Rotting bandits are not harder than stochastic ones

Julien Seznec Lelivrescolaire.fr
Andrea Locatelli OvGU Magdeburg
Alexandra Carpentier OvGU Magdeburg
Alessandro Lazaric FAIR Paris
Michal Valko Inria Lille

Abstract

In stochastic multi-armed bandits, the reward distribution of each arm is assumed to be stationary. This assumption is often violated in practice (e.g., in recommendation systems), where the reward of an arm may change whenever is selected, i.e., rested bandit setting. In this paper, we consider the *non-parametric rotting bandit* setting, where rewards can only decrease. We introduce the *filtering on expanding window average* (FEWA) algorithm that constructs moving averages of increasing windows to identify arms that are more likely to return high rewards when pulled once more. We prove that for an unknown horizon T, and without any knowledge on the decreasing behavior of the K arms, FEWA achieves problem-dependent regret bound of $\widetilde{\mathcal{O}}(\log(KT))$, and a problem-independent one of $\widetilde{\mathcal{O}}(\sqrt{KT})$. Our result substantially improves over the algorithm of Levine et al. (2017), which suffers regret $\widetilde{\mathcal{O}}(K^{1/3}T^{2/3})$. FEWA also matches known bounds for the stochastic bandit setting, thus showing that the rotting bandits are not harder. Finally, we report simulations confirming the theoretical improvements of FEWA.

1 Introduction

The multi-arm bandits framework (Bubeck and Cesa-Bianchi, 2012; Lattimore and Szepesvári, 2019) formalizes the exploration-exploitation dilemma in online learning, where an agent has to trade off the *exploration* of the environment to gather information and the *exploitation* of the current knowledge to maximize reward. In the *stochastic setting* (Thompson, 1933; Auer et al., 2002a), each arm is characterized by a stationary reward distribution. Whenever an arm is pulled, an i.i.d. sample from the corresponding distribution is observed. Despite the extensive algorithmic and theoretical study of this setting, the stationarity assumption is often too restrictive in practice, e.g., the preferences of users may change over time. The *adversarial setting* (Auer et al., 2002b) addresses this limitation by removing any assumption on how the rewards are generated and learning agents should be able to perform well for any *arbitrary* sequence of rewards. While algorithms such as Exp3 (Auer et al., 2002b) are guaranteed to achieve small regret in this setting, their behavior is conservative as all arms are repeatedly explored to avoid incurring too much regret because of unexpected changes in arms' values. This behavior results in unsatisfactory performance in practice, where arms' values, while non-stationary, are far from being adversarial. Garivier and Moulines (2011) proposed a variation of the stochastic setting, where the distribution of each arm is *piecewise stationary*. Similarly, Besbes et al. (2014) introduced an adversarial setting where the total amount of change in arms' values is bounded. These settings fall into the *restless* bandit scenario, where the arms' value evolves *independently* from the decisions of the agent. On the other hand, for *rested* bandits, the value of an arm changes only when it is pulled. For instance, the value of a service may deteriorate only when it is actually used, e.g., if a recommender system shows always the same item to the users, they may get bored (Warlop et al., 2018). Similarly, a student can master a frequently taught topic in an intelligent tutoring system and extra learning on that topic would be less effective. A particularly interesting case is represented by the *rotting bandits*, where the value of an arm may decrease whenever pulled. Heidari et al. (2016) studied this problem when rewards are deterministic (i.e., no noise) and showed how a greedy policy (i.e., selecting the arm that returned the largest reward the last time it was pulled) is optimal up to a small constant factor depending on the number of arms K and the largest per-round decay in the arms' value L. Bouneffouf and Féraud (2016) considered the stochastic setting when

Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

the dynamics of the rewards is known up to a constant factor. Finally, Levine et al. (2017) considered both non-parametric and parametric noisy rotting bandits, for which they derive algorithms with regret guarantees. In the non-parametric case, where the decrease in reward is neither constrained nor known, they introduce the *sliding-window average* (wSWA) algorithm, which is shown to achieve a regret to the optimal policy of order $\widetilde{\mathcal{O}}(K^{1/3}T^{2/3})$, where T is the number of rounds in the experiment.

In this paper, we study the non-parametric rotting setting of Levine et al. (2017) and introduce *Filtering on Expanding Window Average* (FEWA) algorithm, a novel method that constructs moving average estimates of increasing windows to identify the arms that are more likely to perform well if pulled once more. Under the assumption that the reward decays are bounded, we show that FEWA achieves a regret of $\widetilde{\mathcal{O}}(\sqrt{KT})$, thus *significantly improving over* wSWA and matching the minimax rate of stochastic bandits up to a logarithmic factor. This shows that learning with non-increasing rewards is not more difficult than in the stationary case. Furthermore, when rewards are constant, we recover *standard problem-dependent regret guarantees* (up to constants), while in the rotting bandit scenario with no noise, the regret reduces to the one of Heidari et al. (2016). Numerical simulations confirm our theoretical results and show the superiority of FEWA over wSWA.

2 Preliminaries

We consider a rotting bandit scenario similar to the one of Levine et al. (2017). At each round t, an agent chooses an arm $i(t) \in \mathcal{K} \triangleq \{1, ..., K\}$ and receives a noisy reward $r_{i(t),t}$. The reward associated to each arm i is a σ^2-sub-Gaussian r.v. with expected value of $\mu_i(n)$, which depends on the number of times n it was pulled before; $\mu_i(0)$ is the initial expected value.[1] Let $\mathcal{H}_t \triangleq \big\{\big\{i(s), r_{i(s),s}\big\}, \forall s < t\big\}$ be the sequence of arms pulled and rewards observed until round t, then

$$r_{i(t),t} \triangleq \mu_{i(t)}(N_{i(t),t}) + \varepsilon_t \quad \text{with } \mathbb{E}[\varepsilon_t|\mathcal{H}_t] = 0$$
$$\text{and } \forall \lambda \in \mathbb{R}, \quad \mathbb{E}\big[e^{\lambda \varepsilon_t}\big] \leq e^{\frac{\sigma\lambda^2}{2}},$$

where $N_{i,t} \triangleq \sum_{s=1}^{t-1} \mathbb{I}\{i(t) = i\}$ is the number of times arm i is pulled before round t. We use $r_i(n)$ to denote the random reward of arm i when pulled for the n+1-th time, i.e., $r_{i(t),t} = r_{i(t)}(N_{i(t),t})$. We introduce a non-parametric rotting assumption with bounded decay.

