Stochastic Bandits with Delay-Dependent Payoffs

Leonardo Cella **Nicolò Cesa-Bianchi**
DSRC & Dept. of Computer Science, Università degli Studi di Milano, Italy

Abstract

Motivated by recommendation problems in music streaming platforms, we propose a nonstationary stochastic bandit model in which the expected reward of an arm depends on the number of rounds that have passed since the arm was last pulled. After proving that finding an optimal policy is NP-hard even when all model parameters are known, we introduce a class of ranking policies provably approximating, to within a constant factor, the expected reward of the optimal policy. We show an algorithm whose regret with respect to the best ranking policy is bounded by $\widetilde{\mathcal{O}}\big(\sqrt{kT}\big)$, where k is the number of arms and T is time. Our algorithm uses only $\mathcal{O}\big(k \ln \ln T\big)$ switches, which helps when switching between policies is costly. As constructing the class of learning policies requires ordering the arms according to their expectations, we also bound the number of pulls required to do so. Finally, we run experiments to compare our algorithm against UCB on different problem instances.

1 Introduction

Multiarmed bandits —see, e.g., (Bubeck and Cesa-Bianchi, 2012)— are a popular mathematical framework for modeling sequential decision problems in the presence of partial feedback; typical application domains include clinical trials, online advertising, and product recommendation. Consider for example the task of learning the genre of songs most liked by a given user of a music streaming platform. Each song genre is viewed as an arm of a bandit problem associated with the user. A bandit algorithm learns by sequentially choosing arms (i.e., recommending songs) and observing the resulting payoff (i.e., whether the user liked the song). The payoff is used by the algorithm to refine its recommendation policy. The distinctive feature of bandits is that, after each recommendation, the algorithm gets only a feedback for the selected arm (i.e., the single genre that was recommended).

In the simplest stochastic bandit framework (Lai and Robbins, 1985) rewards are realizations of i.i.d. draws from fixed and unknown distributions associated to each arm. In this setting the optimal policy is to consistently recommend the arm with the highest reward expectation. On the other hand, in scenarios like song recommendation, users may grow tired of listening to the same music genre over and over. This is naturally formalized as a nonstationary bandit setting, where the payoff of an arm grows with the time since the arm was last played. In this case policies consistently recommending the same arm are seldom optimal. E-learning applications, where arms corresponds to questions that students have to answer, are other natural examples of the same phenomenon, as asking again immediately the same question that the student has just answered is not very effective.

In this paper we introduce a simple nonstationary stochastic bandit model, B2DEP, in which the expected reward $\mu_i(\tau)$ of an arm i is a bounded nondecreasing function of the number τ of rounds that have passed since the arm was last selected by the policy. More specifically, we assume each arm i has an unknown baseline payoff expectation μ_i (equal to the expected payoff when the arm is pulled for the first time) and an unknown delay parameter $d_i > 0$. If the arm was pulled recently (that is, $1 \leq \tau \leq d_i$), then the expected payoff may be smaller that its baseline value: $\mu_i(\tau) \leq \mu_i$. Vice versa, if $\tau > d_i$, then $\mu_i(\tau)$ is guaranteed to match the baseline value μ_i. In the song recommendation example, the delays d_i model the extent to which listening to a song of genre i affects how much a user is willing to listen to more songs of that same genre.

Since τ can be viewed as a notion of state for arm i, our model can be compared to nonstationary models, such as rested bandits (Gittins, 1979) and restless bandits (Whittle, 1988) —see also (Tekin and Liu, 2012).

Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

In restless bandits the reward distribution of an arm changes irrespective of the policy being used, whereas in rested bandits the distribution changes only when the arm is selected by the policy. Our setting is neither rested nor restless, as our reward distributions change differently according to whether the arm is selected by the policy or not.

In Section 4 we make a reduction to the Periodic Maintenance Scheduling Problem (Bar-Noy et al., 2002) to prove that the optimization problem of finding an optimal periodic policy in our setting is NP-Hard. In order to circumvent the hardness of computing the optimal periodic policy, in Section 5 we identify a simple class of periodic policies that are efficiently learnable, and whose expected reward is provably to within a constant factor of that of the optimal policy. Our approximating class is pretty natural: it contains all *ranking policies* that cycle over the r best arms (where r is the parameter to optimize) according to the unknown ordering based on the arms' baseline payoff expectations. As it turns out, learning the best ranking policy can be formulated in terms of minimizing the standard notion of regret. This is unlike the problem of learning the best periodic policy, which instead requires minimizing the harder notion of policy regret (Arora et al., 2012).

Consider the task of learning the best ranking policy. In our music streaming example, a ranking policy is a playlist for the user. As changing the playlist streamed to the user may be costly in practice, we also introduce a *switching cost* for selecting a different ranking policy. Controlling the number of switches could also have a good effect in our nonstationary setting, when the expected reward of a ranking policy may depend on which other ranking polices were played earlier. The learning agent should ensure that a ranking policy is played many times consecutively (i.e., infrequent switches), so that estimates are calibrated (i.e., computed in the same context of past plays).

A standard bandit strategy like UCB (Auer et al., 2002), which guarantees a regret of $\mathcal{O}\big(\sqrt{kT\ln T}\big)$ irrespective of the size of the suboptimality gaps between the expected reward of the optimal ranking policy and that of the other policies, performs a number of switches growing with the squared inverse of these gaps. In Section 6 we show how to learn the best ranking policy using a simple variant of a learning algorithm based on action elimination proposed in (Cesa-Bianchi et al., 2013). Similarly to UCB, this algorithm has a distribution-free regret bounded by $\sqrt{kT}$. However, a bound $\mathcal{O}\big(k\ln\ln T\big)$ on the number of switches is also guaranteed irrespective of the size of the gaps.

