and Q be probability measures on the same measurable space (Ω, \mathcal{F}) , and let $A \in \mathcal{F}$ be an arbitrary event. Then,*

$$P(A) + Q(A^c) \geq \frac{1}{2} \exp(-D(P\|Q)), \quad (4)$$

where $D(P\|Q)$ is the Kullback-Leibler divergence between P and Q .

Lemma 3 (Lemma 15.1 in Lattimore and Szepesvári (2020)). *Let $\nu = (P_1, \dots, P_k)$ be the reward distribution associated with one k -arm bandit, and $\nu' = (P'_1, \dots, P'_k)$ be the reward distribution associated with another k -arm bandit. Fix some policy π and let \mathbb{P}_ν (and $\mathbb{P}_{\nu'}$) denote the probability measures on the stochastic bandit models induced by T -round interactions of π and ν (and ν' , respectively). Then*

$$D(\mathbb{P}_\nu\|\mathbb{P}_{\nu'}) = \sum_{i=1}^k \mathbb{E}_\nu[T_i] D(P_i\|P'_i), \quad (5)$$

where T_i is the number of times the arm i was pulled by policy π .

We now proceed to prove Theorem 1.

Proof. Consider a partition $(S_j)_{j=1}^L$ of $[n]$ into L intervals of equal elements. Let $\mathbb{P}_\mu, \mathbb{E}_\mu$ denote the probabilities and expectations with respect to the model (2) with mean reward μ , and let $\mathbb{P}_{\mu'}, \mathbb{E}_{\mu'}$ denote the probabilities and expectations with respect to the model (2) with mean reward μ' .

Let $T_i = \sum_{t=1}^T 1\{a_t = i\}$ be the number of times the algorithm chooses action i , we have

$$\mathbb{E}_\mu[T_2] = \mathbb{E}_\mu \left[\sum_{t=1}^T 1\{a_t = 2\} \right] = \mathbb{E}_\mu \left[\sum_{j=1}^L \sum_{t \in S_j} 1\{a_t = 2\} \right], \quad (6)$$

which implies that there exists $j^* \in [L]$ such that

$$\mathbb{E}_\mu \left[\sum_{t \in S_{j^*}} 1\{a_t = 2\} \right] = \min_{j \in [L]} \mathbb{E}_\mu \left[\sum_{t \in S_j} 1\{a_t = 2\} \right] \leq \frac{1}{L} \mathbb{E}_\mu \left[\sum_{j=1}^L \sum_{t \in S_j} 1\{a_t = 2\} \right] = \frac{\mathbb{E}_\mu[T_2]}{L}. \quad (7)$$

Now define the mean reward μ' such that $\mu'_t(1) = \mu_t(1), \forall t \in [T]$, and $\mu'_t(2) = \begin{cases} \mu_t(2) + \epsilon & \text{if } t \in S_{j^*} \\ \mu_t(2) & \text{otherwise} \end{cases}$,

where $\epsilon = \sqrt{\frac{2L}{\mathbb{E}_\mu[T_2]}}$. Then, we have

$$\mathbb{P}_\mu \left(\sum_{t \in S_{j^*}} 1\{a_t = 2\} \geq \frac{|S_{j^*}|}{2} \right) + \mathbb{P}_{\mu'} \left(\sum_{t \in S_{j^*}} 1\{a_t = 2\} < \frac{|S_{j^*}|}{2} \right) \stackrel{1.}{\geq} \frac{1}{2} \exp(-D(\mathbb{P}_\mu\|\mathbb{P}_{\mu'})) \stackrel{2.}{\geq} \frac{1}{2} \exp\left(-\frac{\mathbb{E}_\mu[T_2]\epsilon^2}{2L}\right) \stackrel{3.}{=} \frac{1}{2e}, \quad (8)$$

where 1. follows from Lemma 2, and 2. follows from Lemma 3 and applying the facts that (i) the expected number of pulls of arm 2 is $\mathbb{E}_\mu[T_2]/L$, and (ii) $D(\mathcal{N}(\mu(2), I), \mathcal{N}(\mu'(2), I)) = \|\mu(2) - \mu'(2)\|_2^2$. Finally, 3. follows from our choice of ϵ .