Assumption 1. *The reward functions μ_i are non-increasing with bounded decays $-L \leq \mu_i(n+1) - \mu_i(n) \leq 0$. The initial expected value is bounded as $\mu_i(0) \in [0, L]$. We refer to this set of functions as $\mathcal{L}_L$.*

[1] Our definition slightly differs from the one of Levine et al. (2017). We use $\mu_i(n)$ for the expected value of arm i *after n pulls* instead of when it is pulled *for the n-th time*.

The learning problem A learning policy π is a function from the history of observations to arms, i.e., $\pi(\mathcal{H}_t) \in \mathcal{K}$. In the following, we often use $\pi(t) \triangleq \pi(\mathcal{H}_t)$. The performance of a policy π is measured by the (expected) rewards accumulated over time,

$$J_T(\pi) \triangleq \sum_{t=1}^{T} \mu_{\pi(t)}\big(N_{\pi(t),t}\big).$$

Since π depends on the (random) history observed over time, $J_T(\pi)$ is also random. We define the expected cumulative reward as $\overline{J}_T(\pi) \triangleq \mathbb{E}\big[J_T(\pi)\big]$. We now restate a characterization of the optimal (oracle) policy.

Proposition 1 (Heidari et al., 2016)**.** *If the expected value of each arm $\{\mu_i(n)\}_{i,n}$ is known, the policy $\pi^\star$ maximizing the expected cumulative reward $\overline{J_T}(\pi)$ is greedy at each round, i.e.,*

$$\pi^\star(t) = \arg\max_i \mu_i(N_{i,t}). \tag{1}$$

We denote by $J^\star \triangleq \overline{J}_T(\pi^\star) = J_T(\pi^\star)$, the cumulative reward of the optimal policy.

The objective of a learning algorithm is to implement a policy π with performance as close to $\pi^\star$'s as possible. We define the (random) regret as

$$R_T(\pi) \triangleq J^\star - J_T(\pi). \tag{2}$$

Notice that the regret is measured against an optimal allocation over arms rather than a fixed-arm policy as it is a case in adversarial and stochastic bandits. Therefore, even the adversarial algorithms that one could think of applying in our setting (e.g., Exp3 of Auer et al., 2002a) are not known to provide any guarantee for our definition of regret. On the other hand, for constant $\mu_i(n)$-s, our problem and definition of regret reduce to the one of standard stochastic bandits.

Let $N^\star_{i,T}$ be the (deterministic) number of times that arm i is pulled by the oracle policy $\pi^\star$ up to time T (excluded). Similarly, for a policy π, let $N^\pi_{i,T}$ be the (random) number pulls of arm i. The cumulative reward can be rewritten as

$$J_T(\pi) = \sum_{t=1}^{T} \sum_{i \in \mathcal{K}} \mathbb{I}_{\{\pi(t)=i\}} \mu_i\big(N^\pi_{i,t}\big) = \sum_{i \in \mathcal{K}} \sum_{s=0}^{N^\pi_{i,T}} \mu_i(s).$$

Then, we can conveniently rewrite the regret as

$$R_T(\pi) = \sum_{i\in\mathcal{K}} \left(\sum_{s=0}^{N^\star_{i,T}} \mu_i(s) - \sum_{s=0}^{N^\pi_{i,T}} \mu_i(s) \right)$$
$$= \sum_{i\in \text{UP}} \sum_{s=N^\pi_{i,T}+1}^{N^\star_{i,T}} \mu_i(s) - \sum_{i\in \text{OP}} \sum_{s=N^\star_{i,T}+1}^{N^\pi_{i,T}} \mu_i(s), \quad (3)$$

where we define $\text{UP} \triangleq \left\{ i \in \mathcal{K} | N^{\pi^\star}_{i,T} > N^\pi_{i,T} \right\}$ and likewise $\text{OP} \triangleq \left\{ i \in \mathcal{K} | N^{\pi^\star}_{i,T} < N^\pi_{i,T} \right\}$ as the sets of arms that are respectively under-pulled and over-pulled by π w.r.t. the optimal policy.

Known regret bounds We report existing regret bounds for two special cases. We start with the minimax regret lower bound for stochastic bandits.

Proposition 2. *(Auer et al., 2002b, Thm. 5.1) For any learning policy π and any horizon T, there exists a stochastic stationary problem $\left\{ \mu_i(n) \triangleq \mu_i \right\}_i$ with K σ-sub-Gaussian arms such that π suffers a regret*

$$\mathbb{E}[R_T(\pi)] \geq \frac{\sigma}{10} \min\left(\sqrt{KT}, T\right).$$

where the expectation is w.r.t. both the randomization over rewards and algorithm's internal randomization.

Heidari et al. (2016) derived regret lower and upper bounds for deterministic rotting bandits (i.e., $\sigma = 0$).

Proposition 3. *(Heidari et al., 2016, Thm. 3) For any learning policy π, there exists a deterministic rotting bandits (i.e., $\sigma = 0$) satisfying Assumption 1 with bounded decay L such that π suffers an expected regret*

$$\mathbb{E}[R_T(\pi)] \geq \frac{L}{2}(K-1).$$

Let π^{σ_0} be the greedy policy that selects at each round the arm with the largest reward observed so far, i.e., $\pi^{\sigma_0}(t) \triangleq \arg\max_i (\mu_i(N_{i,t}-1))$. For any deterministic rotting bandits (i.e., $\sigma = 0$) satisfying Assumption 1 with bounded decay L, π^{σ_0} suffers an expected regret

$$\mathbb{E}[R_T(\pi^{\sigma_0})] \leq L(K-1).$$

Any problem in the two settings above is a rotting problem with parameters (σ, L). Therefore, the performance of any algorithm on the general rotting problem is also bounded by these two lower bounds.