In Section 7 we turn to the problem of constructing the class of ranking policies, which amounts to learning the ordering of the arms according to their baseline payoff expectations $\mu_1,\dots,\mu_k$. Assuming $\mu_1 > \cdots > \mu_k$, this can be reduced to the problem of learning the ordering of reward expectations in a standard stochastic bandit with i.i.d. rewards. We show that this is possible with a number of pulls bounded by $\sum_i 1/\Delta_i^2$ (ignoring logarithmic factors), where Δ_i is the smallest gap between $\mu_{i-1}-\mu_i$ and $\mu_i-\mu_{i-1}$. Note that this bound is not significantly improvable, because $1/\Delta_i^2$ samples of each arm i are needed to verify that $\mu_{i-1} < \mu_i < \mu_{i+1}$.

Finally, in Section 8 we describe experiments comparing our low-switch algorithm against UCB in both large-gap and in small-gap settings.

2 Related works

Our setting is a variant of the model introduced by Kleinberg and Immorlica (2018). In that work, $\mu_i(\tau)$ are concave, nondecreasing functions satisfying $\mu_i(\tau) \le \mu_i(\tau-1)+1$. Note that this setting and ours are incomparable. Indeed, unlike (Kleinberg and Immorlica, 2018) we assume a specific parametric form for the functions $\mu_i(\cdot)$, which are nondecreasing and bounded by 1. On the other hand, we do not assume concavity, which plays a key role in their analysis.

Pike-Burke and Grunewalder (2019) consider a setting in which the expected reward functions $\mu_i(\cdot)$ are sampled from a Gaussian Process with known kernel. The main result is a bound of order $\sqrt{kT}$ on the Bayesian d-step lookahead regret, where d is a user-defined parameter. This notion of regret is defined by dividing time in length-d blocks, and then summing the regret in each block against the greedy algorithm optimizing the next d pulls given the agent's current configuration of delays (i.e., how long ago each arm was last pulled). Similarly to (Pike-Burke and Grunewalder, 2019), we also compete against a greedy block strategy. However, in our case the block length is unknown, and the greedy strategy is not defined in terms of the agent's delay configuration.

A special case of our model is investigated in the very recent work by Basu et al. (2019). Unlike B2DEP, they assume $\mu_i(\tau) = 0$ for all $\tau \le d_i$ and complete knowledge of the delays d_i. In fact, they even assume that every arm i cannot be selected in the next d_i time steps after a pull. Their main result is a regret bound for a variant of UCB competing against the greedy policy. They also show NP-hardness of finding the optimal policy through a reduction similar to ours. It is not clear how their learning approach could be extended to prove results in our more general setting, where $\mu_i(\tau)$ could be positive even when $\tau \le d_i$ and the delays d_i are unknown.

A different approach to nonstationary bandits in recommender systems considers expected reward functions that depend on the number of times the arm was played so far, (Levine et al., 2017; Cortes et al., 2017; Bouneffouf and Féraud, 2016; Heidari et al., 2016; Seznec et al., 2019; Warlop et al., 2018). These cases correspond to a rested bandit model, where each arm's expected reward can only change when the arm is played.

The fact that we learn ranking strategies is reminiscent of stochastic combinatorial semi-bandits (Kveton et al., 2015b), where the number of arms in the schedule is a parameter of the learning problem. In particular, similarly to (Radlinski et al., 2008; Kveton et al., 2015a; Katariya et al., 2016) our strategies learn rankings of the actions, but unlike those approaches in our case the optimal number of elements in the ranking must be learned too.

3 The B2DEP setting

In the classical stochastic multiarmed bandit model, at each round $t = 1, 2, \dots$ the agent pulls an arm from $\mathcal{K} = \{1, \dots, k\}$ and receives the associated payoff, which is a $[0, 1]$-valued random variable independently drawn from the (fixed but unknown) probability distribution associated with the pulled arm. The payoff is the only feedback revealed to the agent at each round. The agent's goal is to maximize the expected cumulative payoff over any number of rounds.

In the B2DEP (Bandits with DElay DEpendend Payoff) variant introduced here, when the agent plays an arm $i \in \mathcal{K}$ the $[0, 1]$-valued payoff has expected value

$$\mu_i(\tau) = \left(1 - f(\tau)\mathbb{I}\left\{0 < \tau \leq d_i\right\}\right)\mu_i \tag{1}$$

where μ_i is the unknown *baseline reward expectation* for arm i, $f : \mathbb{N} \to [0, 1]$ is an unknown nonincreasing function, and τ is the number of rounds that have passed since that arm was last pulled (conventionally, $\tau = 0$ means that an arm is pulled for the first time). When f is identically zero, B2DEP reduces to the standard stochastic bandit model with payoff expectations $\mu_1, \dots, \mu_k$. The unknown arm-dependent delay parameters $d_i > 0$ control the number of rounds after which the arm's expected payoff is guaranteed to return to its baseline value μ_i.

A policy π maps a sequence of past observed payoffs to the index of the next arm to pull. Let $g_t(\pi)$ be the payoff collected by policy π at round t. Given an instance of B2DEP, the optimal policy π^* maximizes, over all policies π, the long term expected average payoff

$$\lim_{T\to\infty} \frac{G_T(\pi)}{T} \quad \text{where} \quad G_T(\pi) = \mathbb{E}\left[\sum_{t=1}^{T} g_t(\pi)\right].$$

Note that, the payoff expectations at any time step t are fully determined by the current *delay vector* $\boldsymbol{\tau}(t) = \big(\boldsymbol{\tau}_1(t), \dots, \boldsymbol{\tau}_k(t)\big)$, where each integer $0 \leq \tau_i(t) \leq d_i$ counts how many rounds have passed since $i \in \mathcal{K}$ was last pulled (setting $\tau_i(t) = 0$ if i was never pulled or if it was last pulled more than d_i steps ago). Hence, any delay-based policy —e.g., any deterministic function of the current delay vector— is eventually periodic, meaning that $\pi\big(\boldsymbol{\tau}(t)\big) = \pi\big(\boldsymbol{\tau}(t + P)\big)$ for all $t_0 \leq t \leq T$, where P is the period and t_0 is the length of the transient.

