Rotting bandits

1. Rested rotting bandits are not harder th

1.1 Rested rotting bandit : model and preliminaries

1.1.1 Model

Feedback loop

At each round t, an agent chooses an arm $i_t \in \mathscr{K} \triangleq \{1,...,K\}$ and receives a noisy reward o_t. The reward associated to each arm i is a σ^2-sub-Gaussian random variable 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. 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*. Let $\mathscr{H}_t \triangleq \{\{i_s, o_s\}, \forall s < t\}$ be the sequence of arms pulled and rewards observed until round t, then

$$o_t \triangleq \mu_{i_t}(N_{i_t,t-1}) + \varepsilon_t \text{ with } \mathbb{E}[\varepsilon_t | \mathscr{H}_t] = 0 \text{ and } \forall \lambda \in \mathbb{R}, \mathbb{E}\left[e^{\lambda \varepsilon_t}\right] \leq e^{\frac{\sigma\lambda^2}{2}}, \tag{1.1}$$

where $N_{i,t} \triangleq \sum_{s=1}^{t} \mathbb{I}\{i_s = i\}$ is the number of times arm i is pulled after round t.

Definition 1.1.1 We introduce $\mathscr{L}_L$, the set of non-increasing reward functions with bounded decay L,

$$\mathscr{L}_L \triangleq \left\{\mu : \{0,\dots,T-1\} \to [-L(T-1), L] \;\middle|\; 0 \leq \mu(n) - \mu(n+1) \leq L \text{ and } \mu(0) \in [0,L]\right\}.$$

R We define the set of constant reward function in $[0,L]$:

$$\mathscr{S}_L \triangleq \left\{\mu : \{0,\dots,T-1\} \to [0,L] \;\middle|\; \mu(n) = \mu_i\right\}.$$

We have that $\mathscr{S}_L \subset \mathscr{L}_L$. Hence, we can conclude that the rotting bandits model include all the stationary bandits problems.

Online and offline objectives

In this chapter, we will only consider deterministic agents which output an arm i at each round t. They are degenerate cases of probabilistic agent, which outputs a probability distribution over arm at each round. For the sake of simplicity, we present only the deterministic formalism.

We will distinguish two types of policies. On the one hand, an offline (or oracle) policy $\pi \in \Pi_{\mathrm{O}}$ is a function which maps the round t and the set of reward functions $\mu \triangleq \{\mu_i\}_{i\in\mathscr{K}}$ to arms, i.e. $\pi(t,\mu) \in \mathscr{K}$. On the other hand, an online (or learning) policy $\pi \in \Pi_{\mathrm{L}}$ is a function from the history of observations at time t (which includes the knowledge of the round t) to arms, i.e., $\pi(\mathscr{H}_t) \in \mathscr{K}$. For both types of policies, we often use the shorter notation $\pi(t)$, where the dependencies on μ or $\mathscr{H}_t$ is implicit.

For a policy π, let $N_{i,t}^{\pi} \triangleq \sum_{s=1}^{t} \mathbb{I}\{\pi(s) = i\}$ be the number of pulls of arm i at the end of round t. The performance of a policy π is measured by the (expected) rewards accumulated over time,

$$J_T(\pi) \triangleq \sum_{t=1}^{T} \mu_{\pi(t)}\left(N_{\pi(t),t-1}\right) = \sum_{i\in\mathscr{K}} \sum_{n=0}^{N_{i,T}^{\pi}-1} \mu_i(n). \tag{1.2}$$

R The cumulative reward depends only on the number of pull of each arm at the horizon T: it does not depend on the specific pulling order of the arms. Hence, two distinct policies with the same pulling allocation at the horizon T, *i.e.* $N_{i,T}^{\pi_1} = N_{i,T}^{\pi_2}$ for all i, have the same cumulative reward.

We notice that $\pi \in \Pi_{\mathrm{L}}$ depends on the (random) history observed over time, and $J_T(\pi)$ is also random for learning policies. The goal of the learning agent is to maximize the expected reward $\mathbb{E}[J_T(\pi)]$. On the contrary, oracle policies do not depend on the (random) history. They can be computed entirely before the start of the game. Hence, finding $\pi^\star \in \arg\max_{\pi\in\Pi_{\mathrm{O}}} J_T(\pi)$ is called the *offline problem*. For a given problem μ, there is a finite number (K^T) of policies, hence the maximum always exists and it could be found by brute-force with infinite computational power.

We set a policy $\pi^\star \in \arg\max_{\pi\in\Pi_{\mathrm{O}}} J_T(\pi)$. Calling $J_T^\star = J_T(\pi^\star)$ the largest cumulative reward achievable, one can measure the regret of any policy (learning or oracle) compared to the optimal one,

$$R_T(\pi) \triangleq J^\star - J_T(\pi). \tag{1.3}$$

Let $N_{i,T}^{\star} \triangleq N_{i,T}^{\pi^{\star}}$ be the number of times that arm i is pulled by the oracle policy $\pi^{\star}$ up to time T (excluded). Using Equation 1.2, we can conveniently rewrite the regret as

$$\begin{aligned} R_T(\pi) &= \sum_{i \in \mathscr{K}} \left(\sum_{n=0}^{N_{i,T}^{\star}-1} \mu_i(n) - \sum_{n=0}^{N_{i,T}^{\pi}-1} \mu_i(n) \right) \\ &= \sum_{i \in \mathrm{UP}} \sum_{n=N_{i,T}^{\pi}}^{N_{i,T}^{\star}-1} \mu_i(n) - \sum_{i \in \mathrm{OP}} \sum_{n=N_{i,T}^{\star}}^{N_{i,T}^{\pi}-1} \mu_i(n), \end{aligned} \tag{1.4}$$

where we define $\mathrm{UP} \triangleq \left\{ i \in \mathscr{K} | N_{i,T}^{\star} > N_{i,T}^{\pi} \right\}$ and likewise $\mathrm{OP} \triangleq \left\{ i \in \mathscr{K} | N_{i,T}^{\star} < N_{i,T}^{\pi} \right\}$ as the sets of arms that are respectively under-pulled and over-pulled by π with respect to the optimal policy.

R 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 **auer2002finite**) are not known to provide any guarantee for our definition of regret. Moreover, for constant $\mu_i(n)$-s, our problem and definition of regret reduce to the one of stationary stochastic bandits.

We give an upperbound on the regret that holds for any policy and will be used in the analysis of all the presented learning policies. First, we upper-bound all the rewards in the first double sum - the underpulls - by their maximum $\mu_T^+(\pi) \triangleq \max_{i \in \mathscr{K}} \mu_i(N_{i,T}^{\pi})$. Indeed, for any overpulls $\mu_i(n_i)$ (with $n_i > N_{i,T}^{\pi}$), we have that

$$\mu_i(n_i) \leq \mu_i(N_{i,T}^{\pi}) \leq \max_{i \in \mathscr{K}} \mu_i(N_{i,T}^{\pi}),$$

where the first inequality follows by the non-increasing property of μ_is; and the second by the defintion of the maximum operator. Second, we notice that there are as many underpulls than overpulls (terms of the second double sum) because there both policies π and $\pi^{\star}$ pull T arms. Notice that this does *not* mean that for each arm i, the number of overpulls equals to the number of underpulls, which cannot happen anyway since an arm cannot be simultaneously underpulled and overpulled. Therefore, we keep only the second double sum,

$$R_T(\pi) \leq \sum_{i \in \mathrm{OP}} \sum_{n=N_{i,T}^{\star}}^{N_{i,T}^{\pi}-1} \left(\mu_T^+(\pi) - \mu_i(n) \right). \tag{1.5}$$

The *online problem* is to find a learning policy which maximizes the expected cumulative reward (or equivalently minimizes the expected regret). In the next sections, we will present the main results of **heidari2016tight**, which has solved the offline problem and the online problem in the absence of noise, and **levine2017rotting**, which has presented the first learning policy with non trivial guarantees for rotting bandits with noise.

1.1.2 The offline problem (heidari2016tight)

We consider the greedy policy π_O (Alg. 1) which at each round selects the arm with the best value.