Moreover,

$$\mathbb{P}_\mu \left(\sum_{t \in S_{j^*}} 1\{a_t = 2\} \geq \frac{|S_{j^*}|}{2} \right) \stackrel{1.}{\leq} \frac{2}{|S_{j^*}|} \mathbb{E}_\mu \left(\sum_{t \in S_{j^*}} 1\{a_t = 2\} \right) \stackrel{2.}{\leq} \frac{2\mathbb{E}_\mu[T_2]}{L|S_{j^*}|} \stackrel{3.}{\leq} \frac{1}{\Delta^2|S_{j^*}|}. \quad (9)$$

Here, 1. follows from Markov's inequality; 2. follows from the linearity of expectation. In 3., we choose $L = \lceil 2\Delta^2\mathbb{E}_\mu[T_2] \rceil$ and take T large enough so that $L \leq T$. In particular, this choice of L implies that $\epsilon \geq 2\Delta$ so that μ' satisfies $\min_{t \in [T]} |\mu'_t(1) - \mu'_t(2)| \geq \Delta$.

Therefore,

$$\begin{aligned} R_T(\mu') &\stackrel{1.}{\geq} (\epsilon - \Delta) \cdot \mathbb{E}_{\mu'} \left[\sum_{t \in S_{j^*}} 1\{a_t \neq 2\} \right] \stackrel{2.}{\geq} (\epsilon - \Delta) \cdot \mathbb{P}_{\mu'} \left(\sum_{t \in S_{j^*}} 1\{a_t \neq 2\} \geq \frac{|S_{j^*}|}{2} \right) \cdot \frac{|S_{j^*}|}{2} \\ &\stackrel{3.}{\geq} \frac{(\epsilon - \Delta)|S_{j^*}|}{2} \left(\frac{1}{2e} - \frac{1}{\Delta^2|S_{j^*}|} \right) \stackrel{4.}{\geq} \left\lfloor \frac{T}{L} \right\rfloor \frac{\Delta}{4e} - \frac{1}{2\Delta} \stackrel{5.}{\geq} \frac{1}{8e} \frac{T}{\Delta \mathbb{E}_\mu[T_2]} - \frac{1}{2\Delta}. \end{aligned} \quad (10)$$

Here, 1. follows from only considering the regret for $t \in S_{j^*}$ and 2. follows from Markov's inequality. In step 3., we combine the results in (8) and (9). Finally, we apply the definition of the partition S_j and L in steps 4 and 5, respectively.

Recall that $R_T(\mu) = \Delta \mathbb{E}_\mu[T_2]$; therefore if $R_T(\mu) = o(T)$, then we know that for sufficiently large T , we have $R_T(\mu') \geq \frac{1}{\lceil 8e \rceil} \frac{T}{R_T(\mu)} = \frac{1}{22} \frac{T}{R_T(\mu)}$. \square

3 A Brief History of Non-Stationary Bandits

Interactive learning in non-stationary context has been a problem of interest for decades. An early line of work to explicitly frame this as a bandit problem was initiated by Gittins (1974), which simplified the problem by assuming that only the current best arm could change. This avoids the difficulties of sampling non-optimal arms without accruing too much regret. The restless bandit problem (Whittle, 1988) does not make this simplifying assumption, instead assuming that all of the means can change by some known stochastic process. This is a “known hard problem.” In 2002, Auer (Auer et al., 2002) re-framed non-stationarity as an adversarial process — now there did not need to be any statistical process or distribution behind the changes. In addition to introducing the concept of best-arm-in-hindsight regret to make the fully adversarial case tractable, he also introduced the notion of measuring regret against an arbitrary baseline policy, which could be parameterized by a *hardness* measured by the number of times the baseline switches which arm it is pulling. This is the first instance of what was later formalized by Yu and Mannor (2009) and Garivier and Moulines (2011a) as “switching regret.” Other branches of work measure regret relative to absolute variation of arm means (Slivkins and Upfal, 2008; Besbes, Gur, and Zeevi, 2014), which coined “dynamic regret.” To differentiate it from the switching case. In this document we use “dynamic regret” to refer to the switching case, following our textbook. Until recently, algorithms that solve the switching bandit problem (see Table 1) have required parameter tuning with advanced knowledge of the number of switches in order to achieve optimal regret. Recently, several works (see Table 2) have attempted to drop this requirement, at the expense of suboptimal regret. AdSwitch (Auer et al., 2019), and concurrent work (Chen et al., 2019) are the first works to achieve optimal regret *without* needing this additional information.