Consider the greedy policy π_{greedy} defined as follows: At each round t, π_{greedy} pulls the arm $i \in \mathcal{K}$ with the highest expected reward according to current delays

$$\pi_{\text{greedy}}\big(\boldsymbol{\tau}(t)\big) = \arg\max_{i\in\mathcal{K}} \mu_i\big(\tau_i(t)\big) \tag{2}$$

where $\tau_i(t) = 0$ if i was never pulled before. It is easy to see that π_{greedy} is not always optimal. For example consider the following instance of $B2DEP$ with $k = 2$: $f(\tau) = \frac{1}{2}$ for all τ, $\mu_1 = 1$, $\mu_2 = \frac{1}{2} - \varepsilon$, $d_1 = d_2 = 1$. Then π_{greedy} always pulls arm 1 and achieves $G_t(\pi_{\text{greedy}}) = 1 + \frac{T-1}{2}$, whereas $G_T(\pi^*) = 1 + \frac{T-1}{2}\big(\frac{3}{2} - \varepsilon\big)$ where π^* alternates between arm 1 and arm 2. Hence $G_T(\pi_{\text{greedy}}) \leq \frac{2}{3} G_T(\pi^*)$.

In the next section we show that the problem of finding the optimal periodic policy for B2DEP is intractable.

4 Hardness results

We show that the optimization problem of finding an optimal policy for B2DEP is NP-hard, even when all the instance parameters are known. Our proof relies on the NP-completeness of the Periodic Maintenance Scheduling Problem (PMSP) shown by Bar-Noy et al. (2002). Although a very similar result can also be proven using the reduction of Basu et al. (2019), introduced for a special case of our B2DEP setting, we give our proof for completeness.

A maintenance schedule on n machines $\{1, \dots, n\}$ is any infinite sequence over $\{0, 1, \dots, n\}$, where 0 indicates that no machine is scheduled for service at that time. An instance of the PMSP decision problem is given by integer service intervals $\ell_1, \dots, \ell_n$ such that $\sum_{i=1}^{n} \frac{1}{\ell_i} \leq 1$. The question is whether there exists a maintenance schedule such that the consecutive service times of each machine i are exactly ℓ_i times apart. The following result holds (proof in the supplementary material).

Theorem 1. *It is NP-hard to decide whether an instance of B2DEP has a periodic policy π achieving*

$$\lim_{T\to\infty} \frac{G_T(\pi)}{T} \geq \sum_{i=1}^{k} \frac{\mu_i}{d_i + 1}\ .$$

5 Approximating the optimal policy

In order circumvent the computational problem of finding the best periodic policy, we introduce a simple class $\Pi_{\mathcal{K}}$ of periodic *ranking policies* whose best element π_{ghost} has a cumulative expected payoff not too far from that of π^*. Without loss of generality, assume that $\mu_1 > \cdots > \mu_k$. Let $\Pi_{\mathcal{K}} \equiv \{\pi_m : m \in \mathcal{K}\}$, where each policy π_m cycles over the arm sequence $1, \dots, m$. The average reward $g(m)$ of policy π_m is defined by

$$g(m) = \frac{1}{m} \sum_{j=1}^{m} \mu_j(m) \ .$$

Since π_{ghost} maximizes $g(m)$ over $m \in \mathcal{K}$, $\pi_{\text{ghost}} \equiv \pi_{r^\star}$ where

$$r^\star \in \arg\max_{m=1,\dots,k} \frac{1}{m} \sum_{j=1}^{m} \mu_j(m) \tag{3}$$

We now bound $G_T(\pi_{\text{ghost}})$ in terms of $G_T(\pi^*)$.

Theorem 2.

$$G_T(\pi_{\text{ghost}}) \geq \big(1 - f(r_0)\big) G_T(\pi^*) + O(1)$$

where r_0 is the largest arm index r such that

$$\mu_i > \max_{j=1,\dots,i-1} \mu_j(i-j) \quad i = 2, \dots, r$$

and $r_0 = 1$ if $\mu_2 \leq \mu_1(1)$.

The definition of r_0 is better understood in the context of the more intuitive delay-based policy π_{greedy}. Note indeed that $r_0 + 1$ is the first round in which π_{greedy} prefers to pull one of the arms that were played in the first r_0 rounds rather than the next arm $r_0 + 1$.

Proof. Since $r^\star$ maximizes (3),

$$\begin{aligned} G_T(\pi_{\text{ghost}}) &= \frac{T}{r^\star} \sum_{i=1}^{r^\star} \mu_i(r^\star) + \mathcal{O}(1) \\ &\geq \frac{T}{r_0} \sum_{i=1}^{r_0} \mu_i(r_0) + \mathcal{O}(1) \\ &\geq \frac{T}{r_0} \sum_{i=1}^{r_0} \big(1 - f(r_0)\big) \mu_i + \mathcal{O}(1) \end{aligned}$$

where the $\mathcal{O}(1)$ term takes into account that $r^\star$ may not divide T, and the fact that in the first $r^\star$ rounds the expected reward is $\mu_1 + \cdots + \mu_{r^\star}$ instead of $\mu_1(r^\star) + \cdots + \mu_{r^\star}(r^\star)$. Now split the T time steps in blocks of length r_0. Because r_0 is —by definition— the largest expected reward any policy can achieve in r_0 consecutive steps, the expected reward of π^* in any of these blocks is at most $\mu_1 + \cdots + \mu_{r_0}$. Therefore

$$G_T(\pi^*) \leq \frac{T}{r_0} \sum_{i=1}^{r_0} \mu_i + \mathcal{O}(1)$$

where, as before, the $\mathcal{O}(1)$ term takes into account that r_0 may not divide T. This concludes the proof. □

The proof of Theorem 2 actually shows that both $r^\star$ and r_0 achieve the claimed approximation. However, by definition $G_T(\pi_{\text{ghost}})$ is bigger than the total reward of the policy that cycles over $1, \dots, r_0$. Also, learning π_{ghost} is relatively easy, as we show in Section 6.

It is easy to see that $g(m)$ is not monotone due to the presence of the coefficients d_i. For example, consider the B2DEP instance defined by $k = 3$, $\mu_1 = 1$, $\mu_2 = \frac{2}{3}$, $\mu_3 = \frac{1}{2}$, $d_1 = d_2 = d_3 = 2$, and $f(\tau) = 2^{-\tau}$. Then $g(2) < g(1) < g(3)$.