Algorithm 1 Greedy Oracle π_O (or $\mathscr{A}_0$, **heidari2016tight**)

Require: $\mathscr{K}$, $\{\mu_i\}_{i\in\mathscr{K}}$
1: Initialize $N_i \leftarrow 0$ for all $i \in \mathscr{K}$
2: **for** $t \leftarrow 1,2,\dots$ **do**
3: PULL $i_t \in \arg\max_{i\in\mathscr{K}} \mu_i(N_i)$[a]
4: $N_{i_t} \leftarrow N_{i_t} + 1$
5: **end for**

[a] One can choose the tie break selection rule arbitrarily, e.g. by selecting the arm with the smallest index.

Proposition 1.1.1 — heidari2016tight. For any reward functions $\mu \in \mathscr{L}_L^K$ and any horizon T, $\pi_O \in \arg\max_{\pi\in\Pi_O} J_T(\pi)$.

Proof. At each round t, π_O collects the largest reward that can be available in the future, *i.e.*

$$\forall i \in \mathscr{K},\ \forall n_i \geq N_{i,t}^{\pi_O},\ \mu_{\pi_O(t)}\left(N_{\pi_O(t),t}^{\pi_O}\right) \geq \mu_i\left(N_{i,t}^{\pi_O}\right) \geq \mu_i(n_i).$$

The first inequality is due to the selection rule of the policy; the second is due to the decreasing reward functions.

A direct consequence is that, at round T, π_O has selected the T largest reward sample among the KT possible ones. Therefore, any other policy which would select other reward samples can only have worse or equal cumulative reward. ■

According to Remark 1.1.1, for a given horizon T, all the policies with the same number of pulls of each arm than π_O at round T have the optimal cumulative reward. Yet, we show in the following Proposition that π_O is the only *anytime* optimal policy.

Proposition 1.1.2 Let π such that $\pi(t) \notin \arg\max_{i\in\mathscr{K}} \mu_i(N_{i,t}^{\pi})$.

Then, $J_t(\pi) < \max_{\pi\in\Pi_O} J_t(\pi)$.

Proof. Let $i_t^\star \in \arg\max_{i\in\mathscr{K}} \mu_i(N_{i,t}^{\pi})$. We consider the policy π^+ which selects the same arm than π during the $t-1$ first rounds and selects $i_t^\star$ at round t. Therefore, the two policies π and π^+ collects the same rewards except the last one. Notice that before the last round t, the two policies have the same pulling allocation $N_{j,t-1}^{\pi} = N_{j,t-1}^{\pi^+}$ for all $j \in \mathscr{K}$. Hence, there is only a difference between the two last reward samples,

$$J_t(\pi^+) - J_t(\pi) = \mu_{i_t^\star}(N_{i_t^\star,t-1}^{\pi^+}) - \mu_{\pi(t)}(N_{\pi(t),t-1}^{\pi}) = \mu_{i_t^\star}(N_{i_t^\star,t-1}^{\pi}) - \mu_{\pi(t)}(N_{\pi(t),t-1}^{\pi}) > 0.$$

The inequality follows from $\pi(t) \notin \arg\max_{i\in\mathcal{K}} \mu_i(N_{i,t}^{\pi})$ and $i_t^\star \in \arg\max_{i\in\mathcal{K}} \mu_i(N_{i,t}^{\pi})$. ■

R **Complexity.** We have already highlighted that the offline problem is a computational problem. Indeed, the optimal solution can always be computed by brute force by iterating all the possible policies, i.e. with exponential time complexity per round $\mathcal{O}(K^T)$. By contrast, π_{O} can be computed with space complexity $\mathcal{O}(K)$ and time complexity per round $\mathcal{O}(\log K)$. Indeed, at each round one should find the maximum among K values. Yet, from one round to another, there is only one value which changes : the value of the last selected arm. Thus, one can store a sorted list of the K arm's value and change one element at each round which costs $\mathcal{O}(\log K)$. Then, accessing the first element of the sorted list is a $\mathcal{O}(1)$ operation.

To conclude, π_{O} solves the offline problem in the sense that it provides a cheap way to compute the optimal policy without any knowledge of the horizon T. Interestingly, π_{O} takes the optimal decision by being greedy on the current values. It shows that there is no planning aspect in this problem : the learner never has to sacrifice rewards in the present to get more reward in the future.

1.1.3 The noise-free online problem (heidari2016tight)

In the online problem, the learner does not have access to the current value of the arms. Can they track the best current value using only the observed past values ? **heidari2016tight** first studied the simpler noise-free problem ($\sigma = 0$), where the learner observes the true value of an arm after selecting it (instead of a noisy sample). They suggested the greedy bandit π_{G} (Alg. 2), a policy which selects greedily the arm with the largest last observed value. Indeed, instead of looking at the (unavaible) current values as π_{O}, π_{G} looks at the closest past.

Algorithm 2 Greedy Bandit π_{G} (or $\mathscr{A}_2$, **heidari2016tight**)

Require: $\mathcal{K}$
1: Initialize $\widehat{\mu}_i^1 \leftarrow +\infty$ for all $i \in \mathcal{K}$
2: **for** $t \leftarrow 1, 2, \ldots$ **do**
3: PULL $i_t \in \arg\max_{i\in\mathcal{K}} \widehat{\mu}_i^1$[a]; RECEIVE o_t
4: $\widehat{\mu}_{i_t}^1 \leftarrow o_t$
5: **end for**

[a] One can choose the tie break selection rule arbitrarily, e.g. by selecting the arm with the smallest index.

Proposition 1.1.3 — heidari2016tight. For any problem $\mu \in \mathscr{L}_L^K$ and any horizon T,

$$R_T(\pi_{\mathrm{G}}) \leq (K-1)L.$$

Surprisingly, the worst case regret is upper-bounded by a constant with respect to T.

Proof. We start from Equation 1.5 applied to policy π_{G},

$$R_T(\pi_{\mathrm{G}}) \leq \sum_{i\in \mathrm{OP}} \sum_{n=N^\star_{i,T}}^{N^{\pi_{\mathrm{G}}}_{i,T}-1} \left(\mu_T^+(\pi_{\mathrm{G}}) - \mu_i(n)\right). \tag{1.6}$$

Let $i \in \mathscr{K}$ an arm which is pulled at least twice at the end of the game $N^{\pi_{\mathrm{G}}}_{i,T} \geq 2$. We call $t_i \triangleq \min\{t \leq T \mid N_{i,t} = N_{i,T}\}$ the last round at which i is pulled. For any arm $j \in \mathscr{K}$ pulled at least once at the end of the game $N^{\pi_{\mathrm{G}}}_{j,T} \geq 1$, and for all $n_i \leq N^{\pi_{\mathrm{G}}}_{i,T} - 2$,

$$\mu_i(n_i) \geq \mu_i(N^{\pi_{\mathrm{G}}}_{i,T} - 2) = \mu_i(N^{\pi_{\mathrm{G}}}_{i,t_i-1} - 1) \geq \mu_j(N^{\pi_{\mathrm{G}}}_{j,t_i-1} - 1). \tag{1.7}$$

The first inequality follows by the non-increasing hypothesis on the reward function. The equality follows by definition of t_i. The last inequality is by definition of the policy : at time t_i, π_{G} selects $i \in \arg\max_{j\in\mathscr{K}} \mu_j(N^{\pi_{\mathrm{G}}}_{j,t_i-1} - 1)$, the largest last observed sample.