3 FEWA: Filtering on expanding window average

Since the expected rewards μ_i change over time, the main difficulty in the non-parametric rotting bandits is that we cannot rely on all samples observed until time t to predict which arm is likely to return the highest reward in the future. In fact, the older a sample, the less representative it is for future rewards. This suggests constructing estimates using the more recent samples. Nonetheless, discarding older rewards reduces the number of samples used in the estimates, thus increasing their variance. In Alg. 1 we introduce FEWA (or π_{F}) that at each round t, relies on estimates using windows of increasing length to filter out arms that are suboptimal with high probability and then pulls the least pulled arm among the remaining arms.

Algorithm 1 FEWA

Input: σ, $\mathcal{K}$, δ_0, α
1: pull each arm once, collect reward, and initialize $N_{i,K} \leftarrow 1$
2: **for** $t \leftarrow K+1, K+2, \ldots$ **do**
3: $\quad \delta_t \leftarrow \delta_0/(Kt^\alpha)$
4: $\quad h \leftarrow 1$ {*initialize bandwidth*}
5: $\quad \mathcal{K}_1 \leftarrow \mathcal{K}$ {*initialize with all the arms*}
6: $\quad i(t) \leftarrow$ `none`
7: $\quad$ **while** $i(t)$ is `none` **do**
8: $\quad\quad \mathcal{K}_{h+1} \leftarrow \text{FILTER}(\mathcal{K}_h, h, \delta_t)$
9: $\quad\quad h \leftarrow h+1$
10: $\quad\quad$ **if** $\exists i \in \mathcal{K}_h$ such that $N_{i,t} = h$ **then**
11: $\quad\quad\quad i(t) \leftarrow \arg\min_{i\in\mathcal{K}_h} N_{i,t}$
12: $\quad\quad$ **end if**
13: $\quad$ **end while**
14: $\quad$ receive $r_i(N_{i,t+1}) \leftarrow r_{i(t),t}$
15: $\quad N_{i(t),t} \leftarrow N_{i(t),t-1} + 1$
16: $\quad N_{j,t} \leftarrow N_{j,t-1}, \quad \forall j \neq i(t)$
17: **end for**

We first describe the subroutine FILTER in Alg. 2, which receives a set of active arms $\mathcal{K}_h$, a window h, and a confidence parameter δ as input and returns an updated set of arms $\mathcal{K}_{h+1}$. For each arm i that has been pulled n times, the algorithm constructs an estimate $\widehat{\mu}^h_i(n)$ that averages the $h \leq n$ most recent rewards observed from i. The subroutine FILTER discards all the arms whose mean estimate (built with window h) from $\mathcal{K}_h$ is lower than the empirically best arm by more than twice a threshold $c(h, \delta_t)$ constructed by standard Hoeffding's concentration inequality (see Prop. 4).

The FILTER subroutine is used in FEWA to incrementally refine the set of active arms, starting with a window of size 1, until the condition at Line 10 is met. As a result, $\mathcal{K}_{h+1}$ only contains arms that passed the filter for all windows from 1 up to h. Notice that it is important to start filtering arms from a small window and to keep refining the previous set of active arms. In fact, the estimates constructed using a small window use recent rewards, which are closer to the future value of an arm. As a result, if there is enough evidence that

Algorithm 2 FILTER

Input: $\mathcal{K}_h$, h, δ_t
1: $c(h, \sigma, \delta_t) \leftarrow \sqrt{(2\sigma^2/h)\log(1/\delta_t)}$
2: **for** $i \in \mathcal{K}_h$ **do**
3: $\quad \widehat{\mu}_i^h(N_{i,t}) \leftarrow \frac{1}{h}\sum_{j=1}^{h} r_i(N_{i,t}-j)$
4: **end for**
5: $\widehat{\mu}^h_{\max,t} \leftarrow \max_{i\in\mathcal{K}_h} \widehat{\mu}_i^h(N_{i,t})$
6: **for** $i \in \mathcal{K}_h$ **do**
7: $\quad \Delta_i \leftarrow \widehat{\mu}^h_{\max,t} - \widehat{\mu}_i^h(N_{i,t})$
8: $\quad$ **if** $\Delta_i \le 2c(h, \sigma, \delta_t)$ **then**
9: $\quad\quad$ add i to $\mathcal{K}_{h+1}$
10: $\quad$ **end if**
11: **end for**
Output: $\mathcal{K}_{h+1}$

an arm is suboptimal already at a small window h, it should be directly discarded. On the other hand, a suboptimal arm may pass the filter for small windows as the threshold $c(h, \sigma, \delta_t)$ is large for small h (i.e., as few samples are used in constructing $\widehat{\mu}_i^h(N_{i,t})$, the estimation error may be high). Thus, FEWA keeps refining $\mathcal{K}_h$ for larger windows in the attempt of constructing more accurate estimates and discard more suboptimal arms. This process stops when we reach a window as large as the number of samples for at least one arm in the active set $\mathcal{K}_h$ (i.e., Line 10). At this point, increasing h would not bring any additional evidence that could refine $\mathcal{K}_h$ further (recall that $\widehat{\mu}_i^h(N_{i,t})$ is not defined for $h > N_{i,t}$). Finally, FEWA selects the active arm $i(t)$ whose number of samples matches the current window, i.e., the least pulled arm in $\mathcal{K}_h$. The set of available rewards and the number of pulls are then updated accordingly.

Runtime and memory usage At each round t, FEWA needs to store and update up to t averages per-arm. Since moving from an average computed on window h to $h+1$ can be done incrementally at a cost $\mathcal{O}(1)$, the worst-case time and memory complexity per round is $\mathcal{O}(Kt)$, which amounts to a total $\mathcal{O}(KT^2)$ cost. This is not practical for large T.[2] We have a fix.