6 Learning the ghost policy

In this section we deal with the problem of learning $r^\star$ assuming the correct ordering $1, \dots, k$ of the arms (such that $\mu_1 > \cdots > \mu_k$) is known. In the next section, we consider the problem of learning this ordering.

Our search space is the set of ranking policies $\Pi_{\mathcal{K}} \equiv \{\pi_m : m \in \mathcal{K}\}$, where each policy π_m cycles over the arm sequence $1, \dots, m$. Note that, by definition, $\pi_{\text{ghost}} \equiv \pi_{r^\star}$. The average reward $g(m)$ of policy π_m is defined by $g(m) = \big(\mu_1(m) + \cdots + \mu_m(m)\big)/m$. Note that every time the learning algorithm chooses to play a different policy $\pi_m \in \Pi_{\mathcal{K}}$, an extra cost is incurred due to the need of calibrating the estimates for $g(m)$. In fact, if we played a policy different from π_m in the previous round, the reward expectation associated with the play of π_m in the current round is potentially different from $g(m)$. This is due to the fact that we cannot guarantee that each arm in the schedule used by π_m was pulled exactly m steps earlier. This implies that we need to play each newly selected policy more than once, as the first play cannot be used to reliably estimate $g(m)$.

We now introduce the policy π_{low} (Algorithm 1), a simple variant of a learning algorithm based on action elimination proposed in (Cesa-Bianchi et al., 2013). This policy has a regret bound similar to UCB while guaranteeing a bound $\mathcal{O}\big(k \ln \ln T\big)$ on the number of switches, irrespective of the size of the gaps. In Section 8 we compare $\pi_{\text{low}}$with UCB.

In each stage s, algorithm π_{low} plays each policy π_m in the active set $\mathcal{A}_s$ for $T_s/\big(m|\mathcal{A}_s|\big) + 1$ times, where $T_s = T^{1-2^{-s}}$. Then, the algorithm computes the sample average reward $\widehat{g}_s(m)$ based on these plays, excluding the first one because of calibration (lines 3–6). After that, the empirically best policy is selected (7). Finally, the active set is recomputed (line 8) excluding all policies whose sample average reward is significantly smaller than that of the empirically best

Algorithm 1 (π_{low})

Input: Policy set $\Pi_{\mathcal{K}}$, confidence $\delta \in (0,1)$, horizon T

Let $\mathcal{A}_1 \equiv \mathcal{K}$ be the initial set of active policies
repeat ▷ s indexes the stage number
  for $m \in \mathcal{A}_s$ **do**
    Play π_m for $T_s/\big(m|\mathcal{A}_s|\big) + 1$ times
    Compute $\widehat{g}_s(m)$ discarding the first play
  end for
  Let $\widehat{m}_s = \arg\max_{m \in \mathcal{A}_s} \widehat{g}_s(\widehat{m}_s)$
  $\mathcal{A}_{s+1} = \{m \in \mathcal{A}_s \,:\, \widehat{g}_s(m) \geq \widehat{g}_s(\widehat{m}_s) - 2C_s\}$
until overall number of pulls exceeds T

policy. The quantity C_s is derived from a standard Chernoff-Hoeffding bound and is equal to $\sqrt{\frac{k}{2T_s} \ln \frac{2kS}{\delta}}$ where

$$S = \min \left\{ j \in \mathbb{N} \,:\, \sum_{s=1}^{j} \big(|\mathcal{A}_s| + T_s\big) \geq T \right\}$$

implying $S = \mathcal{O}\big(\ln \ln T\big)$. The terms $|\mathcal{A}_s|$ account for the extra calibration pull each time we switch to a new policy in $\Pi_{\mathcal{K}}$. We can prove the following bound on the regret of π_{low} with respect to π_{ghost}.

Theorem 3. *When run on an instance of B2DEP with parameters δ and T, with probability at least $1-\delta$ Algorithm 1 guarantees*

$$G_T(\pi_{\text{ghost}}) - G_T(\pi_{\text{low}}) = \mathcal{O}\left(k^2 \ln \ln T + \sqrt{kT\left(\ln \frac{k}{\delta} + \ln \ln \ln T\right)}\right) \quad (4)$$

with probability at least $1-\delta$.

Note that this bound is distribution-free. That is, it does not depend on the gaps $g(r^\star) - g(m)$ (which in general could be arbitrarily small). The rate $\sqrt{T}$, as opposed to the $\ln T$ rate of distribution-dependent bounds, cannot be improved upon in general Bubeck and Cesa-Bianchi (2012).

Proof. The proof is an adaptation of (Cesa-Bianchi et al., 2013, Theorem 6). Note that $\mathcal{A}_S \subseteq \cdots \subseteq \mathcal{A}_1$ by construction. Also, our choice of C_s and Chernoff-Hoeffding bound implies that

$$\max_{m \in \mathcal{A}_s} \big|\widehat{g}_s(m) - g(m)\big| \leq C_s \quad (5)$$

simultaneously for all $s = 1, \ldots, S$ with probability at least $1-\delta$. To see this, note that in every stage s the estimates $\widehat{g}_s(m)$ are computed using $T_s/\big(m|\mathcal{A}_s|\big)$ plays. Since a play of π_m consists of $m \leq k$ pulls, we have that each $g(m)$ is estimated using $T_s/|\mathcal{A}_s| \geq T_s/k$ realizations of a sequence of random variables whose expectations have average exactly equal to $g(m)$.

We now claim that, with probability at least $1-\delta$, $r^\star \in \bigcap_{s=1}^{S} A_s$ and $0 \leq \widehat{g}_s(\widehat{m}_s) - \widehat{g}_s(r^\star) \leq 2C_s$ for all $s = 1, \ldots, S$.

We prove the claim by induction on $s = 1, \ldots, S$. We first show that the base case $s = 1$ holds with probability at least $1 - \delta/S$. Then we show that if the claim holds for $s-1$, then it holds for s with probability at least $1 - \delta/S$ over all random events in stage s. Therefore, using a union bound over $s = 1, \ldots, S$ we get that the claim holds simultaneously for all s with probability at least $1-\delta$.