We choose j such that $\mu_j(N^{\pi_{\mathrm{G}}}_{j,T}) = \mu_T^+(\pi_{\mathrm{G}})$ $\left(\triangleq \max_{j'\in\mathscr{K}} \mu_{j'}(N^{\pi_{\mathrm{G}}}_{j',T})\right)$.
Since $t_i \leq T$, $N^{\pi_{\mathrm{G}}}_{j,t_i-1} - 1 < N^{\pi_{\mathrm{G}}}_{j,T}$. By the rotting assumption,

$$\mu_j(N^{\pi_{\mathrm{G}}}_{j,t_i-1} - 1) \geq \mu_j(N^{\pi_{\mathrm{G}}}_{j,T}) = \mu_T^+(\pi_{\mathrm{G}}). \tag{1.8}$$

Gathering Equations 1.7 and 1.8, we have that

$$\forall n_i \leq N^{\pi_{\mathrm{G}}}_{i,T} - 2,\ \ \mu(n) \geq \mu_T^+(\pi_{\mathrm{G}}). \tag{1.9}$$

Therefore, we can upper-bound all the before last terms in each second sum in Equation 1.6 by zero. Hence,

$$\begin{aligned} R_T(\pi_{\mathrm{G}}) &\leq \sum_{i\in\mathrm{OP}} \left(\mu_T^+(\pi_{\mathrm{G}}) - \mu_i(N^{\pi_{\mathrm{G}}}_{i,T} - 1)\right) \\ &\leq \sum_{i\in\mathrm{OP}} \left(\mu_T^+(\pi_{\mathrm{G}}) - \left(\mu_i(N^{\pi_{\mathrm{G}}}_{i,T} - 2) - L\right)\right) \\ &\leq |\mathrm{OP}| L \\ &\leq (K-1)L \end{aligned}$$

In the second inequality, we used $\mu_i \in \mathscr{L}_L$ (see Definition 1.1.1). The third inequality follows from Equation 1.9. We can conclude by noticing that they are at most $K-1$ overpulled arm. Indeed, there are as many overpulls than underpulls since the two policies $\pi^\star$ and π_{G} both pull $T-1$ sample. Hence, if there is at least one overpulled arm, there is necessary at least one underpulled arm. ■

In the next proposition, we state that this rate is minimax optimal at the first order in $\frac{K}{T}$.

Proposition 1.1.4 — heidari2016tight. For any policy $\pi \in \Pi_{\mathrm{L}}$ and any horizon $T \geq K-1$, there exists a stationary problem $\mu \in \mathscr{S}_L \subset \mathscr{L}_L$ (see Remark 1.1.1) ,

$$R_T(\pi) \geq (K-1)L\left(1 - \frac{K-1}{T}\right).$$

We highlight that our proposition is more precise than the one of **heidari2016tight**. Indeed, while they show only a $\mathcal{O}(K)$ worst case rate, we show that π_G is minimax optimal up to a second order term in $\mathcal{O}\left(\frac{K}{T}\right)$. Moreover, we show that this lower bound holds for the easier stationary problem. Hence, it shows that, without noise, rotting bandits are not harder than stationary ones.

Proof. We consider a set of K problems where

- the first arm has always a constant value equals to $L\left(1-\frac{K-1}{T}\right)$;
- problem $p=1$ has all the other arms with a value 0;
- problem $p \in \{2,\dots,K\}$ has arm p with value L and the other arms $i \in \mathcal{K} \setminus \{1,i\}$ with a value 0.

The learner can distinguish between problem $p \in \{2,\dots,K\}$ and problem 1 only by pulling arm p once. If the learner $\pi \in \Pi_{\mathrm{L}}$ pulls every arm $i \in \{2,\dots,K\}$ once, it suffers on problem 1,

$$R_T^1(\pi) \geq (K-1)L\left(1-\frac{K-1}{T}\right).$$

If there exists an arm $i \in \{2,\dots,K\}$ which is never pulled, π suffers on problem i,

$$R_T(\pi)^i \geq T\left(L - L\left(1-\frac{K-1}{T}\right)\right) = L(K-1).$$

Therefore, we have that for any π, there exists a stationary problem $\mu \in \mathcal{S}_L$ such that,

$$R_T(\pi) \geq (K-1)L\left(1-\frac{K-1}{T}\right)$$

■

R **heidari2016tight** have also studied rested bandits with increasing and concave reward function (without noise).The offline analysis shows that the optimal policy selects always the same arm. This is very different from the rotting case, where the optimal allocation may pull several arms. They suggest an online policy which plays Round-robin on an active set of arms. An arm is excluded from this active set if the optimistic projection of its total available reward untill the end of the game (which can be computed thanks to the concavity assumption) is lower than the pessimistic projection of any other arm (i.e. the arm stays constant). They prove a $o(T)$ regret bound (in the noise-less case !) for this algorithm. While they do not provide a lower bound, it suggests that this problem is harder than the rotting case, where the minimax rate is only in $\mathcal{O}(KL)$.

1.1.4 levine2017rotting : wSWA, a first policy for the noisy problem

Sliding-Window Average (SWA)

When the feedback is noisy ($\sigma > 0$), selecting greedily on the last observed reward may be very risky. Indeed, a sample from an optimal pull could be underestimated by $\mathscr{O}(\sigma)$. π_{G} may not pull this good underestimated arm for a long time, because it only estimates the value of the arm with the last sample. This behaviour may cause a regret of $\mathscr{O}(\sigma T)$ which could be much larger from the noise-free rate $\mathscr{O}(KL)$ depending on the parameters.

levine2017rotting suggested to use the Sliding-Window Average (SWA) policy, a policy which selects the arm with the largest average of its h last sample. Averaging in the presence of noise is a straightforward idea. Yet, it is unclear how the learner should choose h. Before going through the detailed analysis, we give the high-level idea. First, we notice that when $h = 1$, SWA reduces to π_{G}. Indeed, intuitively, the smaller the noise, the less averaging we need. On the one hand, with a window h, the learner should expect to do $\mathscr{O}(h)$ overpulls for an arm which abruptly decay at $N^{\star}_{i,T}$. Indeed, its estimator $\widehat{\mu}_i^h$ will be positively bias during the next h pulls. Hence, the learner may suffer up to $\mathscr{O}(KLh)$ due to this bias. On the other hand, the learner will also take decision based on estimators with variance $\widetilde{\mathscr{O}}(\frac{\sigma}{\sqrt{h}})$ which may cost $\widetilde{\mathscr{O}}(\frac{\sigma T}{\sqrt{h}})$ on the long run. Choosing $h = \widetilde{\mathscr{O}}\left(\frac{\sigma T}{KL}\right)^{2/3}$, we get the regret rate of $\widetilde{\mathscr{O}}\left(L^{1/3}\sigma^{2/3}K^{1/3}T^{2/3}\right)$.

Algorithm 3 SWA (**levine2017rotting**)

Require: $\mathscr{K}$, h
Initialize $\widehat{\mu}_i^h \leftarrow +\infty$ for all $i \in \mathscr{K}$
Initialize $\mathbf{H}(i) \leftarrow []$ for all $i \in \mathscr{K}$
for $t \leftarrow 1, 2, \ldots, Kh$ **do**
    PULL ROUND-ROBIN $i_t \leftarrow t\%h$; RECEIVE o_t
    $\mathbf{H}(i_t) \leftarrow \mathbf{H}(i_t)$.append$(o_t)$
end for
for $t \leftarrow Kh+1, Kh+2, \ldots$ **do**
    PULL $i_t \in \arg\max_{i \in \mathscr{K}} \widehat{\mu}_i^h$[a]; RECEIVE o_t
    $\mathbf{H}(i_t) \leftarrow \mathbf{H}(i_t)$.append$(o_t)$
    if len$(\mathbf{H}(i_t)) \geq h$ **then**
        $\widehat{\mu}_{i_t}^h \leftarrow$ MEAN$(\mathbf{H}(i_t)[-h:])$
    end if
end for

[a] One can choose the tie break selection rule arbitrarily, e.g. by selecting the arm with the smallest index.

R SWA uses a rested sliding-window mechanism. Indeed, the window of arm i slides only when arm i is selected. Notice the difference with the restless sliding-window of SW-UCB (**garivier2011upper-confidence-bound**), which slides for all arms at every round.

Analysis

The analysis of **levine2017rotting** uses the set of bounded decaying function instead of $\mathscr{L}_L$.

Definition 1.1.2 Let $\mathscr{B}_{B,x}$, the set of non-increasing reward functions with bounded amplitude B,

$$\mathscr{B}_{B,x} \triangleq \left\{\mu : \{0,\dots,T-1\} \to [x, x+B] \;\middle|\; \mu(n) \geq \mu(n+1)\right\}.$$

The choice of origin x is not important. Without loss of generality, we will carry the analysis on $\mathscr{B}_B \triangleq \mathscr{B}_{B,0}$.

R We have that $\mathscr{B}_L \subset \mathscr{L}_L$. Hence, any guarantee of any algorithm on $\mathscr{L}_L$ applies on $\mathscr{B}_B$ by setting $L := B$. We also have that $\mathscr{L}_L \subset \mathscr{B}_{LT,-L(T-1)}$. Hence, any guarantee of any algorithm on $\mathscr{B}_{B,x}$ applies on $\mathscr{L}_L$ by setting $B := LT$.