In App. E we detail EFF-FEWA, an efficient variant of FEWA. EFF-FEWA is built around two main ideas.[3] First, at any time t we can avoid calling FILTER for all possible windows h starting from 1 with an increment of 1. In fact, the confidence interval $c(h, \sigma, \delta_t)$ decreases as $1/\sqrt{h}$ and we could select windows h with an exponential increment so that confidence intervals between two consecutive calls to FILTER have a constant ratio. In practice, we replace the window increment (Line 9 of FEWA) by a geometric window $h \triangleq 2^j$. This modification alone is not enough to reduce the computation. While we reduce the number of estimates that we construct, updating $\widehat{\mu}_i^h$ from $h = 2^j$ to $h = 2^{j+1}$ still requires spanning over past samples, thus leading to the same $\mathcal{O}(Kt)$ complexity in the worst-case. In order to reduce the overall complexity, we avoid recomputing $\widehat{\mu}_i^h$ at each call of FILTER and by replacing it with *precomputed* estimates. Whenever $N_{i,t} = 2^j$ for some j, we create an estimate $\widehat{s}^c_{i,j}$ by averaging all the last $N_{i,t}$ samples. These estimates are then used whenever FILTER is called with $h = 2^j$. Instead of updating $\widehat{s}^c_{i,j}$ at each new sample, we create an associated *pending* estimate $\widehat{s}^p_{i,j}$ which averages all the more recent samples. More formally, let t be the time when $N_{i,t} = 2^j$, then $\widehat{s}^p_{i,j}$ is initialized at 0 and it then stores the average of all the samples observed from t to t', when $N_{i,t'} = 2^{j+1}$ (i.e., $\widehat{s}^p_{i,j}$ is averaging at most 2^j samples). At this point, the 2^j samples averaged in $\widehat{s}^c_{i,j}$ are *outdated* and they are replaced by the new average $\widehat{s}^p_{i,j}$, which is then reinitialized to 0. The sporadic update of the precomputed estimates and the small number of them drastically reduces per-round time and space complexity to $\mathcal{O}(K\log t)$. Furthermore, EFF-FEWA preservers the same regret guarantees as FEWA. In the worst case, $\widehat{s}^c_{i,j}$ may not cover the last $2^{j-1}-1$ samples. Nonetheless, the precomputed estimates with smaller windows (i.e., $j' < j$) are updated more frequently, thus effectively covering the $2^{j-1}-1$ samples "missed" by $\widehat{s}^c_{i,j}$. As a result, the active sets returned by FILTER are still accurate enough to derive regret guarantees that are only a constant factor worse than FEWA (App. E).

4 Regret Analysis

We first give problem-independent regret bound for FEWA and sketch its proof in Sect. 4.1. Then, we derive problem-dependent guarantees in Sect. 4.2.

Theorem 1. *For any rotting bandit scenario with means $\{\mu_i(n)\}_{i,n}$ satisfying Asm. 1 with bounded decay L and any time horizon T,* FEWA *run with $\alpha = 5$ and $\delta_t = 1/(Kt^5)$, suffers an expected regret*[4] *of*

$$\mathbb{E}[R_T(\pi_F)] \le 13\sigma(\sqrt{KT}+K)\sqrt{\log(KT)} + KL.$$

Comparison to Levine et al. (2017) The regret of wSWA is bounded by $\widetilde{\mathcal{O}}(\mu_{\max}^{1/3}K^{1/3}T^{2/3})$ for rotting functions with range in $[0, \mu_{\max}]$. In our setting, we do not restrict rewards to stay positive but we bound the per-round decay by L, thus leading to rotting functions with range in $[-LT, L]$. As a result, when applying

[2] This analysis is worst-case. In many cases, the number of samples for the suboptimal arms may be much smaller than $\mathcal{O}(t)$. For instance, in stochastic bandits it is as little as $\mathcal{O}(\log t)$, thus reducing the complexity to $\mathcal{O}(KT\log T)$.

[3] As pointed by a reviewer, a similar yet different approach has appeared independently in the context of streaming mining (Bifet and Gavaldà, 2007).

[4] See Corollary 3 and 4 for the high-probability result.

wSWA to our setting, we should set $\mu_{\max} = L(T+1)$, which leads to $\mathcal{O}(T)$ regret, thus showing that according to its original analysis, wSWA may not be able to learn in our general setting. On the other hand, we could use FEWA in the setting of Levine et al. (2017) by setting $L = \mu_{\max}$ as the largest drop that could occur. In this case, FEWA suffers a regret of $\widetilde{\mathcal{O}}(\sqrt{KT})$, thus significantly improving over wSWA. The improvement is mostly due to the fact that FEWA exploits filters using moving averages with increasing windows to discard arms that are suboptimal w.h.p. Since this process is done at each round, FEWA smoothly tracks changes in the value of each arm, so that if an arm becomes worse later on, other arms would be recovered and pulled again. On the other hand, wSWA relies on a fixed exploratory phase where all arms are pulled in a round-robin way and the tracking is performed using averages constructed with a fixed window. Moreover, FEWA is anytime, while the fixed exploratory phase of wSWA requires either to know T or to resort to a doubling trick, which often performs poorly in practice.

Comparison to deterministic rotting bandits For $\sigma = 0$, our upper bound reduces to KL, thus matching the prior (upper and lower) bound of Heidari et al. (2016) for deterministic rotting bandits. Moreover, the additive decomposition of regret shows that there is *no coupling* between the stochastic problem and the rotting problem as terms depending on the noise level σ are separated from the terms depending on the rotting level L, while in wSWA these are coupled by a $L^{1/3}\sigma^{2/3}$ factor in the leading term.

Comparison to stochastic bandits The regret of FEWA matches the worst-case optimal regret bound of the standard stochastic bandits (i.e., $\mu_i(n)$s are constant) up to a logarithmic factor. Whether an algorithm can achieve $\mathcal{O}(\sqrt{KT})$ regret bound is an open question. On one hand, FEWA needs confidence bounds to hold for different windows at the same time, which requires an additional union bound and thus larger confidence intervals w.r.t. UCB1. On the other hand, our worst-case analysis shows that some of the difficult problems that reach the worst-case bound of Thm. 1 are realized with constant functions, which is the standard stochastic bandits, for which MOSS-like (Audibert and Bubeck, 2009) algorithms achieve regret guarantees without the $\log T$ factor. Thus, the necessity of the extra $\log T$ factor for the worst-case regret of rotting bandits remains an open problem.