For the base case $s = 1$ note that $r^\star \in A_1$ by definition, and thus $0 \leq \widehat{g}_1(\widehat{m}_1) - \widehat{g}_1(r^\star)$ holds. Moreover: $\widehat{g}_1(\widehat{m}_1) - g(\widehat{m}_1) \leq C_1$, $g(r^\star) - \widehat{g}_1(r^\star) \leq C_1$, and $g(\widehat{m}_1) - g(r^\star) \leq 0$, where the two first inequalities hold with probability at least $1-\delta$ because of (5). This implies $0 \leq \widehat{g}_1(\widehat{m}_1) - \widehat{g}_1(r^\star) \leq 2C_1$ as required. We now prove the claim for $s > 1$. The inductive assumption

$$r^\star \in \mathcal{A}_{s-1} \text{ and } 0 \leq \widehat{g}_{s-1}(\widehat{m}_{s-1}) - \widehat{g}_{s-1}(r^\star) \leq 2C_{s-1}$$

directly implies that $r^\star \in \mathcal{A}_s$. Thus we have $0 \leq \widehat{g}_s(\widehat{m}_s) - \widehat{g}_s(r^\star)$, because $\widehat{m}_s$ maximizes $\widehat{g}_s$ over a set that contains $r^\star$. The rest of the proof of the claim closely follows that of the base case $s = 1$.

We now return to the proof of the theorem. For any $s = 1, \ldots, S$ and for any $m \in \mathcal{A}_s$ we have that

$$\begin{aligned}
g(r^\star) - g(m) &\leq g(r^\star) - \widehat{g}_{s-1}(m) + C_{s-1} \quad \text{by (5)} \\
&\leq g(r^\star) - \widehat{g}_{s-1}(\widehat{m}_{s-1}) + 3C_{s-1} \\
&\quad \text{by definition of } \mathcal{A}_{s-1}, \text{ since } m \in \mathcal{A}_s \subseteq \mathcal{A}_{s-1} \\
&\leq g(r^\star) - \widehat{g}_{s-1}(r^\star) + 3C_{s-1} \\
&\quad \text{since } \widehat{m}_{s-1} \text{ maximizes } \widehat{g}_{s-1} \text{ in } \mathcal{A}_{s-1} \\
&\leq 4C_{s-1} \quad \text{by (5)}
\end{aligned}$$

holds with probability at least $1 - \delta/S$. Hence, recalling that the number of switches between two different policies in $\Pi_{\mathcal{K}}$ is deterministically bounded by kS, the regret of the player can be bounded as follows,

$$\begin{aligned}
&G_T(\pi_{\text{ghost}}) - G_T(\pi_{\text{low}}) \\
&\quad = k^2 S + \sum_{s=1}^{S} \frac{T_s}{|\mathcal{A}_s|} \sum_{m \in \mathcal{A}_s} \Big(g(r^\star) - g(m)\Big) \\
&\quad = k^2 S + T_1 + \sum_{s=2}^{S} \frac{T_s}{|\mathcal{A}_s|} \sum_{m \in \mathcal{A}_s} \Big(g(r^\star) - g(m)\Big) \\
&\quad \leq k^2 S + T_1 \; + \sum_{i=2}^{S} 4T_s \sqrt{\frac{k}{2T_{s-1}} \ln \frac{2kS}{\delta}} \\
&\quad = k^2 S + T_1 + 4\sqrt{k \ln \frac{2kS}{\delta}} \sum_{s=2}^{S} \frac{T_s}{\sqrt{T_{s-1}}}
\end{aligned}$$

Algorithm 2 (`BanditRanker`)

Input: Confidence $\delta \in (0,1)$
Output: A permutation $[1],\dots,[k]$ of $\mathcal{K}$.
1: Let $\mathcal{A}_1 \equiv \mathcal{K}$ be the initial set of active arms
2: **repeat** ▷ r indexes the round number
3: Sample once all arms in $\mathcal{A}_r$ ▷ sampling round
4: Sort the empirical means $\widehat{\mu}_{[1],r} \geq \cdots \geq \widehat{\mu}_{[n],r}$
5: **for** $i = 1$ to $|\mathcal{A}|$ **do**
6: **if** $\widehat{\mu}_{[i],r} + 2\varepsilon_r < \min_{j \in \mathcal{K}^+_{[i],r}} \widehat{\mu}_{j,r}$ **then**
7: **if** $\widehat{\mu}_{[i],r} - 2\varepsilon_r > \max_{j \in \mathcal{K}^-_{[i],r}} \widehat{\mu}_{[s],r}$ **then**
8: Remove $[i]$ from $\mathcal{A}_r$
9: Rank before $[i]$ all arms in $\mathcal{K}^+_{[i],r}$
10: Rank after $[i]$ all arms in $\mathcal{K}^-_{[i],r}$
11: **end if**
12: **end if**
13: **end for**
14: **until** $|\mathcal{A}_t| \leq 1$

where the $k^2 S$ term accounts for the regret suffered in the kS plays where we switched between two policies in $\Pi_{\mathcal{K}}$ and paid maximum regret due to calibration for at most k steps (as each policy in $\Pi_{\mathcal{K}}$ is implemented with at most k pulls). Now, since $T_1 = \sqrt{T}$, $T_s/\sqrt{T_{s-1}} = \sqrt{T}$ and $S = \mathcal{O}\big(\ln \ln T\big)$, we obtain that with probability at least $1-\delta$ the regret is at most of order $k^2 \ln \ln T + \sqrt{T} + \sqrt{kT\left(\ln \frac{k}{\delta} + \ln \ln \ln T\right)}$. □

7 Learning the ordering of the arms

In this section we show how to recover, with high probability, the correct ordering $\mu_1 > \cdots > \mu_k$ of the arms. Initially, we ignore the problem of calibration, and focus on the task of learning the arm ordering when each pulls of arm i returns a sample from the true baseline reward distribution with expectation μ_i.