Estimators

For policy π, we define the average of the last h observations of arm i at time t as

$$\widehat{\mu}_i^h(t,\pi) \triangleq \frac{1}{h}\sum_{s=1}^{t-1} \mathbb{1}\left(\pi(s) = i \wedge N_{i,s}^{\pi} > N_{i,t-1}^{\pi} - h\right) o_s \tag{1.10}$$

and the average of the associated means as

$$\overline{\mu}_i^h(t,\pi) \triangleq \frac{1}{h}\sum_{s=1}^{t-1} \mathbb{1}\left(\pi(s) = i \wedge N_{i,s}^{\pi} > N_{i,t-1}^{\pi} - h\right) \mu_i(N_{i,s-1}^{\pi}). \tag{1.11}$$

We notice that $\overline{\mu}_i^h(t,\pi) = \frac{1}{h}\sum_{h'=1}^{h} \mu_i(N_{i,t-1}^{\pi} - h') = \overline{\mu}_i^h(N_{i,t-1}^{\pi})$. With a slight abuse of notation, we will also use $\widehat{\mu}_i^h(N_{i,t}^{\pi}) \triangleq \widehat{\mu}_i^h(t,\pi)$. Indeed, the average of the observations depends on the realization of the noise ε_t at time t. Yet, these h samples of noise are i.i.d. and thus do not perturb the analysis.

A favorable event

Proposition 1.1.5 For a confidence level $\delta_T \triangleq 2T^{-3}$, let

$$\xi_{\text{SWA}} \triangleq \left\{\forall t \in \{Kh+1,\dots,T\}, \forall i \in \mathscr{K}, \forall n \in \{h,\dots,t-1\}, \left|\widehat{\mu}_i^h(n) - \overline{\mu}_i^h(n)\right| \leq c(h,\delta_T)\right\} \tag{1.12}$$

be the event under which all the possible estimates constructed at round t are all accurate up to $c(h,\delta_T) \triangleq \sqrt{2\sigma^2 \log(2/\delta_T)/h}$. Then, for a policy which pulls every arm h times at the beginning (like SWA),

$$\mathbb{P}\left[\overline{\xi_{\text{SWA}}}\right] \leq \frac{K}{T}.$$

Proof. We want to upper bound the probability

$$\mathbb{P}\left[\overline{\xi_{\text{SWA}}}\right] = \mathbb{P}\left[\exists t \in \{Kh+1,\dots,T\}, \exists i \in \mathscr{K}, \exists n \in \{h,\dots,t-1\}, \left|\widehat{\mu}_i^h(n) - \overline{\mu}_i^h(n)\right| > c(h,\delta_T)\right].$$

For $N_{i,t-1}^{\pi_{\text{SWA}}} = n$, we have that,

$$\widehat{\mu}_i^h(n) - \overline{\mu}_i^h(n) = \frac{1}{h}\sum_{s=1}^{t-1} \mathbb{1}\left(i_s = i \,|\, N_{i,s} > n-h\right)\varepsilon_s.$$

By Doob's optional skipping (e.g. see **chow1997probability**, Section 5.3) there exists a sequence of random independent variable $(\varepsilon'_l)_{l\in\mathbb{N}}$, σ^2 sub-Gaussian such that

$$\widehat{\mu}_i^h(n) - \overline{\mu}_i^h(n) = \frac{1}{h}\sum_{s=1}^{t-1} \mathbb{1}\left(i_s = i \,|\, N_{i,s} > n-h\right)\varepsilon_s = \frac{1}{h}\sum_{l=n-h+1}^{n} \varepsilon'_l \triangleq \widehat{\varepsilon}_n^h.$$

Hence,

$$\begin{aligned}
&\mathbb{P}\left[\exists t \in \{Kh+1,\dots,T\}, \exists i \in \mathscr{K}, \exists n \in \{h,\dots,t-1\}, \left|\widehat{\mu}_i^h(n) - \overline{\mu}_i^h(n)\right| > c(h,\delta_T)\right] \\
&\quad = \mathbb{P}\left[\exists t \le T, \exists i \in \mathscr{K}, \exists n \in \{h,\dots,t-1\}, |\widehat{\varepsilon}_n^h| > c(h,\delta_T)\right] \\
&\quad \le \sum_{t=Kh+1}^{T} \sum_{i\in\mathscr{K}} \sum_{n=h}^{t-1} \mathbb{P}\left[|\widehat{\varepsilon}_n^h| > c(h,\delta_T)\right] \\
&\quad \le \frac{KT(T-1)}{2}\cdot\delta_T \\
&\quad \le \frac{K}{T},
\end{aligned}$$

where we used the Chernoff inequality at the before last line and $\delta_T = 2T^{-3}$ at the last one. ■

R Notice that **levine2017rotting** suggests to use $\delta_T = \frac{1}{T^2}$ to recover the same probability $\frac{K}{T}$. They argue that SWA only uses less than KT statistics along a trajectory. Yet, this argument is wrong. Indeed, $N_{i,t}^{\pi_{\text{SWA}}}$ is a random variable which depends on the past observations. When an arm

Regret upper-bound

Proposition 1.1.6 — levine2017rotting. For a problem $\boldsymbol{\mu} \in \mathscr{B}_B^K$, the expected regret of SWA tuned with h is bounded as

$$\mathbb{E}\left[R_T(\pi_{\text{SWA}})\right] \le 2\sigma T \cdot \sqrt{\frac{6\log(T)}{h}} + K(h+1)B$$

Proof. We split the regret on the events ξ_{SWA} and $\overline{\xi_{\text{SWA}}}$,

$$\mathbb{E}\left[R_T(\pi_{\text{SWA}})\right] \leq \mathbb{E}\left[\mathbb{1}\left[\xi_{\text{SWA}}\right] R_T(\pi_{\text{SWA}})\right] + \mathbb{E}\left[\mathbb{1}\left[\overline{\xi_{\text{SWA}}}\right] R_T(\pi_{\text{SWA}})\right].$$

The regret on the unfavorable event $\mathbb{1}\left[\overline{\xi_{\text{SWA}}}\right]$ can be bounded by the maximal regret LT (since $\mu \in \mathscr{B}_B^K$),

$$\mathbb{E}\left[R_T(\pi_{\text{SWA}})\right] \leq \mathbb{E}\left[\mathbb{1}\left[\xi_{\text{SWA}}\right] R_T(\pi_{\text{SWA}})\right] + \mathbb{P}\left[\overline{\xi_{\text{SWA}}}\right] BT.$$

Using Proposition 1.1.5, we get,

$$\mathbb{E}\left[R_T(\pi_{\text{SWA}})\right] \leq \mathbb{E}\left[\mathbb{1}\left[\xi_{\text{SWA}}\right] R_T(\pi_{\text{SWA}})\right] + KB. \tag{1.13}$$

We will now bound the regret on the favorable event,

$$R_T(\pi_{\text{SWA}}|\xi_{\text{SWA}}) \triangleq \mathbb{1}\left[\xi_{\text{SWA}}\right] R_T(\pi_{\text{SWA}})$$

We start from Equation 1.5 applied to policy `SWA`,

$$R_T(\pi_{\text{SWA}}|\xi_{\text{SWA}}) \leq \mathbb{1}\left[\xi_{\text{SWA}}\right] \sum_{i \in \text{OP}} \sum_{n=N^\star_{i,T}}^{N_{i,T}^{\pi_{\text{SWA}}}-1} \left(\mu_T^+(\pi_{\text{SWA}}) - \mu_i(n)\right). \tag{1.14}$$