4.1 Sketch of the proof

We now give a sketch of the proof of our regret bound. We first introduce the expected value of the estimators used in FEWA. For any n and $1 \le h \le n$, we define

$$\overline{\mu}_i^h(n) \triangleq \mathbb{E}\big[\widehat{\mu}_i^h(n)\big] = \frac{1}{h}\sum_{j=1}^{h} \mu_i(n-j).$$

Notice that at round t, if the number of pulls of arm i is $N_{i,t}$, then $\overline{\mu}_i^1(N_{i,t}) = \mu_i(N_{i,t}-1)$, which is the expected value of arm i the last time it was pulled. We introduce Hoeffding's concentration inequality and the favorable event that we leverage in the analysis.

Proposition 4. *For any fixed arm i, number of pulls n, and window h, we have that with probability $1-\delta$,*

$$\left|\widehat{\mu}_i^h(n) - \overline{\mu}_i^h(n)\right| \le c(h,\delta) \triangleq \sqrt{\frac{2\sigma^2}{h}\log\frac{1}{\delta}}. \qquad (4)$$

For any round t and confidence $\delta_t \triangleq \delta_0/(Kt^\alpha)$, let

$$\xi_t \triangleq \left\{\forall i \in \mathcal{K}, \forall n \le t, \forall h \le n, \left|\widehat{\mu}_i^h(n) - \overline{\mu}_i^h(n)\right| \le c(h,\delta_t)\right\}$$

be the event under which the estimates constructed by FEWA *at round t are all accurate up to $c(h,\delta_t)$. Taking a union bound gives $\mathbb{P}(\xi_t) \ge 1 - Kt^2\delta_t/2$.*

Active set We derive an important lemma that provides support for the arm selection process obtained by a series of refinements through the Filter subroutine. Recall that at any round t, after pulling arms $\{N_{i,t}^{\pi_F}\}_i$ the greedy (oracle) policy would select an arm

$$i_t^\star\Big(\big\{N_{i,t}^{\pi_F}\big\}_i\Big) \in \arg\max_{i\in\mathcal{K}} \mu_i\big(N_{i,t}^{\pi_F}\big).$$

We denote by $\mu_t^+(\pi_F) \triangleq \max_{i\in\mathcal{K}} \mu_i(N_{i,t}^{\pi_F})$, the reward obtained by pulling $i_t^\star$. The dependence on π_F in the definition of $\mu_t^+(\pi_F)$ stresses the fact that we consider what the oracle policy would do at the state reached by π_F. While FEWA cannot directly match the performance of the oracle arm, the following lemma shows that the reward averaged over the last h pulls of any arm in the active set is close to the performance of the oracle arm up to four times $c(h,\delta_t)$.

Lemma 1. *On the favorable event ξ_t, if an arm i passes through a filter of window h at round t, i.e., $i \in \mathcal{K}_h$, then the average of its h last pulls satisfies*

$$\overline{\mu}_i^h(N_{i,t}^{\pi_F}) \ge \mu_t^+(\pi_F) - 4c(h,\delta_t). \qquad (5)$$

This result relies heavily on the non-increasing assumption of rotting bandits. In fact, for any arm i and any window h, we have

$$\overline{\mu}_i^h(N_{i,t}^{\pi_F}) \ge \overline{\mu}_i^1(N_{i,t}^{\pi_F}) \ge \mu_i(N_{i,t}^{\pi_F}).$$

While the inequality above for i_t^* trivially satisfies Eq. 5, Lem. 1 is proved by integrating the possible errors introduced by the filter in selecting active arms due to the error of the empirical estimates.

Relating FEWA to the oracle policy While Lem. 1 provides a link between the value of the arms returned by the filter and the oracle arm, $i_t^\star$ is defined according to the number of pulls obtained by FEWA up to t, which may significantly differ from the sequence of pulls of the oracle policy. In order to bound the regret, we need to relate the actual performance of the optimal policy to the value of the arms pulled by FEWA. Let $h_{i,t} \triangleq \left|N_{i,t}^{\pi_\mathrm{F}} - N_{i,t}^{\pi^\star}\right|$ be the absolute difference in the number of pulls between π_F and the optimal policy up to t. Since $\sum_i N_{i,t}^{\pi_\mathrm{F}} = \sum_i N_{i,t}^{\pi^\star} = t$, we have that $\sum_{i\in\mathrm{OP}} h_{i,t} = \sum_{i\in\mathrm{UP}} h_{i,t}$ which means that there are as many total overpulls as underpulls. Let $j \in \mathrm{UP}$ be an underpulled arm[5] with $N_{j,T}^{\pi_\mathrm{F}} < N_{j,T}^{\pi^\star}$, then, for all $s \in \{0,\dots,h_{j,t}\}$, we have the inequality

$$\mu_T^+(\pi_\mathrm{F}) = \max_{i\in\mathcal{K}} \mu_i(N_{i,T}^{\pi_\mathrm{F}}) \geq \mu_j(N_{j,T}^{\pi_\mathrm{F}} + s). \tag{6}$$

As a result, from Eq. 3 we have the regret upper bound

$$R_T(\pi_\mathrm{F}) \leq \sum_{i\in\mathrm{OP}} \sum_{h=0}^{h_{i,T}-1} \Big(\mu^+(\pi_\mathrm{F}) - \mu_i(N_{i,T}^{\pi^\star} + h)\Big), \tag{7}$$

where we have obtained the inequality by bounding $\mu_i(t') \leq \mu_T^+(\pi_\mathrm{F})$ in the first summation and then using $\sum_{i\in\mathrm{OP}} h_{i,T} = \sum_{i\in\mathrm{UP}} h_{i,T}$. While the previous expression shows that we can just focus on over-pulled arms in OP, it is still difficult to directly control the expected reward $\mu_i(N_{i,T}^{\pi^\star} + h)$, as it may change at each round (by at most L). Nonetheless, we notice that its cumulative sum can be directly linked to the average of the expected reward over a suitable window. In fact, for any $i \in \mathrm{OP}$ and $h_{i,T} \geq 2$, we have

$$(h_{i,T}-1)\overline{\mu}_i^{h_{i,T}-1}(N_{i,T}-1) = \sum_{t'=0}^{h_{i,T}-2} \mu_i(N_{i,T}^{\pi^\star} + t').$$

At this point we can control the regret for each $i \in \mathrm{OP}$ in Eq. 7 by applying the following corollary of Lem. 1.