AdSwitch builds on insights from much previous work, each of which solves an additional layer of problems. The first is that in a switching bandit, any arm could change, so we need to sample non-optimal arms more frequently to see if they get better. In Auer et al. (2002), EXP3.S is proposed, which adds an extra uniform term to each arm's weight at every step of EXP3 to encourage more exploration. The knowledge of the number of switches is needed to tune this weight; intuitively, more switches means you need to check sub-optimal arms more frequently. As we saw with the UCB example earlier, just sampling sub-optimal

Algorithm	Regret	Required knowledge
EXP3.S (Auer et al., 2002)	$\mathcal{O}\left(\sqrt{KLT \log(KT)}\right)$	L
	$\mathcal{O}\left(L\sqrt{KT \log(KT)}\right)$	
EXP4	$\mathcal{O}\left(\sqrt{KLT \log(KT/L)}\right)$	L
Online Mirror Descent	$\mathcal{O}\left(\sqrt{KLT \log(KT/L)}\right)$	L
	$\mathcal{O}\left(L\sqrt{KT \log(KT/L)}\right)$	
Discounted UCB (Kocsis and Szepesvári, 2006; Garivier and Moulines, 2011b)	$\mathcal{O}\left(K\sqrt{LT \log T}\right)$	L
Sliding-window UCB (Garivier and Moulines, 2011b)	$\mathcal{O}\left(K\sqrt{LT \log T}\right)$	L
ADSWITCH (Auer et al., 2019)	$\mathcal{O}\left(\sqrt{KLT \log T}\right)$	

Table 1: A selected summary of existing results on non-stationary multi-arm bandit problem. Here, we denote by T the number of rounds, L the number of changes in the mean reward, and K the number of arms. We note that a known lower bound for *any policy* is $\Omega(\sqrt{KLT})$.

arms does not necessarily stop us from getting stuck on old best-arms. To do this you need some sort of forgetting, either down-weighting (Kocsis and Szepesvári, 2006) or throwing out (Garivier and Moulines, 2011a) older samples in case they are out-of-date. Again, knowing the number of switches in advance lets you tune how much or how frequently to forget. The problem with dropping out old samples is that it makes our estimates less accurate — ideally we should only be throwing out information when there actually is a switch. Detecting these switches is the domain of Change Point Analysis; a topic too broad to cover here, we refer the reader to Aminikhanghahi and Cook (2017) for a survey of methods. In the context of switching bandits, Hartland et al. (2006) used change point detection to predict when a switch occurs, and tried both down-weighting and dropping older samples, which they evaluated empirically. Later, Yu and Mannor (2009), Karnin and Anava (2016), Auer, Gajane, and Ortner (2018), and Luo et al. (2018) incorporated change point detection into their bandit algorithms with regret guarantees.

Change point detection is not infallible, and it can be susceptible to problems if changing arms are sampled too infrequently so that an actual switch is within the uncertainty of the mean given how many samples we have. To circumvent this, Karnin and Anava (2016) sample the same arm consecutively to get a good current mean estimate (for 2-armed contextual bandits), under the assumption that switches in the middle of a consecutive sampling are unlikely. Luo et al. (2018) extend this idea to multi-armed contextual bandits. The final key contribution that AdSwitch builds on is the realization that we only care about changes to an arm larger than their optimality gap, and that the number of samples needed to detect a change is smaller for larger gaps, which presents an opportunity to balance sampling of suboptimal arms based on their gap sizes. This was used to get optimal regret for the 2-arm case by Auer, Gajane, and Ortner (2018), the direct predecessor to AdSwitch.

4 Adaptively Tracking the Best Bandit Arm with an Unknown Number of Distribution Changes

This paper improves upon past results by achieving $\mathcal{O}(\sqrt{KLT \log T})$ regret (which matches the lower bounds up to logarithmic factors) *without knowledge of the number of switches*. Recall the core challenge of the switching bandits setting: you must sample suboptimal arms frequently enough to know whether they have changed, but not so frequently that you incur large regret when they have stayed suboptimal. Attaining the lower bound without knowing the number of switches is difficult because, at first glance, it seems like you