The remaining of the proof is similar to the proof of Proposition 1.1.3 about algorithm π_{G}. Instead of showing that the before last terms in the sums are equals to zeros, we will show that the terms before h last one are less than $2c(h,\delta_T)$. Let $i \in \mathscr{K}$ an arm which is pulled at least $h+1$ times at the end of the game $N_{i,T}^{\pi_{\text{SWA}}} \geq h+1$. We call $t_i \triangleq \min\left\{t \leq T \mid N_{i,t}^{\pi_{\text{SWA}}} = N_{i,T}^{\pi_{\text{SWA}}}\right\}$ the last round at which i is pulled. For any arm $j \in \mathscr{K}$ pulled at least h at the end of the game $N_{j,T}^{\pi_{\text{SWA}}} \geq h$, and for all $n_i \leq N_{i,T}^{\pi_{\text{SWA}}} - (h+1)$,

$$\begin{aligned}
\mu_i(n_i) &\geq \mu_i(N_{i,T}^{\pi_{\text{SWA}}} - (h+1)) \\
&\geq \overline{\mu}_i^h(N_{i,t_i-1}^{\pi_{\text{SWA}}}) \\
&\geq \widehat{\mu}_i^h(N_{i,t_i-1}^{\pi_{\text{SWA}}}) - c(h,\delta_T) \\
&\geq \widehat{\mu}_j^h(N_{j,t_i-1}^{\pi_{\text{SWA}}}) - c(h,\delta_T) \\
&\geq \overline{\mu}_j^h(N_{j,t_i-1}^{\pi_{\text{SWA}}}) - 2c(h,\delta_T).
\end{aligned} \tag{1.15}$$

The first inequality follows by the non-increasing hypothesis on the reward function. The second inequality is because $\overline{\mu}_i^h(N_{i,t_i-1}^{\pi_{\text{SWA}}})$ is the average of h reward sample of arm i after the $N_{i,T}^{\pi_{\text{SWA}}} - (h+1)$-th. The third and fifth one use the concentration of all the possible estimates on the event ξ_{SWA}. The fourth inequality follows by definition of the policy : at time t_i, π_{SWA} selects $i \in \arg\max_{j \in \mathscr{K}} \widehat{\mu}_j^h(N_{j,t_i-1}^{\pi_{\text{SWA}}})$, the largest last observed sample.

We choose j such that $\mu_j(N_{j,T}^{\pi_{\text{SWA}}}) = \mu_T^+(\pi_{\text{SWA}})$ $\left(\triangleq \max_{j' \in \mathcal{K}} \mu_{j'}(N_{j',T}^{\pi_{\text{SWA}}})\right)$.
Since $t_i \leq T$, by the rotting assumption,

$$\overline{\mu}_j^h(N_{j,t_i-1}^{\pi_{\text{SWA}}}) \geq \mu_j(N_{j,T}^{\pi_{\text{SWA}}}) = \mu_T^+(\pi_{\text{SWA}}). \tag{1.16}$$

Gathering Equations 1.15 and 1.16, we have that

$$\forall n_i \leq N_{i,T}^{\pi_{\text{SWA}}} - (h+1), \quad \left(\mu_T^+(\pi_{\text{SWA}}) - \mu_i(n_i)\right) \leq 2c(h, \delta_T). \tag{1.17}$$

Therefore, in Equation 1.14, we can split the sum on $N_{i,T}^{\pi_{\text{SWA}}} - h$. Hence,

$$\begin{aligned}
R_T(\pi_{\text{SWA}}|\xi_{\text{SWA}}) \leq & \mathbb{1}\left[\xi_{\text{SWA}}\right] \sum_{i \in \text{OP}} \sum_{n=N_{i,T}^\star}^{N_{i,T}^{\pi_{\text{SWA}}}-1} \left(\mu_T^+(\pi_{\text{SWA}}) - \mu_i(n)\right) \\
= & \mathbb{1}\left[\xi_{\text{SWA}}\right] \sum_{i \in \text{OP}} \sum_{n=N_{i,T}^\star}^{N_{i,T}^{\pi_{\text{SWA}}}-(h+1)} \left(\mu_T^+(\pi_{\text{SWA}}) - \mu_i(n)\right) \\
& + \mathbb{1}\left[\xi_{\text{SWA}}\right] \sum_{i \in \text{OP}} \sum_{n=N_{i,T}^{\pi_{\text{SWA}}}-h}^{N_{i,T}^{\pi_{\text{SWA}}}-1} \left(\mu_T^+(\pi_{\text{SWA}}) - \mu_i(n)\right) \\
\leq & 2Tc(h, \delta_T) + KhB.
\end{aligned} \tag{1.18}$$

In the last inequality, we used Equation 1.17 and that there is less than T overpulls in the first sums. We also use $\mu \in \mathcal{B}_B$ to bound each term in the second sum by B. Finally, we can conclude by plugging Equation 1.18 in Equation 1.13 and by using the definition of $c(h, \delta_T)$ and $\delta_T = 2T^{-3}$ in Proposition 1.1.5,

$$\mathbb{E}\left[R_T(\pi_{\text{SWA}})\right] \leq 2\sigma T \cdot \sqrt{\frac{6\log(T)}{h}} + K(h+1)B$$

■

Corollary 1.1.7 — levine2017rotting. For C such that $h := \left\lceil C\left(\frac{\sigma T}{KB}\right)^{2/3} (6\log(T))^{1/3} \right\rceil$,

$$R_T(\pi_{\text{SWA}}) \leq \left(\frac{2}{C^{1/2}} + C\right) \left(6\sigma^2 BKT^2 \log(T)\right)^{1/3} + 2KB.$$

Hence, if the learner knows T and the ratio $\frac{\sigma}{B}$, they can set $h := \left\lceil \left(\frac{\sigma T}{KB}\right)^{2/3} (6\log(T))^{1/3} \right\rceil$ (i.e. $C = 1$) and be guaranteed to perform

$$R_T(\pi_{\text{SWA}}) \leq 6\left(\sigma^2 BKT^2 \log(T)\right)^{1/3} + 2KB.$$

R We highlighted in Remark 1.1.4 that we have to use tighter confidence level δ_T in the analysis that what **levine2017rotting** suggest. It slightly impacts the theoretical optimal choice of the window as they recommand $h := \left\lceil \left(\frac{\sigma T}{KB}\right)^{2/3} \left(4\log\left(\sqrt{2}T\right)\right)^{1/3} \right\rceil$.

Empirical evaluation of the anytime version wSWA

The theoretical window choice require the knowledge of the horizon T, the subgaussian parameter σ and the reward range B (or at least the ratio $\frac{B}{\sigma}$). **levine2017rotting** suggest wSWA, which wraps SWA with the doubling trick. The algorithm is initialized with a first (small) guess of the horizon. When the horizon is reached, the algorithm is fully reinitialized with a doubled horizon. This is a classic trick in the litterature : it is known to recover the problem-independent rate of a given algorithm (with a worse constant factor), but the empirical performance is often significantly reduced (**besson2018**). In the case of wSWA, the doubling trick erases all the history $\mathbf{H}_t$ and increases the window. In Algorithm 4, we reproduce the version suggested by **levine2017rotting** (without the small modification of the tuning h) .

Algorithm 4 wSWA (**levine2017rotting**)

Require: α, σ, $T_0 \leftarrow 1$
$T \leftarrow T_0$
$h \leftarrow \left\lceil \alpha \left(\frac{4\sigma T}{K}\right)^{2/3} \left(\log\left(\sqrt{2}T\right)\right)^{1/3} \right\rceil$
for $t \leftarrow 1, 2, \ldots, T$ **do**
    RUN SWA (h)
end for
CLEAN SWA'S MEMORY
wSWA (α, σ, $2T_0$)

We notice that the parameter α of wSWA hide the dependency in B. Indeed, the best theoretical tuning corresponds to $\alpha := (2B)^{-2/3}$. In their experimental section, **levine2017rotting** select $\alpha := 0.2$ by grid-search on one problem. Yet, the reader should not forget that the tuning of α is dependent on B, and more generally on which $\boldsymbol{\mu} \in \mathscr{B}_B$.

1.1.5 Open problems

Minimax rate

We report existing regret bounds for two special cases. First, in Proposition 1.1.4, **heidari2016tight** show that in the absence of noise, the regret is lower bounded by $\mathscr{O}(KL)$. Second, we recall the minimax regret lower bound for stochastic stationary bandits.

Proposition 1.1.8 **auer2002nonstochastic** 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.

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. For reward functions in $\mathscr{B}_B$, SWA is guaranteed to achieve $\mathscr{O}\left(T^{2/3}\right)$ regret rate. Yet, **levine2017rotting** do not provide a lower bound while they suggest it could be an interesting future work direction.