Corollary 1. *Let $i \in \mathrm{OP}$ be an arm overpulled by* FEWA *at round t and $h_{i,t} \triangleq N_{i,t}^{\pi_\mathrm{F}} - N_{i,t}^{\pi^\star} \geq 1$ be the difference in the number of pulls w.r.t. the optimal policy $\pi^\star$ at round t. On the favorable event ξ_t, we have*

$$\mu_t^+(\pi_\mathrm{F}) - \overline{\mu}_i^{h_{i,t}}(N_{i,t}) \leq 4c(h_{i,t},\delta_t). \tag{8}$$

4.2 Problem-dependent guarantees

Since our setting generalizes the standard stochastic bandit setting, a natural question is whether we pay any price for this generalization. While the result of Levine et al. (2017) suggested that learning in rotting bandits could be more difficult, in Thm. 1 we actually proved that FEWA nearly matches the problem-independent regret $\widetilde{\mathcal{O}}(\sqrt{KT})$. We may wonder whether this is true for the *problem-dependent* regret as well.

[5] If such arm does not exist, then π_F suffers no regret.

Remark 1. *Consider a stationary stochastic bandit setting with expected rewards $\{\mu_i\}_i$ and $\mu_\star \triangleq \max_i \mu_i$. Corollary 1 guarantees that for $\delta_t \geq 1/(KT^\alpha)$,*

$$\mu_\star - \mu_i \leq 4c(h_{i,T}-1,\delta_t) = 4\sqrt{\frac{2\alpha\sigma^2\log(KT)}{h_{i,T}-1}}$$
$$\text{or equivalently, } h_{i,T} \leq 1 + \frac{32\alpha\sigma^2\log(KT)}{(\mu_\star-\mu_i)^2}. \tag{9}$$

Therefore, our algorithm matches the lower bound of Lai and Robbins (1985) up to a constant, thus showing that learning in the rotting bandits are never harder than in the stationary case. Moreover, this upper bound is at most α larger than the one for UCB1 (Auer et al., 2002a).[6] The main source of suboptimality is the use of a confidence bound filtering instead of an upper-confidence index policy. Selecting the less pulled arm in the active set is conservative as it requires uniform exploration until elimination, resulting in a factor 4 in the confidence bound guarantee on the selected arm (vs. 2 for UCB), which implies 4 times more overpulls than UCB (see Eq. 9). We conjecture that this may not be necessarily needed and it is an open question whether it is possible to derive either an index policy or a better selection rule. The other source of suboptimality w.r.t. UCB is the use of larger confidence bands because of the higher number of estimators computed at each round (Kt^2 instead of Kt for UCB).

Remark 1 also reveals that Corollary 1 can be used to derive a general problem-dependent result in the rotting case. In particular, with Corollary 1 we upper-bound the maximum number of overpulls by a problem dependent quantity

$$h_{i,T}^+ \triangleq \max\left\{h \leq 1 + \frac{32\alpha\sigma^2\log(KT)}{\Delta_{i,h-1}^2}\right\}, \tag{10}$$
$$\text{where } \Delta_{i,h} \triangleq \min_{j\in\mathcal{K}} \mu_j\big(N_{j,T}^\star - 1\big) - \overline{\mu}_i^h\big(N_{i,t}^\star + h\big).$$

We then use Corollary 1 again to upper-bound the regret caused by $h_{i,T}^+$ overpulls for each arm, leading to Corollary 2 (see the full proof in App. D).

Corollary 2. *For $\delta_t \triangleq 1/(Kt^5)$ and $C_\alpha \triangleq 32\alpha\sigma^2$, the regret of* FEWA *is bounded as*

$$\mathbb{E}[R_T(\pi_\mathrm{F})] \leq \sum_{i\in\mathcal{K}} \left(\frac{C_5\log(KT)}{\Delta_{i,h_{i,T}^+-1}} + \sqrt{C_5\log(KT)} + L\right).$$

[6] To make the results comparable to the one of Auer et al. (2002a), we need to replace $2\sigma^2$ by $1/2$ for sub-Gaussian noise.

5 Numerical simulations

2-arms We design numerical simulations to study the difference between wSWA and FEWA. We consider rotting bandits with two arms defined as

$$\mu_1(n) = 0, \quad \forall n \leq T \quad \text{and} \quad \mu_2(n) = \begin{cases} \frac{L}{2} & \text{if } n < \frac{T}{4}, \\ -\frac{L}{2} & \text{if } n \geq \frac{T}{4}. \end{cases}$$

The rewards are then generated by applying a Gaussian i.i.d. noise $\mathcal{N}(0, \sigma = 1)$. The single point of non-stationarity in the second arm is designed to satisfy Asm. 1 with a bounded decay L. It is important to notice that in this specific case, L also plays the role of defining the gap Δ between the arms, which is known to heavily impact the performance both in stochastic bandits and in the rotting bandits (Cor. 2). In particular, for any learning strategy, the gap between the two arms is always $\Delta = |\mu_1(n_1) - \mu_2(n_2)| = L/2$. Recall that in stochastic bandits, the problem independent bound $\mathcal{O}(\sqrt{KT})$ is obtained by the worst-case choice of $\Delta \triangleq \sqrt{K/T}$. In the two-arm setting defined above, the optimal allocation is $N^\star_{1,T} = 3T/4$ and $N^\star_{2,T} = T/4$.

Algorithms Both algorithms have a parameter α to tune. In wSWA, α is a multiplicative constant to tune the window. We test four different values of α, including the recommendation of Levine et al. (2017), $\alpha = 0.2$. In general, the smaller the α, the smaller the averaging window and the more reactive the algorithm is to large drops. Nonetheless, in stationary regimes, this may correspond to high variance and poor regret. On the other hand, a large value of α may reduce variance but increase the bias in case of rapidly rotting arms. Thm. 3.1 of Levine et al. (2017) reveals this trade-off in the regret bound of wSWA, which has a factor $(\alpha\mu_{\max} + \alpha^{-1/2})$, where $\mu_{\max}$ is the largest arm value. The best choice of α is then $\mu_{\max}^{-2/3}$, which reduces the previous constant to $\mu_{\max}^{1/3}$. In our experiment, $\mu_{\max} = L$ and we could expected that for any fixed α, wSWA may perform well in cases when $\alpha \approx \mu_{\max}^{-2/3}$, while the performance may degrade for larger $\mu_{\max}$.