Algorithm	Regret	MAB Variant
Karnin and Anava (2016)	$\mathcal{O}(V^{.82}T^{.18} + T^{.77})$	2-armed Contextual
Luo et al. (2018)	$\mathcal{O}(L^{1/4}T^{3/4}V^{1/5}T^{4/5}T^{3/4})$	Contextual
Auer, Gajane, and Ortner (2018)	$\mathcal{O}(\sqrt{LT})$	2-arm
Cheung, Simchi-Levi, and Zhu (2019)	$\mathcal{O}(V^{1/3}T^{2/3} + T^{3/4})$	Linear
	$\mathcal{O}(\sqrt{LT} + T^{3/4})$	General

Table 2: A selected summary of existing results on non-stationary multi-arm bandit problem. Here, we denote by T the number of rounds, L the number of changes in the mean reward (or V the total variation, see Equation (18)), and K the number of arms. We note that a known lower bound for *any policy* is $\Omega(\sqrt{KLT})$ or $\tilde{\mathcal{O}}(V^{\frac{1}{3}}T^{\frac{2}{3}})$.

must know the total number of switches in order to choose a sampling rate. For example, suppose an arm has a suboptimality gap of Δ compared with the best arm. In order to balance the regret due to sampling that arm with probability p when it doesn't change ($pT\Delta K$ over all K arms) against the regret due to missing out on sampling that arm if it became the best arm ($L\Delta\frac{1}{p\Delta^2}$ across all L changepoints), it would suggest a sampling rate that sets these two quantities equal:

$$p = \frac{\sqrt{\frac{L}{KT}}}{\Delta}. \quad (11)$$

4.1 The AdSwitch Approach

What can be done in the absence of knowledge of L ? This paper addresses this question by providing an elimination-style algorithm in which eliminated arms are occasionally checked for changes. This paper contains three main innovations:

- Instead of sampling each suboptimal arm with some probability, and waiting for those samples to accumulate to detect a change, this paper introduces the notion of *sampling obligations*. With some probability, the algorithm decides to take enough samples to detect a change in a given suboptimal arm. These samples are collected nearly sequentially, which reduces the chance that a changepoint occurs in the middle of sampling the arm. The frequency of enqueueing sampling obligations is the second innovation.
- If an eliminated arm improves by twice as much, it should be sampled twice as often, but it only requires a quarter as many samples to identify the change. We can hedge against any size change in the suboptimal arms by sampling for a change of $2^m\Delta$ in $(2^m\Delta)^{-2}$ steps, which means we can sample for a change of size $2^m\Delta$ with probability proportional to $2^m\Delta$, and if $m = 0, 1, 2, \dots$, the number of samples we spend on suboptimal arms is a summable sequence.
- Finally, to decide with what probability to check a suboptimal arm for a gap of size ε , this paper uses

$$p_\varepsilon = \frac{\sqrt{\frac{\ell}{KT}}}{\Delta} \quad (12)$$

where ℓ , the number of changepoints detected so far, replaces L , the known number of changepoints, in Equation (11). Even though ℓ is an underestimate of L , the authors show that this sampling schedule is sufficient to get optimal regret.

An annotated version of the algorithm is shown in Figure 2. We also note that this algorithm has high computational complexity, driven by the $O(Kt^3)$ complexity of a naive implementation of changepoint detection for good arms. This can be reduced with caching, or by discretizing time into intervals for the purpose of changepoint detection.

4.2 Proof Sketch and Analysis Details

We begin with an interesting analysis technique from the paper: the idea of “intervals without change.” Recall that one innovation of this paper was to enqueue an entire sampling obligation at once to detect a change of size ε in a suboptimal arm, instead of sampling each arm with some probability at every time. The purpose was to roughly sample arms consecutively, so that arms didn’t change between samples. However, there is still no way to ensure that the arms don’t change within a single “check” of a bad arm. The analysis addresses this by dividing the time horizon into “intervals of no change” - defined as time intervals where the means don’t switch *and* the algorithm does not detect a changepoint (therefore starting a new episode). The claim is that there are at most $2L$ such segments, so running the analysis over these intervals (instead of the L intervals where the arms are not changing) only increases the regret by a factor of 2. We sketch the argument below:

Proposition 1. *If the good event holds (ie, all estimated quantities stay within their confidence intervals), there are no false positive changepoint detections.*