`BanditRanker` (Algorithm 2) is an action elimination procedure. The arms in the set $\mathcal{A}_r$ of active arms are sampled once each (line 3), and their average rewards are kept sorted in decreasing order (line 4). We use $\widehat{\mu}_{i,r}$ to denote the sample average of rewards obtained from arm i after r sampling rounds, and define the indexing $[1],\dots,[k]$ be such that $\widehat{\mu}_{[1],r} \geq \cdots \geq \widehat{\mu}_{[k],r}$, where ties are broken according to the original arm indexing.

When the confidence interval around the average reward of an arm $[i]$ is not overlapping anymore with the confidence intervals of the other arms (lines 6–7), $[i]$ is removed from $\mathcal{A}_r$ and not sampled anymore (line 8). Moreover, the set $\mathcal{K}^+_{[i],r}$ of all arms $[b] \in \mathcal{A}_r$ such that $\widehat{\mu}_{[b],r} \geq \widehat{\mu}_{[i],r}$ (if any) is ranked before $[i]$ (line 9). Similarly, the set let $\mathcal{K}^-_{[i],r}$ of all arms $[s] \in \mathcal{A}_r$ such that $\widehat{\mu}_{[s],r} \leq \widehat{\mu}_{[i],r}$ (if any) is ranked after $[i]$ (line 10). The algorithm ends when all arms are removed (line 14).

The parameter ε_r determining the confidence interval after r sampling rounds is defined by

$$\varepsilon_r = \sqrt{\frac{1}{2r} \ln \frac{2kr(r+1)}{\delta}} \ . \tag{6}$$

The sequence of removed arms can be stored in a binary tree whose root is the first removed arm and whose left (resp., right) leaf contain all arms whose average reward was bigger (resp., smaller) when the first arm was removed. When a new arm is removed, the leaf to which it belongs is split using the same logic that we used for the root. Eventually, all nodes contain a single arm and the in-order traversal of the tree provides the desired ordering.

We introduce the following quantity, measuring the suboptimality gaps between arm that are adjacent in the correct ordering,

$$\Delta_i = \begin{cases} \Delta_{1,2} & \text{if } i = 1 \\ \min\big\{\Delta_{i-1,i}, \Delta_{i,i+1}\big\} & \text{if } 1 < i < k \\ \Delta_{k-1,k} & \text{if } i = k \end{cases}$$

where $\Delta_{i,j} = \mu_i - \mu_j$.

We are now ready to state and prove the main result of this section.

Theorem 4. *If Algorithm 2 is run with parameter δ on a k-armed stochastic bandit problem, the correct ordering $\mu_1 > \cdots > \mu_k$ of the arms is returned with probability at least $1-\delta$ after a number of pulls of order*

$$\sum_{i=1}^{k-1} \frac{1}{\Delta_i^2} \ln \frac{1}{\delta \Delta_i} \ . \tag{7}$$

Note that, up to logarithmic factors, the bound stated in Theorem 4 is of the same order as the sample used by an ideal procedure that knows $\Delta_1,\dots,\Delta_k$ and uses the optimal order $1/\Delta_i^2$ of samples to determine the position of each arm i in the correct ordering.

Proof. The proof is an adaptation of Even-Dar et al. (2006, Theorem 8). Using Chernoff-Hoeffding bounds, the choice of ε_r ensures that

$$\begin{aligned} \mathbb{P}\Big(\exists\, r \geq 1\ \exists i \in \mathcal{K}\ \big|\widehat{\mu}_{i,r} - \mu_i\big| > \varepsilon_r\Big) &\leq 2k \sum_{r \geq 1} e^{-2\varepsilon_t^2 r} \\ &\leq \delta \ . \end{aligned} \tag{8}$$

If an action $[i]$ is eliminated after r sampling rounds, then it must be that $\widehat{\mu}_{[b],r} - 2\varepsilon_r > \widehat{\mu}_{[i],r} > \widehat{\mu}_{[s],r} + 2\varepsilon_r$ for all $[b] \in \mathcal{K}^+_{[i],r}$ and all $[s] \in \mathcal{K}^-_{[i],r}$. Condition (8) then ensures that, with probability at least $1-\delta$, $\mu_{[b]} >$

$\mu_{[i]} > \mu_{[s]}$ for all such b and s. This implies that the current ordering of $\mu_{[j],r}$ for $j \in \mathcal{A}_r$ is correct with respect to $[i]$. Since $\varepsilon_r \to 0$, every action is eventually eliminated. Therefore, with probability at least $1-\delta$ the sequence of eliminated arms i and their corresponding sets $\mathcal{K}^+_{[i],r}, \mathcal{K}^-_{[i],r}$ provide the correct arm ordering.

We now proceed to bounding the number of samples. Under condition (8), for all $b < i < s$,

$$\Delta_{b,i} - 2\varepsilon_r = \big(\mu_b - \varepsilon_r\big) - \big(\mu_i + \varepsilon_r\big) \le \widehat{\mu}_{b,r} - \widehat{\mu}_{i,r} \ .$$

Therefore, if $\widehat{\mu}_{b,r} - \widehat{\mu}_{i,r} \le 2\varepsilon_r$, then $\Delta_{b,i} \le 4\varepsilon_r$. Recalling the definition (6) of ε_r and solving by $r = r(b,i)$ we get

$$r(b,i) = \mathcal{O}\left(\frac{1}{\Delta^2_{b,i}} \ln \frac{1}{\delta \Delta_{b,i}}\right) \ .$$

Thus, after $r(b,i)$ sampling rounds, $\widehat{\mu}_{b,r(b,i)} - \widehat{\mu}_{i,r(b,i)} > 2\varepsilon_{r(b,i)}$ with probability at least $1-\delta$. Similarly, after $r(i,s)$ sampling rounds, $\widehat{\mu}_{i,r(i,s)} - \widehat{\mu}_{s,r(i,s)} > 2\varepsilon_{r(i,s)}$ with probability at least $1-\delta$.