In FEWA, α tunes the confidence $\delta_t = 1/(t^\alpha)$ used in $c(h, \delta_t)$. While our analysis suggests $\alpha = 5$, the analysis of confidence intervals, union bounds, and filtering algorithms is too conservative. Therefore, we use more aggressive values, $\alpha \in \{0.03, 0.06, 0.1\}$.

Experiments In Fig. 1, we compare the performance of the two algorithms and their dependence on L. The first plot shows the regret at T for various values of L. The second and the third plot show the regret as a function of time for $L = 0.2$ and $L = 4.24$, which corresponds to the worst empirical performance for FEWA and to the $L \gg \sigma$ regime respectively. All experiments have $T = 10000$ and are averaged over 500 runs.

Before discussing the results, we point out that in the rotting setting, the regret can increase and decrease over time. Consider two simple policies: π_1, which first pulls arm 1 for $N^\star_{1,T}$ times and then arm 2 for $N^\star_{2,T}$ times, and π_2 in reversed order (first arm 2 and then arm 1). If we take π_2 as reference, π_1 has an increasing regret for the first $T/4$ rounds, which then would plateau from $T/4$ up to $3T/4$ as both π_1 and π_2 are pulling arm 1. Then from $3T/4$ to T, the regret of π_1 would reverse back to 0 since π_2 would keep selecting arm 1 and getting a reward of 0, while π_1 transitions to pulling arm 2 with a reward of $L/2$.

Results Fig. 2 shows that the performance of wSWA depends on the proper tuning of α w.r.t. $\mu_{\max} = L$, as predicted by Thm. 3.1 of Levine et al. (2017). In fact, for small values of L, the best choice is $\alpha = 0.2$, while for larger values of L a smaller α is preferable. In particular, when L grows very large, the regret tends to grow linearly with L. On the other hand, FEWA seems much more robust to different values of L. Whenever T and σ are large compared to L, Thm. 1 suggests that the regret of FEWA is dominated by $\mathcal{O}(\sigma\sqrt{KT})$, while the term KL becomes more relevant for large values of the drop L. We also notice that since L defines the gap between the value of μ_1 and μ_2, the problem-independent bound is achieved for the worst-case choice of $L \sim 2\sqrt{K/T}$, when the regret of FEWA is indeed the largest. Fig. 1 middle and right confirm these findings for the extreme choice of the worst-case value of L and the regime where the drop is much larger than the noise level, i.e., where the term KL dominates the regret. We conclude that FEWA is more robust than wSWA as it almost always achieves the best performance across different problems while being agnostic to the value of L. On the other hand, wSWA's performance is very sensitive to the choice of α and the same value of the parameter may correspond to significantly different performance depending on L. Finally, we notice that EFF-FEWA has a regret comparable to FEWA when L is large, while for a small value of L, EFF-FEWA suffers the cost of the delay in the update of its statistics, which is larger for the last filter.

10-arms We also tested a rotting setting with 10 arms. The mean of 1 arm is constant with value 0 while the means of 9 arms abruptly decrease after 1000 pulls from $+\Delta_i$ to $-\Delta_i$. Δ_i is ranging from 0.001 to 10 in a geometric sequence. In this setting, the regret can be written as $R_T(\pi) = \sum_{i=1}^{9} h_{i,T}\Delta_i$. Hence, the regret per arm is $R^i_T(\pi) \triangleq \Delta_i h_{i,T}$. In Fig. 2, we compare the performance of different algorithms for their best parameter. The left plot shows the average regret as a

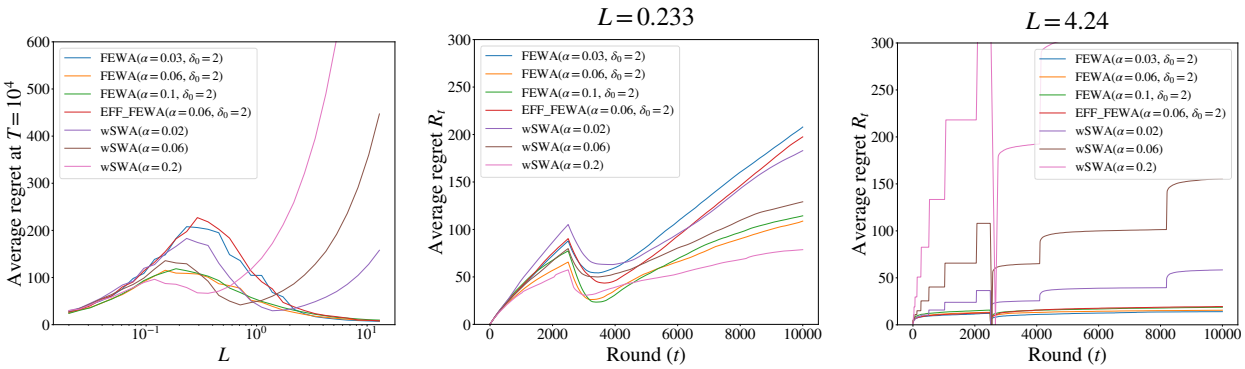

Figure 1: Comparison between FEWA and wSWA in the 2-arm setting. **Left:** Regret at $T = 10000$ for different values of L. **Middle-right:** Regret over time for $L = 0.2$ (worst case for FEWA) and $L = 4.24$ (case of $L \gg \sigma$).