Proof. The proof follows immediately from the way we set up changepoint detection - confidence intervals are constructed on the means of an arm for subsets of the episode, and a changepoint is declared if the confidence intervals on means at different time intervals do not overlap. \square

Proposition 2. *If there are L changepoints and the good event holds, then there are at most $2L$ “intervals of no change,” in which there is no changepoint and the algorithm does not start a new episode.*

Proof. This hinges on the absence of false positive changepoint detections; we know there are at most L changepoints detected, and each changepoint can only add at most one interval without change to the total. See Figure 3 for a sketch. \square

The following proof sketch proceeds over these intervals of no change. The full analysis includes several cases detailing the possible ways the arms can switch, but in this treatment, we just consider regret incurred from sampling suboptimal arms, and from failing to detect changed arms.

Proposition 3. *If the good event holds (all estimates stay within their confidence intervals), the total regret is $\tilde{O}(\sqrt{LKT})$.*

Sketch. We consider two cases: regret incurred by sampling suboptimal arms when they have not changed, and regret incurred by failing to sample an eliminated arm that is now good.

We begin with the first case. If the arm is Δ -suboptimal, and the algorithm takes $n_\varepsilon \approx \varepsilon^{-2}$ samples with probability $p_\varepsilon \approx \varepsilon \sqrt{\frac{\ell}{KT}}$ at every time step, then the regret incurred is

$$\sum_{\varepsilon=\Delta, 2\Delta, 4\Delta, \dots} p_\varepsilon T n_\varepsilon \Delta \approx \sum_{\varepsilon=\Delta, 2\Delta, 4\Delta, \dots} \frac{\Delta}{\varepsilon} \sqrt{\frac{\ell T}{K}} \quad (13)$$

$$= \sqrt{\frac{\ell T}{K}} \sum_{m=1, 2, 4, \dots} \frac{1}{m} \quad (14)$$

$$\leq 2\sqrt{\frac{\ell T}{K}} \quad (15)$$

Summing over K inferior arms, the regret is $\tilde{O}(\sqrt{\ell KT})$, which, since $\ell \leq L$, is also $\tilde{O}(\sqrt{LKT})$.

The second source of regret we consider is the regret that comes from failing to sample an eliminated arm that has become the best arm since its elimination. In this case, we will incur regret until we perform a sampling obligation that detects this change. A change of size ε will be detected if we perform a check for gaps $\leq \varepsilon$, which happens with probability of order p_ε . Therefore, the regret incurred while waiting to perform this check is

$$\sum_{\ell \in [L]} \varepsilon \cdot \frac{1}{p_\varepsilon} \approx \sum_{\ell \in [L]} \sqrt{\frac{KT}{\ell}} \quad (16)$$

$$\lesssim \sqrt{KTL}. \quad (17)$$

We conclude that the regret from these two sources is $\tilde{O}(\sqrt{LKT})$. \square

4.2.1 Other Cases

We will not cover other three cases, since that analysis is more verbose. However, in all three cases we assume that arm a_t is a bad arm.

2. We are coming back to a bad arm to see if it became optimal, and the gap between it optimal arm has not changed much.
3. We are coming back to a bad arm to see if it became optimal, and it got worse than when we evicted it. We want to minimize sampling in this case, since we suffer large regret, but we still have to check these arms.
4. We are coming back to a bad arm to see if it became optimal, and the best arm got better than when arm a_t was evicted. We also want to minimize sampling in this case, since we suffer large regret. However, since we might choose arm a_t (due to initial estimate of the gap being small) that change in arm mean should be fast to detect.

As we can see in all cases we want to limit the number of times arm is pulled. However, in case 2nd case we limit that number, and show that suffered regret is still $\tilde{O}(\sqrt{KLT})$. In cases 3rd and 4th case we show that the change is quickly detected and the algorithm starts a new episode, where arm is quickly eliminated, thus reducing the regret.

5 Other Topics in Non-Stationary Bandits

5.1 Measuring Non-Stationarity

The works we have discussed so far use L , the total number of switches, to measure the non-stationarity of nature. The tacit assumption is that means are changing abruptly at discrete point in time. Another view of non-stationarity says that means are changing gradually but frequently. Works that take this view use V , the total variation of the reward μ , as a measure of non-stationarity (Besbes, Gur, and Zeevi, 2014):

$$V := \sum_{t=1}^T \sup_{k \in [K]} |\mu_t(k) - \mu_{t+1}(k)|. \quad (18)$$

Table 2 contains a few selected results with V as the measure of non-stationarity; we remark that a corresponding lower bound in this case is $\tilde{O}\left(V^{\frac{1}{3}}T^{\frac{2}{3}}\right)$ (Lattimore and Szepesvári, 2020).