This further implies that after $N_i = \mathcal{O}\left(\frac{1}{\Delta_i^2} \ln \frac{1}{\delta \Delta_i}\right)$ many sampling rounds, action i is eliminated and not sampled any more.

Re-define the indexing $[1], \dots, [k]$ so that $\Delta_{[1]} > \cdots > \Delta_{[k]}$. Hence $N_{[1]} < \cdots < N_{[k]}$ by definition. We now compute a bound on the overall number of pulls based on our bound on the number of sampling rounds. With probability at least $1-\delta$, we have that: $kN_{[1]}$ pulls are needed to eliminate arm $[1]$, $(k-1)\big(N_{[2]} - N_{[1]}\big)$ pulls are needed to eliminate arm $[2]$, and so on. Hence, with probability at least $1-\delta$ the total number of pulls needed to eliminate all arms is

$$\begin{aligned}
&\sum_{i=0}^{k-2} (k-i)\big(N_{[i+1]} - N_{[i]}\big) \\
&\quad = kN_{[k-1]} - \sum_{i=0}^{k-2} i\big(N_{[i+1]} - N_{[i]}\big) \\
&\quad = kN_{[k-1]} - (k-1)N_{[k-1]} + \sum_{i=0}^{k-2} N_{[i+1]} \\
&\quad = N_{[k-1]} + \sum_{i=1}^{k-1} N_{[i]}
\end{aligned}$$

where we set conventionally $N_{[0]} = 0$. This concludes the proof of the theorem. □

In order to apply `BanditRanker` to an instance of B2DEP, we assume that an upper bound $d_0 > \max_i d_i$ be available in advance to the algorithm. This ensures that $\mu_i(d_0) = \mu_i$ for all $i \in \mathcal{K}$. In each sampling round r, we partition the arms in $\mathcal{A}_r$ in groups of size d_0 and make $2d_0$ pulls for each group by cycling twice over the arms in an arbitrary order. Then, the first d_0 pulls in each group are discarded, while the last d_0 pulls are used to estimate the expectations μ_i (when d_0 does not divide $|\mathcal{A}_r|$ we can add to $\mathcal{A}_r$ arms that were already removed, or arms from previous groups, just for the purpose of calibrating). The sample size bound (7) remains of the same order (because the extra pulls only add a factor of two).

8 Experiments

In this section we present an empirical evaluation of our policy π_{low} in a synthetic environment with Bernoulli rewards. In order to study the impact the switching cost on ranking policies when the suboptimality gap is small, we also define a setting in which there are two distinct ranking policies that are both optimal —see Figure 1.

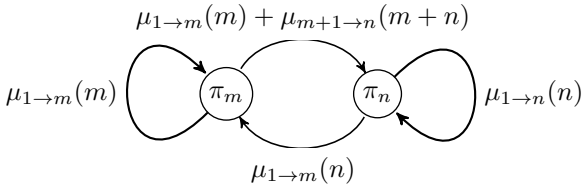

Figure 1: Transitions between policies π_m and π_n assuming $n > m$, where the notation $\mu_{m\to n}(d)$ stands for $\mu_m(d) + \cdots + \mu_n(d)$. The expected reward obtaining by switching between policies is different from the expected reward obtaining by cycling over the same policy.

We plot regrets against the policy π_{ghost}. Our policy π_{low} is run without any specific tuning (other than the knowledge of the horizon T) and with δ set to 0.1 in all experiments. The benchmark π_{ucb} consists of running UCB1 —with the same scaling factor as in the original article by Auer et al. (2002)— over the class $\Pi_{\mathcal{K}}$ of ranking policies, where calibration is addressed by rolling out twice each ranking policy selected by UCB1 and using only the second roll-out to compute reward estimates. Since both π_{low} and π_{ghost} are run over $\Pi_{\mathcal{K}}$, we implicitly assume that `BanditRanker` successfully ranked the arms in a preliminary stage.

Figure 2 shows that when the gap between the best and the second best ranking policy is not too small (0.1 on average in these experiments), then π_{ucb} is competitive against π_{low} even in the presence of unit switching costs. This happens because, in order to minimize the number of switches, π_{low} samples a suboptimal policy more frequently than π_{ucb}. Although this oversampling

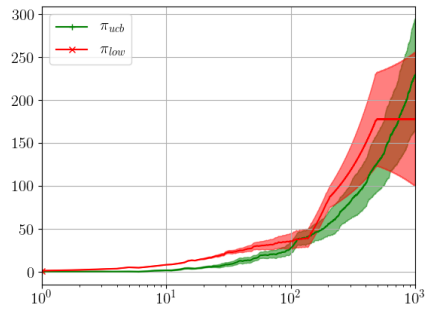

Figure 2: Comparing regrets of π_{low} and π_{ucb} against π_{ghost} with 7 arms and baseline expectations $0, 1/3, 2/3, 4/5, 13/15, 14/15, 1$ and $f(\tau) = (0.999)^{\tau}$. A unit cost is charged for switching between ranking policies. Curves are averages of 5 runs each using a different sample of delays $d_1, \ldots, d_7$ uniformly drawn from $\{1, \ldots, 6\}$. We plot expectations of sampled arms rather than realized rewards.

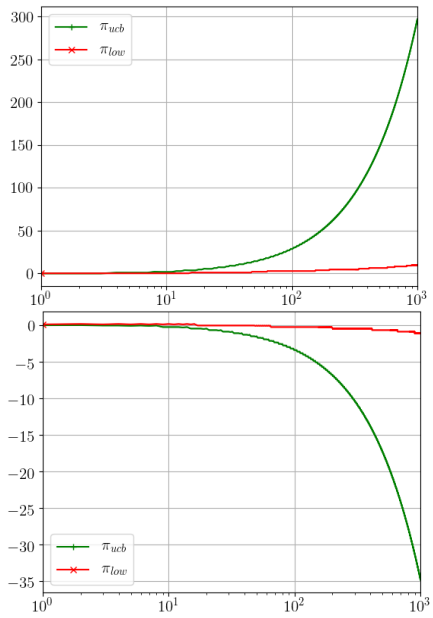

Figure 3: Comparing regrets of π_{low} and π_{ucb} against π_{ghost} with 2 arms such that $g(1) = g(2)$ with unit cost charged for switching between the two policies (upper part) and without any cost for switching (lower part).

does not affect the distribution-free regret bound of π_{low}, it hurts performance unless the suboptimality gap is small enough to cause the switching costs to prevail, a case which is addressed next. Note also that π_{low} eventually stops exploration because all policies but one have been eliminated, while π_{ucb} keeps on exploring, albeit at a logarithmic rate.