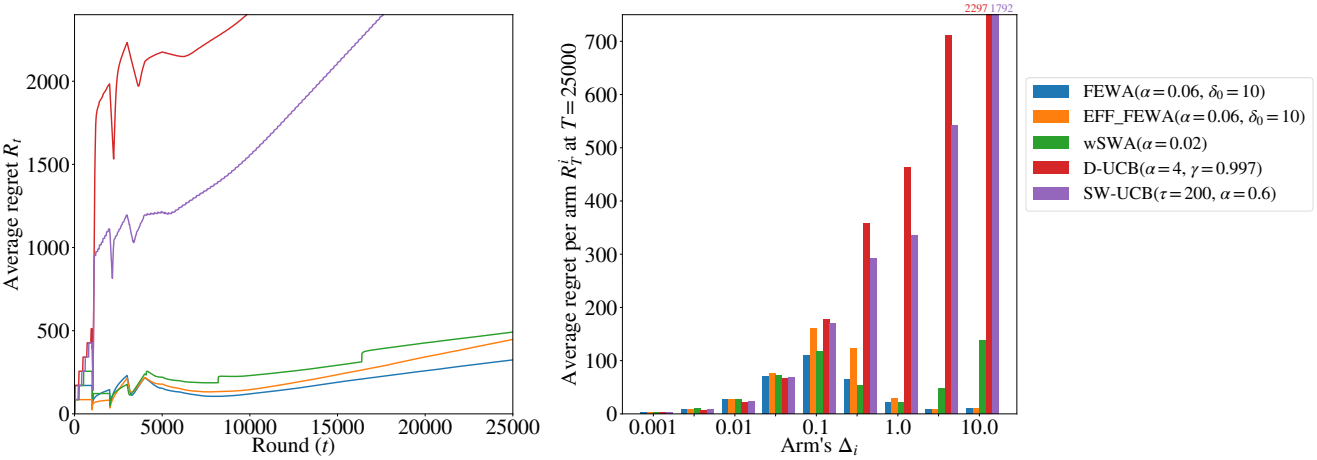

Figure 2: 10-arm setting. **Left:** Regret over time. **Right:** Regret per arm at the end of the experiment.

function of time. The right plot shows the regret per arm (indexed by Δ_i) at the end of the experiment.

Results On Fig. 2 left, we see that FEWA outperforms wSWA at the end of the game. We remark that the best tuning for wSWA corresponds to a rather small window which corresponds to a good choice for $L = 2$ in the 2-arm settings. Similar results can be observed on the right plot: wSWA slightly outperforms FEWA for $\Delta_i = 0.3$ and $\Delta_i = 1$. However, this window size is too large for $\Delta_i = 3.2$ and $\Delta_i = 10$. We also remark that EFF-FEWA is penalized for arms with small Δ_i, for which the impact of the delay is more significant.

We also tested SW-UCB and D-UCB (Garivier and Moulines, 2011) with parameters tuned for this experiment. While the two algorithms are known benchmarks for non-stationary *restless* bandits, they are penalized in our *rested* bandits problem. Indeed, they keep exploring arms that have not been pulled for many rounds which is detrimental in our case as the arms stay constant when they are not pulled. Hence, there is no good choice for their forgetting parameters: A fast forgetting rate makes the policies repeatedly pull bad arms (whose mean rewards do not change when they are not pulled in the rested setting) while a slow forgetting rate makes the policies not able to adapt to abrupt shifts.

6 Conclusion

We introduced FEWA, a novel algorithm for the non-parametric rotting bandits. We proved that FEWA achieves an $\widetilde{\mathcal{O}}(\sqrt{KT})$ regret without any knowledge of the decays by using moving averages with a window that effectively adapts to the changes in the expected rewards. This result greatly improves over the wSWA algorithm by Levine et al. (2017), that suffers a regret of order $\widetilde{\mathcal{O}}(K^{1/3}T^{2/3})$. Thus our result shows that the rotting bandit scenario is not harder than the stochastic one. Our technical analysis of FEWA hinges on the *adaptive* nature of the window size. The most interesting aspect of the proof technique is that confidence bounds are used not only for the action selection but also for the *data* selection, i.e., to identify the best window to trade off the bias and the variance in estimating the current value of each arm.

Acknowledgements We thank Nir Levine for his helpful remarks. The research presented was supported by European CHIST-ERA project DELTA, French Ministry of Higher Education and Research, Nord-Pas-de-Calais Regional Council, Inria and Otto-von-Guericke-Universität Magdeburg associated-team north-European project Allocate, and French National Research Agency projects ExTra-Learn (n.ANR-14-CE24-0010-01) and BoB (n.ANR-16-CE23-0003). The work of A. Carpentier is also partially supported by the Deutsche Forschungsgemeinschaft (DFG) Emmy Noether grant MuSyAD (CA 1488/1-1), by the DFG - 314838170, GRK 2297 MathCoRe, by the DFG GRK 2433 DAEDALUS, by the DFG CRC 1294 Data Assimilation, Project A03, and by the UFA-DFH through the French-German Doktorandenkolleg CDFA 01-18. This research has also benefited from the support of the FMJH Program PGMO and from the support to this program from Criteo. Part of the computational experiments was conducted using the Grid'5000 experimental testbed (https://www.grid5000.fr).

References

Jean-Yves Audibert and Sébastien Bubeck. Minimax policies for adversarial and stochastic bandits. In *Conference on Learning Theory*, 2009.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002a.

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multi-armed bandit problem. *Journal on Computing*, 32(1):48–77, 2002b.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Stochastic multi-armed bandit problem with non-stationary rewards. In *Neural Information Processing Systems*, 2014.

Albert Bifet and Ricard Gavaldà. Learning from time-changing data with adaptive windowing. In *International Conference on Data Mining*, 2007.

Djallel Bouneffouf and Raphael Féraud. Multi-armed bandit problem with known trend. *Neurocomputing*, 205(C):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*, 5:1–122, 2012.

Aurélien Garivier and Eric Moulines. On upper-confidence-bound policies for switching bandit problems. In *Algorithmic Learning Theory*, 2011.

Hoda Heidari, Michael Kearns, and Aaron Roth. Tight policy regret bounds for improving and decaying bandits. In *International Conference on Artificial Intelligence and Statistics*, 2016.

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

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. 2019.

Nir Levine, Koby Crammer, and Shie Mannor. Rotting bandits. In *Neural Information Processing Systems*, 2017.

William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25:285–294, 1933.

Romain Warlop, Alessandro Lazaric, and Jérémie Mary. Fighting boredom in recommender systems with linear reinforcement learning. In *Neural Information Processing Systems*, 2018.