References

- [AC17] Samaneh Aminikhanghahi and Diane J Cook. “A survey of methods for time series change point detection”. In: *Knowledge and information systems* 51.2 (2017), pp. 339–367.
- [AGO18] Peter Auer, Pratik Gajane, and Ronald Ortner. “Adaptively tracking the best arm with an unknown number of distribution changes”. In: *European Workshop on Reinforcement Learning*. Vol. 14. 2018, p. 375. URL: https://ewrl.files.wordpress.com/2018/09/ewrl_14_2018_paper_28.pdf.
- [Aue+02] Peter Auer et al. “The nonstochastic multiarmed bandit problem”. In: *SIAM journal on computing* 32.1 (2002), pp. 48–77. URL: <https://cseweb.ucsd.edu/~yfreund/papers/bandits.pdf>.
- [Aue+19] Peter Auer et al. “Achieving optimal dynamic regret for non-stationary bandits without prior information”. In: *Conference on Learning Theory*. PMLR. 2019, pp. 159–163. URL: <http://proceedings.mlr.press/v99/auer19b/auer19b.pdf>.

- [BGZ14] Omar Besbes, Yonatan Gur, and Assaf Zeevi. “Stochastic multi-armed-bandit problem with non-stationary rewards”. In: *Advances in neural information processing systems* 27 (2014), pp. 199–207. URL: <https://papers.nips.cc/paper/2014/file/903ce9225fca3e988c2af215d4e544d3-Paper.pdf>.
- [Che+19] Yifang Chen et al. “A new algorithm for non-stationary contextual bandits: Efficient, optimal and parameter-free”. In: *Conference on Learning Theory*. PMLR. 2019, pp. 696–726. URL: <http://proceedings.mlr.press/v99/chen19b/chen19b.pdf>.
- [CSLZ19] Wang Chi Cheung, David Simchi-Levi, and Ruihao Zhu. “Learning to optimize under non-stationarity”. In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 1079–1087. URL: <http://proceedings.mlr.press/v89/cheung19b/cheung19b.pdf>.
- [Git74] John Gittins. “A dynamic allocation index for the sequential design of experiments”. In: *Progress in statistics* (1974), pp. 241–266. URL: <https://www.jstor.org/tc/accept?origin=%2Fstable%2Fpdf%2F2335176.pdf>.
- [GM11a] Aurélien Garivier and Eric Moulines. “On upper-confidence bound policies for switching bandit problems”. In: *Proceedings of the 22nd international conference on Algorithmic learning theory*. ALT’11. Espoo, Finland: Springer-Verlag, Oct. 2011, pp. 174–188. URL: https://link.springer.com/chapter/10.1007/978-3-642-24412-4_16.
- [GM11b] Aurélien Garivier and Eric Moulines. “On upper-confidence bound policies for switching bandit problems”. In: *International Conference on Algorithmic Learning Theory*. Springer. 2011, pp. 174–188. URL: <https://arxiv.org/pdf/0805.3415.pdf>.
- [Har+06] Cédric Hartland et al. “Multi-armed bandit, dynamic environments and meta-bandits”. In: (2006). URL: <https://hal.archives-ouvertes.fr/hal-00113668/document>.
- [KA16] Zohar S Karnin and Oren Anava. “Multi-armed bandits: Competing with optimal sequences”. In: *Advances in Neural Information Processing Systems* 29 (2016), pp. 199–207. URL: <https://proceedings.neurips.cc/paper/2016/file/47d1e990583c9c67424d369f3414728e-Paper.pdf>.
- [KS06] Levente Kocsis and Csaba Szepesvári. “Discounted ucb”. In: *2nd PASCAL Challenges Workshop*. Vol. 2. 2006. URL: <https://www.lri.fr/~sebag/Slides/Venice/Kocsis.pdf>.
- [LS20] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, July 2020.
- [Luo+18] Haipeng Luo et al. “Efficient contextual bandits in non-stationary worlds”. In: *Conference On Learning Theory*. PMLR. 2018, pp. 1739–1776. URL: <https://arxiv.org/pdf/1708.01799.pdf>.
- [SU08] Aleksandrs Slivkins and Eli Upfal. “Adapting to a Changing Environment: the Brownian Restless Bandits.” In: *COLT*. 2008, pp. 343–354. URL: <https://www.learningtheory.org/colt2008/papers/45-Slivkins.pdf>.
- [Whi88] Peter Whittle. “Restless bandits: Activity allocation in a changing world”. In: *Journal of applied probability* (1988), pp. 287–298. URL: <https://www.jstor.org/stable/pdf/3214163.pdf>.
- [YM09] Jia Yuan Yu and Shie Mannor. “Piecewise-stationary bandit problems with side observations”. In: *Proceedings of the 26th annual international conference on machine learning*. 2009, pp. 1177–1184.