In the second experiment we consider two arms with $\mu_1 = 1$, $f(1) = 0.3$, $f(2) = 0.25$, $d_1 = d_2 = 2$, and μ_2 chosen so that $g(1) = g(2)$ to simulate a vanishing suboptimality gap between π_1 and π_2. Figure 3 (upper part) shows that π_{low} performs better than π_{ucb} due to its low switch regime. On the other hand, Figure 3 (lower part) shows that when the switching cost is zero, switching between two good policies becomes more advantageous than using a single good policy, and the regret of both π_{ucb} and π_{low} becomes negative (in this case π_{ucb}, which has no control over the number of switches, outperforms π_{low}). The reason for this advantage is explained by Fact 1 below (proof in the supplementary material), see also Figure 1.

Fact 1. *If an instance of B2DEP admits two optimal ranking policies, then consistently switching between these two policies achieves an average expected reward higher than sticking to either one.*

To summarize, the experiments confirm that, in the presence of switching costs, π_{low} works better than π_{ucb} only when the suboptimalty gap is very small. The advantage of π_{low} over π_{ucb} is however reduced by the fact that switching between two good policies is better than consistently playing either one of the two (Fact 1). Note also that π_{low} stops exploring because T is known. This preliminary knowledge can be dispensed with using a doubling trick, or some more sophisticated method. Also, it would be interesting to design a method that achieves the best between the performance of π_{ucb} and π_{low}, according to the size of the suboptimality gap.

9 Conclusions

Motivated by music recommendation in streaming platforms, we introduced a new stochastic bandit model with nonstationary reward distributions. To cope with the NP-hardness of learning the optimal policy caused by nonstationarity, we introduced a restricted class of ranking policies approximating the optimal performance. We then proved sample and regret bounds on the problem of learning the best ranking policy in this class. One of the main problems left open by our work is that of deriving more practical learning algorithms, able to simultaneously learn the ranking of the arms and the best cutoff value $r^{\star}$, while minimizing their regret with respect to the best ranking policy.

Acknowledgments

Nicolò Cesa-Bianchi acknowledges partial support by the Google Focused Award *Algorithms and Learning for AI* (ALL4AI) and by the MIUR PRIN grant *Algorithms, Games, and Digital Markets* (ALGADIMAR).

References

Raman Arora, Ofer Dekel, and Ambuj Tewari. Online bandit learning against an adaptive adversary: from regret to policy regret. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1747–1754. Omnipress, 2012.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3), 2002.

Amotz Bar-Noy, Randeep Bhatia, Joseph (Seffi) Naor, and Baruch Schieber. Minimizing service and operation costs of periodic scheduling. *Mathematics of Operations Research*, 27:518–544, 2002.

Soumya Basu, Rajat Sen, Sujay Sanghavi, and Sanjay Shakkottai. Blocking bandits. In *Advances in Neural Information Processing Systems*, pages 4785–4794, 2019.

Djallel Bouneffouf and Raphael Féraud. Multi-armed bandit problem with known trend. *Neurocomputing*, 205:16–21, 2016.

Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

Nicolo Cesa-Bianchi, Ofer Dekel, and Ohad Shamir. Online learning with switching costs and other adaptive adversaries. In *Advances in Neural Information Processing Systems*, pages 1160–1168, 2013.

Corinna Cortes, Giulia DeSalvo, Vitaly Kuznetsov, Mehryar Mohri, and Scott Yang. Discrepancy-based algorithms for non-stationary rested bandits. *arXiv preprint arXiv:1710.10657*, 2017.

Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *J. Mach. Learn. Res.*, 7:1079–1105, December 2006.

John C Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society: Series B (Methodological)*, 41(2):148–164, 1979.

Hoda Heidari, Michael Kearns, and Aaron Roth. Tight policy regret bounds for improving and decaying bandits. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1562–1570. AAAI Press, 2016.

Sumeet Katariya, Branislav Kveton, Csaba Szepesvari, and Zheng Wen. DCM bandits: Learning to rank with multiple clicks. In *International Conference on Machine Learning*, pages 1215–1224, 2016.

Robert Kleinberg and Nicole Immorlica. Recharging bandits. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 309–319. IEEE, 2018.

Branislav Kveton, Csaba Szepesvari, Zheng Wen, and Azin Ashkan. Cascading bandits: Learning to rank in the cascade model. In *International Conference on Machine Learning*, pages 767–776, 2015a.

Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Tight regret bounds for stochastic combinatorial semi-bandits. In *18th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 535–543, 2015b.

Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

Nir Levine, Koby Crammer, and Shie Mannor. Rotting bandits. In *Advances in Neural Information Processing Systems*, pages 3074–3083, 2017.

Ciara Pike-Burke and Steffen Grunewalder. Recovering bandits. In *Advances in Neural Information Processing Systems*, pages 14122–14131, 2019.

Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th International Conference on Machine Learning*, pages 784–791. ACM, 2008.

Julien Seznec, Andrea Locatelli, Alexandra Carpentier, Alessandro Lazaric, and Michal Valko. Rotting bandits are no harder than stochastic ones. In *The 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 2564–2572, 2019.

Cem Tekin and Mingyan Liu. Online learning of rested and restless bandits. *IEEE Transactions on Information Theory*, 58(8):5588–5611, 2012.

Romain Warlop, Alessandro Lazaric, and Jérémie Mary. Fighting boredom in recommender systems with linear reinforcement learning. In *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 1764–1773, USA, 2018. Curran Associates Inc.

Peter Whittle. Restless bandits: Activity allocation in a changing world. *Journal of applied probability*, 25 (A):287–298, 1988.