Problem-dependent rate

SWA starts by pulling every arm h times. It means that even for simple stationary problem with large difference $\Delta_i > \sigma$ between suboptimal and optimal arms, SWA does $h = \mathscr{O}\left(T^{2/3}\right)$ mistakes per suboptimal arms which is much more than the standard $\mathscr{O}\left(\frac{\sigma \log(T)}{\Delta_i^2}\right)$.

More generally, it is an open-question whether it is possible to get problem-dependent guarantees - which depends on the values $\mu_i(n)$ - while keeping

Agnostic algorithm

SWA requires the knowledge of the horizon T, the subgaussian parameter σ and the reward range B to tune the window h. We showed empirically that the doubling trick leads to large regret increases at each restart. We also showed that the tuning of h

Global budget or Budget per round

The guarantee

1.2 FEWA and RAW-UCB : Two adaptive window algorithms

1.2.1 Towards adaptive windows

Since the expected rewards μ_i change from one pull to another, the main difficulty in the rested 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.

SWA chooses a window which balances the cost due to variance and the cost due to bias.

A favorable event for adaptive windows

Proposition 1.2.1 For any round t and confidence $\delta_t \triangleq 2t^{-\alpha}$, let

$$\xi_t^\alpha \triangleq \left\{ \forall i \in \mathscr{K},\ \forall n \leq t-1,\ \forall h \leq n, \left| \widehat{\mu}_i^h(n) - \overline{\mu}_i^h(n) \right| \leq c(h, \delta_t) \right\} \tag{1.19}$$

be the event under which the estimates at round t are all accurate up to $c(h,\delta_t) \triangleq \sqrt{2\sigma^2 \log(2/\delta_t)/h}$. Then, for a policy π which pulls each arms once at the beginning, and for all $t > K$,

$$\mathbb{P}\left[\overline{\xi_t^{\alpha}}\right] \leq \frac{Kt^2\delta_t}{2} = Kt^{2-\alpha}\,.$$

Proof. We want to upper bound the probability

$$\mathbb{P}\left[\overline{\xi_t^{\alpha}}\right] = \mathbb{P}\left[\exists i \in K, \exists n \leq t-1, \exists h \leq n, \left|\widehat{\mu}_i^h(n) - \overline{\mu}_i^h(n)\right| > c(h,\delta_t)\right].$$

Following the same argument than in Proposition 1.1.5, there exists a sequence of random independent variable $(\varepsilon'_l)_{l\in\mathbb{N}}$, σ^2 sub-Gaussian such that for $\widehat{\varepsilon}_n^h \triangleq (1/h)\sum_{l=n-h+1}^{n}\varepsilon'_l$ we get

$$\begin{aligned}
&\mathbb{P}\left[\exists n \leq t-1, \exists h \leq n, \left|\widehat{\mu}_i^h(t-1,\pi) - \overline{\mu}_i^h(t-1,\pi)\right| > c(h,\delta_t)\right] \\
&\qquad = \mathbb{P}\left[\exists n \leq t-1, \exists h \leq n, |\widehat{\varepsilon}_n^h| > c(h,\delta_t)\right] \\
&\qquad \leq \sum_{n=1}^{t-1}\sum_{h=1}^{n}\mathbb{P}\left[|\widehat{\varepsilon}_n^h| > c(h,\delta_t)\right] \\
&\qquad \leq \frac{t(t-1)}{2}\cdot\delta_t\,,
\end{aligned}$$

where we used the Chernoff inequality in the last line. Thus, a union bound over the arms allows us to conclude that

$$\mathbb{P}\left[\overline{\xi_t^{\alpha}}\right] \leq \frac{K\delta_t t^2}{2}.$$

■

R Compared to the unique favorable event we used for SWA (see Equation 1.12), we use a favorable event for each round t. It will be helpful to obtain anytime guarantees for our algorithms. Moreover, ξ_t^{α} control the deviation of any statistic $\widehat{\mu}_i^h(n)$ for any possible h, i and n. This is different from ξ_{SWA} which uses a fixed h.

1.2.2 FEWA: Filtering on expanding window average

In Alg. 5, 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 6 `FILTER`

Require: $\mathscr{K}_h$, h, α, σ

$c(h,\delta_t) \leftarrow \sqrt{2\alpha\sigma^2\log(t)/h}$
$\widehat{\mu}^h_{\max} \leftarrow \max_{i\in\mathscr{K}_h} \widehat{\mu}^h_i$
for $i \in \mathscr{K}_h$ **do**
    $\Delta_i \leftarrow \widehat{\mu}^h_{\max} - \widehat{\mu}^h_i$
    if $\Delta_i \leq 2c(h,\delta_t)$ **then**
        add i to $\mathscr{K}_{h+1}$
    end if
end for

Ensure: $\mathscr{K}_{h+1}$

Algorithm 5 `FEWA`

Require: $\mathscr{K}$, σ, α

for $t \leftarrow 1,2,\dots,K$ **do** ▷ *Pull each arm once*
    PULL $i_t \leftarrow t$; RECEIVE o_t
    $N_{i_t} \leftarrow 1$
    $\left\{\widehat{\mu}^h_{i_t}\right\}_h \leftarrow$ `UPDATE`$(\left\{\widehat{\mu}^h_{i_t}\right\}_h, o_t)$
end for
for $t \leftarrow K+1, K+2, \dots$ **do**
    $h \leftarrow 1$ ▷ *initialize bandwidth*
    $\mathscr{K}_1 \leftarrow \mathscr{K}$ ▷ *initialize with all the arms*
    $i_t \leftarrow$ `none`
    while i_t is `none` **do**
        $\mathscr{K}_{h+1} \leftarrow$ `FILTER`$(\mathscr{K}_h, h, \alpha, \sigma)$
        $h \leftarrow h+1$
        if $\exists i \in \mathscr{K}_h$ such that $N_{i_t} = h$ **then**
            PULL $i_t \in \{i \in \mathscr{K}_h | N_{i_t} = h\}$[a]; RECEIVE o_t
        end if
    end while
    $N_{i_t} \leftarrow N_{i_t} + 1$
    $\left\{\widehat{\mu}^h_{i_t}\right\}_h \leftarrow$ `UPDATE`$(\left\{\widehat{\mu}^h_{i_t}\right\}_h, o_t)$
end for

[a] One can choose the tie break selection rule arbitrarily, e.g. by selecting the arm with the smallest index.

We first describe the subroutine `FILTER` in Alg. 6, which receives a set of active arms $\mathscr{K}_h$, a window h, a confidence bound tuning parameter α and the subgaussian parameter σ as input and returns an updated set of arms $\mathscr{K}_{h+1}$. For each arm $i \in \mathscr{K}_h$ (that has all been pulled $n \geq h$ times), the algorithm has stored an estimate $\widehat{\mu}^h_i$ that averages the h most recent rewards observed from i. The subroutine `FILTER` discards all the arms whose mean estimate (built with window h) from $\mathscr{K}_h$ 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. 1.2.1).

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 13 is met. As a result, $\mathscr{K}_{h+1}$ 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 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 $c(h,\delta_t)$ is large for small h (i.e., as few samples are used in constructing $\widehat{\mu}_i^h$, the estimation error may be high). Thus, FEWA keeps refining $\mathscr{K}_h$ 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 $\mathscr{K}_h$ (i.e., Line 13). At this point, increasing h would not bring any additional evidence that could refine $\mathscr{K}_h$ further (recall that $\widehat{\mu}_i^h$ is not defined for $h > N_i$). Finally, FEWA selects the active arm i_t whose number of samples matches the current window, i.e., the least pulled arm in $\mathscr{K}_h$. The set of available rewards and the number of pulls are then updated accordingly.

Core guarantee on the favorable event

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 $\{N_{i,t-1}^{\pi_F}\}_i$ the greedy (oracle) policy would select an arm

$$i_t^\star\left(\left\{N_{i,t-1}^{\pi_F}\right\}_i\right) \in \arg\max_{i\in\mathscr{K}} \mu_i\left(N_{i,t-1}^{\pi_F}\right).$$

We recall that $\mu_t^+(\pi_F) \triangleq \max_{i\in\mathscr{K}} \mu_i(N_{i,t-1}^{\pi_F})$ the reward that could be obtained by pulling $i_t^\star$ at round t. While FEWA cannot directly match the performance of the oracle arm, the following lemma guarantees that the past performance of the selected arm is close enough compared to the current best arm value.