Performance of UCB on a switching instance

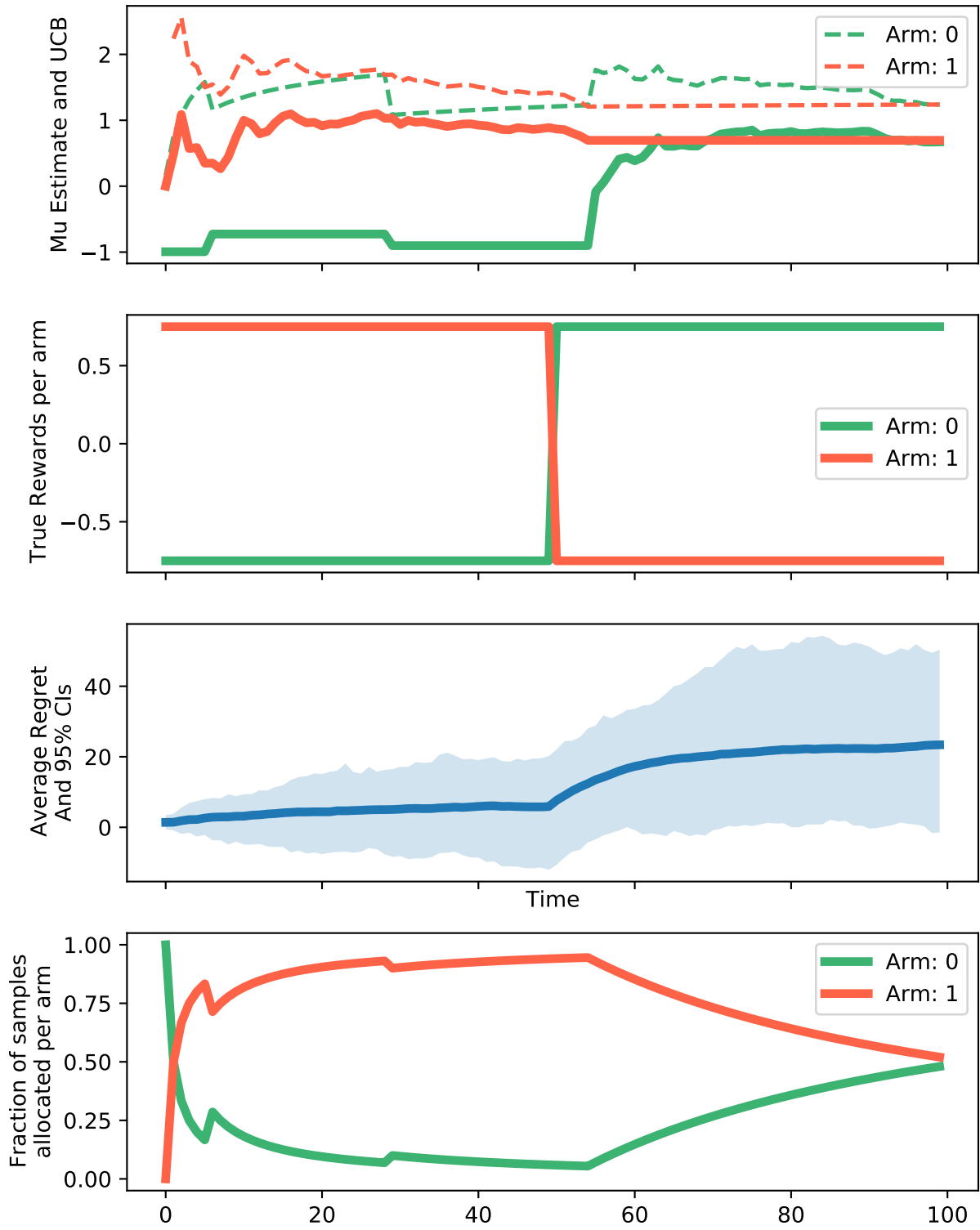


Figure 1: UCB applied to a switching instance.