Lemma 1.2.2 For FEWA tuned with α, on the favorable event ξ_t^α, if an arm i passes through a filter of window h at round t, i.e., $i \in \mathscr{K}_h$, then the average of its h last pulls satisfies

$$\overline{\mu}_i^h(N_{i,t-1}^{\pi_F}) \geq \mu_t^+(\pi_F) - 4c(h,\delta_t). \tag{1.20}$$

Therefore, at round t on favorable event ξ_t^α, if arm i_t is selected by FEWA (α), for any $h \leq N_{i,t-1}^{\pi_F}$, the average of its h last pulls cannot deviate significantly from the best available arm at that round, i.e.,

$$\overline{\mu}_{i_t}^h(N_{i,t-1}^{\pi_F}) \geq \mu_t^+(\pi_F) - 4c(h,\delta_t).$$

Proof. We will proove this property for a more general rotting feedback model than the rested rotting one presented in Equation 1.1. If arm i is selected at round t, the learner π receives,

$$o_t \triangleq \mu_{i,t} + \varepsilon_t \text{ with } \mathbb{E}[\varepsilon_t|\mathscr{H}_t] = 0 \text{ and } \forall\lambda\in\mathbb{R},\ \mathbb{E}\left[e^{\lambda\varepsilon_t}\right] \leq e^{\frac{\sigma\lambda^2}{2}},$$

with $\{\mu_{i,t}\}_{t\leq T}$ a non-increasing sequence. We do not specify how the reward is rotting, while it was assumed in Equation 1.1 that the reward function was evolving with the number of pull $N_{i,t-1}$ of arm i at round t. With this reward model, we cannot use $\overline{\mu}_i^h(N_{i,t-1}^\pi)$ to refer to $\overline{\mu}_i^h(t,\pi)$, the average of the h last means associated to arm i (see the definition in Equation **??** and the following remark). We will use this more general proof in the next Chapter**??**.

Let $i \in \mathscr{K}_h$ be an arm that passed a filter of window h at round t. First, we use the confidence bound for the estimates and we pay the cost of keeping all the arms up to a distance $2c(h,\delta_t)$ of $\widehat{\mu}_{\max,t}^h \triangleq \max_{j\in\mathscr{K}_h} \overline{\mu}_i^h(t,\pi_{\mathrm{F}})$,

$$\overline{\mu}_i^h(t,\pi_{\mathrm{F}}) \geq \widehat{\mu}_i^h(t,\pi_{\mathrm{F}}) - c(h,\delta_t) \geq \widehat{\mu}_{\max,t}^h - 3c(h,\delta_t) \geq \max_{j\in\mathscr{K}_h} \overline{\mu}_j^h(t,\pi_{\mathrm{F}}) - 4c(h,\delta_t), \quad (1.21)$$

where in the last inequality, we used that for all $j \in \mathscr{K}_h$,

$$\widehat{\mu}_{\max,t}^h \geq \widehat{\mu}_j^h(t,\pi_{\mathrm{F}}) \geq \overline{\mu}_j^h(t,\pi_{\mathrm{F}}) - c(h,\delta_t).$$

Second, we call $t_{i,t} < t$ the last round at which arm i was selected. Since the means of arms are decaying, we know that

$$\begin{aligned}
\mu_t^+(\pi_{\mathrm{F}}) &\triangleq \mu_{i_t^\star,t} \\
&\leq \mu_{i_t^\star,t_{i,t}} = \overline{\mu}_i^1(t,\pi_{\mathrm{F}}) \\
&\leq \max_{j\in\mathscr{K}} \overline{\mu}_j^1(t,\pi_{\mathrm{F}}) = \max_{j\in\mathscr{K}_1} \overline{\mu}_j^1(t,\pi_{\mathrm{F}}).
\end{aligned} \quad (1.22)$$

Third, we show that the largest average of the last h' means of arms in $\mathscr{K}_{h'}$ is increasing with h',

$$\forall h' \leq h, \max_{j\in\mathscr{K}_{h'+1}} \overline{\mu}_j^{h'+1}(t,\pi_{\mathrm{F}}) \geq \max_{j\in\mathscr{K}_{h'}} \overline{\mu}_j^{h'}(t,\pi_{\mathrm{F}}).$$

To show the above property, we remark that thanks to our selection rule, the arm that has the largest average of means, always passes the filter. Formally, we show that $\arg\max_{j\in\mathscr{K}_{h'}} \overline{\mu}_j^{h'}(t,\pi_{\mathrm{F}}) \subseteq \mathscr{K}_{h'+1}$. Let $i_{\max}^{h'} \in \arg\max_{j\in\mathscr{K}_{h'}} \overline{\mu}_j^{h'}(t,\pi_{\mathrm{F}})$. Then, for such $i_{\max}^{h'}$, we have

$$\widehat{\mu}_{i_{\max}^{h'}}^{h'}(t,\pi_{\mathrm{F}}) \geq \overline{\mu}_{i_{\max}^{h'}}^{h'}(t,\pi_{\mathrm{F}}) - c(h',\delta_t) \geq \overline{\mu}_{\max,t}^{h'} - c(h',\delta_t) \geq \widehat{\mu}_{\max,t}^{h'} - 2c(h',\delta_t),$$

where the first and the third inequality are due to concentration of the estimates on ξ_t^α, while the second one is due to the definition of $i_{\max}^{h'}$.

Since the arms are decaying, the average of the last $h'+1$ mean values for a given arm is always greater than the average of the last h' mean values and therefore,

$$\max_{j\in\mathscr{K}_{h'}} \overline{\mu}_j^{h'}(t,\pi_{\mathrm{F}}) = \overline{\mu}_{i_{\max}^{h'}}^{h'}(t,\pi_{\mathrm{F}}) \leq \overline{\mu}_{i_{\max}^{h'}}^{h'+1}(t,\pi_{\mathrm{F}}) \leq \max_{j\in\mathscr{K}_{h'+1}} \overline{\mu}_j^{h'+1}(t,\pi_{\mathrm{F}}), \quad (1.23)$$

because $i^{h'}_{\max} \in \mathscr{K}_{h'+1}$. Gathering Equations 1.21, 1.22, and 1.23 leads to the first claim of the lemma,

$$\begin{aligned}\overline{\mu}^h_i(t,\pi_F) &\overset{(1.21)}{\geq} \max_{j\in\mathscr{K}_h} \overline{\mu}^h_j(t,\pi_F) - 4c(h,\delta_t)\\ &\overset{(1.23)}{\geq} \max_{j\in\mathscr{K}_1} \overline{\mu}^1_j(t,\pi_F) - 4c(h,\delta_t)\\ &\overset{(1.22)}{\geq} \mu_t^+(\pi_F) - 4c(h,\delta_t).\end{aligned}$$

To conclude, we remark that if i is pulled at round t, then by the condition at Line 13 of Algorithm 5, it means that i passes through all the filters from $h=1$ up to $N^{\pi_F}_{i,t-1}$. Therefore, for all $h \leq N^{\pi_F}_{i,t-1}$,

$$\overline{\mu}^h_i(t,\pi_F) \geq \mu_t^+(\pi_F) - 4c(h,\delta_t). \tag{1.24}$$

■

1.2.3 RAW-UCB : Rotting Adaptive Window Upper Confidence Bound

Algorithm 7 RAW-UCB

Require: $\mathscr{K}$, σ, α

for $t \leftarrow 1,2,\dots,K$ **do** ▷ *Pull each arm once*
  PULL $i_t \leftarrow t$; RECEIVE o_t
  $N_{i_t} \leftarrow 1$
  $\left\{\widehat{\mu}^h_{i_t}\right\}_h \leftarrow$ UPDATE$(\left\{\widehat{\mu}^h_{i_t}\right\}_h, o_t)$
end for
for $t \leftarrow K+1, K+2, \dots$ **do**
  PULL $i_t \in \arg\max_i \min_{h\leq N_i} \left(\widehat{\mu}^h_i + c(h,\delta_t)\right)$[a]; RECEIVE o_t ▷ *cf.* (1.25);
  $N_{i_t} \leftarrow N_{i_t} + 1$
  $\left\{\widehat{\mu}^h_{i_t}\right\}_h \leftarrow$ UPDATE$(\left\{\widehat{\mu}^h_{i_t}\right\}_h, o_t)$
end for

[a] One can choose the tie break selection rule arbitrarily, e.g. by selecting the arm with the smallest index.