Algorithm 1 ADSWITCH

1: **Input:** Time horizon T .
 2: **Initialization** $\ell \leftarrow 0, t \leftarrow 0$.
 3: **Start a new episode:**
 4: $\ell \leftarrow \ell + 1$.
 5: Set start of the episode $t_\ell \leftarrow t + 1$.
 6: $\text{GOOD}_{t+1} = \{1, \dots, K\}, \text{BAD}_{t+1} = \{\}$.
 7: **Next time step:**
 8: $t \leftarrow t + 1$.
 9: **Add checks for bad arms:**
 10: For all $a \in \text{BAD}_t$, and all $i \geq 1$ with $2^{-i} \geq \bar{\Delta}_\ell(a)/16$,
 11: with probability $2^{-i} \sqrt{\ell/(KT \log T)}$ add $\mathcal{S}_t(a) \leftarrow \mathcal{S}_t(a) \cup (2^{-i}, [2^{2i+1} \log T], t)$.
 12: **Select an arm:** Round-robin sampling of "good" arms and "obligation" arms
 13: Select $a_t = \arg \min_a \{\tau : a \notin \{a_\tau, \dots, a_{t-1}\}, a \in \text{GOOD}_t \vee \mathcal{S}_t(a) \neq \{\}\}$.
 14: Receive reward r_t .
 15: **Check for changes of good arms:**
 16: If there is $a \in \text{GOOD}_t$ and $t_\ell \leq s_1 \leq s_2 \leq t$ and $t_\ell \leq s \leq t$ such that condition (3)
 17: holds, then start a new episode. (3): There is some subset of the episode over which a 's mean is
 18: significantly different than its current mean
 19: **Check for changes of bad arms:**
 20: If there is $a \in \text{BAD}_t$ and $t_\ell \leq s \leq t$ such that condition (4) holds,
 21: then start a new episode. (4): The bad arm's mean has improved by at least its
 22: suboptimality gap
 23: **Evict arms from GOOD_t:**
 24: $\text{BAD}_{t+1} = \text{BAD}_t \cup \{a \in \text{GOOD}_t \mid \exists s \geq t_\ell \text{ for which (1) holds}\}$.
 25: For evicted arms $a \in \text{BAD}_{t+1} \setminus \text{BAD}_t$, calculate $\bar{\mu}_\ell(a)$ and $\bar{\Delta}_\ell(a)$ according to (2), and
 26: set $\mathcal{S}_{t+1}(a) \leftarrow \{\}$.
 27: $\text{GOOD}_{t+1} = \{1, \dots, K\} \setminus \text{BAD}_{t+1}$.
 28: Continue with the next time step.

For $\epsilon = \Delta, 2\Delta, 4\Delta, \dots$
 with probability proportional to ϵ ,
 enqueue a "sampling obligation" lasting
 $\tilde{O}(\frac{1}{\epsilon^2})$ time steps.

Change-point detection

elimination




Figure 2: An annotated copy of the AdSwitch algorithm. AdSwitch is an elimination-style algorithm in which arms are sampled round-robin (line 13) until they are eliminated (lines 22-25). In contrast with a typical elimination algorithm, eliminated arms are occasionally tested to ensure that they have not changed (lines 10-11). Whenever an arm is sampled, changepoint detection is run to test whether the arm mean has changed (lines 15-21). When a changepoint is detected, all history is forgotten and arm means are reset.

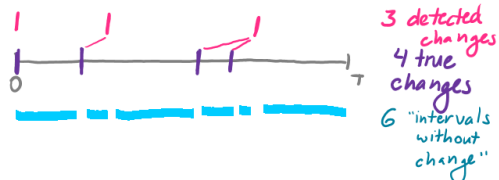


Figure 3: A sketch arguing that, as long as changepoint detection has no false positives, there are at most $2L$ intervals without change.