We will study a single class of policies which select at each round t the arm with the maximal index of the form

$$\mathrm{ind}(i,t,\delta_t) \triangleq \min_{h\leq N_{i,t-1}} \left(\widehat{\mu}^h_i(N_{i,t-1}) + c(h,\delta_t)\right) \quad \text{with} \quad \delta_t \triangleq \frac{2}{t^\alpha}. \tag{1.25}$$

We set and call this algorithm Rotting Adaptive Window UCB (RAW-UCB). There is a bias-variance trade-off for the window choice: more variance for smaller size of the window h and more bias for larger h. The goal of RAW-UCB is to adaptively select the right window to compute the tightest UCB. RAW-UCB uses the indexes of UCB1 computed on all the slices of each arm's history which include the last pull. When the rewards are rotting, all these

indexes are upper confidence bounds on the *next value*. Thus, RAW-UCB simply selects the tightest (minimum) one as index of the arm: it is a pure UCB-index algorithm. By contrast, when reward can increase, the learner can only derive upper-confidence bound on past values which are loosely related to the next value. Hence, all the UCB-index algorithms in the restless non-stationary literature need to add change-detection sub-routine, active random exploration or passive forgetting mechanism.

Core guarantee on the favorable event

Lemma 1.2.3 At round t, on favorable event ξ_t^α, if arm i_t is selected by RAW-UCB (α), for any $h \leq N_{i,t-1}$, the average of its h last pulls cannot deviate significantly from the best available arm at that round, i.e.,

$$\overline{\mu}_{i_t}^{h}(N_{i_t,t-1}) \geq \max_{i\in\mathscr{K}} \mu_i(N_{i,t-1}) - 2c(h,\delta_t)\,.$$

This lemma is comparable with Lemma 1.2.2 about the algorithm FEWA. Yet, RAW-UCB has tighter guarantees than FEWA (2 versus 4 confidence bands), which is the benefit of upper confidence bounds index policies over confidence bound filtering policies.

Proof. Like for Lemma 1.2.2 (see its proof), our proof is done in a more general rotting framework that can be used in the next chapter.

We denote by $i_t^\star \in \arg\max_{i\in\mathscr{K}} \mu_i(t, N_{i,t-1})$, a best available arm at time t and

$$h_{i,t}^{\min} \in \operatorname*{arg\,min}_{h\leq N_{i,t-1}} \widehat{\mu}_i^h(t-1,\pi) + c(h,\delta_t),$$

a window which minimizes RAW-UCB index at time t for arm i. Hence, because the reward functions are non-increasing, we know that

$$\mu_{i_t^\star}(t, N_{i_t^\star,t-1}) \leq \overline{\mu}_{i_t^\star}^{1}(t-1,\pi) \leq \cdots \leq \overline{\mu}_{i_t^\star}^{h_{i_t^\star,t}^{\min}}(t-1,\pi)\cdot$$

On the high-probability event ξ_t, we know that the true average of the means cannot deviate significantly from the average of the observed quantity,

$$\overline{\mu}_{i_t^\star}^{h_{i_t^\star,t}^{\min}}(t-1,\pi) \leq \widehat{\mu}_{i_t^\star}^{h_{i_t^\star,t}^{\min}}(t-1,\pi) + c(h_{i_t^\star,t}^{\min},\delta_t)\,.$$

We know that the selected arm i_t at time t has the largest index, hence,

$$\widehat{\mu}_{i_t^\star}^{h_{i_t^\star,t}^{\min}}(t-1,\pi) + c(h_{i_t^\star,t}^{\min},\delta_t) \leq \widehat{\mu}_{i_t}^{h_{i_t,t}^{\min}}(t-1,\pi) + c(h_{i_t,t}^{\min},\delta_t).$$

From $h_{i,t}^{\min}$ definition, we know that this quantity is below any upper-confidence bound for any other window h

$$\widehat{\mu}_{i_t}^{h_{i_t,t}^{\min}}(t-1,\pi) + c(h_{i_t,t}^{\min},\delta_t) \leq \widehat{\mu}_{i_t}^{h}(t-1,\pi) + c(h,\delta_t).$$

Finally, using again the concentration of the average on the ξ_t^α,

$$\widehat{\mu}_{i_t}^h(t-1,\pi) + c(h,\delta_t) \leq \overline{\mu}_{i_t}^h(t-1,\pi) + 2c(h,\delta_t)\,.$$

Hence, putting all the equations together, we can write

$$\overline{\mu}_{i_t}^h(t-1,\pi) \geq \max_{i\in\mathcal{K}} \mu_i(t,N_{i,t-1}) - 2c(h,\delta_t).$$

■

1.3 Regret Analysis

In the last section, we presented two algorithms which have very different behaviours. Yet, they show two main similarities. First, for each arm they compute several statistics $\widehat{\mu}_i^h(N_{i,t-1})$ for different windows $h \leq N_{i,t-1}$. Second, on the same favorable events ξ_t^α (on which all these aforementioned statistics are well concentrated around their means, see Prop. 1.2.1), we have shown that both algorithms share a guarantee with similar shape (see Lemmas 1.2.2 and 1.2.3). We will see that these preliminary results are the only characterization we need. Therefore, we define $C_{\pi_R} = 2$ and $C_{\pi_F} = 4$, the constant associated to each algorithm in their respective Lemma.

We first give problem-independent regret bound for `FEWA` and `RAW-UCB` and sketch its proof in Subsection 1.3.1. Then, we discuss problem-dependent guarantees in Subsection 1.3.2. Finally, we give the detailed analysis in Subsection 1.3.3.

1.3.1 Problem-independent bound

Theorem 1.3.1 For any rotting bandit scenario with means $\{\mu_i\}_i \in \mathscr{L}_L^K$ and any time horizon T, $\pi \in \{\pi_R, \pi_F\}$ run with $\alpha \geq 5$ suffers an expected regret of

$$\mathbb{E}[R_T(\pi)] \leq C_\pi \sqrt{2\alpha\sigma^2 \log(T)} \left(\sqrt{KT} + K\right) + 3KL\,.$$

Comparison to levine2017rotting

The regret of `SWA` is bounded by $\widetilde{\mathscr{O}}(B^{1/3}K^{1/3}T^{2/3})$ for bounded rotting functions in $\mathscr{B}_B$. According to Remark 1.1.4, the regret guarantee translate in $\mathscr{O}(T)$ for rotting functions in $\mathscr{L}_L$. Thus, according to its original analysis, `SWA` may not be able to learn for our general setting. On the other hand, we could use `FEWA` or `RAW-UCB` with rotting functions in $\mathscr{B}_B$ and recover the same regret bound with $L := B$. In this case, our two algorithms suffer a regret of $\widetilde{\mathscr{O}}(\sqrt{KT})$, thus significantly improving over `SWA`.

The improvement is mostly due to the fact that `FEWA` and `RAW-UCB` use adaptive window mechanisms to smoothly track changes in the value of each arm. Indeed, `SWA` 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. According to

Proposition 1.1.6, this fixed window trades off between the cost of biased estimates $\mathcal{O}(KBh)$ - for scenarios where the arms abruptly decay and their values are overestimated during at most h rounds - and the cost of variance of the estimators $\widetilde{\mathcal{O}}\left(\frac{\sigma T}{\sqrt{h}}\right)$ - for scenarios where the arms keep their value close to each other for $\mathcal{O}(T)$ rounds. In Theorem 1.3.1, the regret of FEWA and RAW-UCB is also bounded by an additive decomposition between the terms depending on the noise level σ and the terms depending on the rotting level L. Yet, adaptive window algorithms do not need to trade-off: their regret is bounded by $\mathcal{O}(KL)+\widetilde{\mathcal{O}}\left(\sigma\sqrt{KT}\right)$. It evidence that our algorithms are able to take decision based on a relevant $h \in \left\{1,\dots,N_{i,t-1}\right\}$ depending on the scenarios.

Last, our algorithms are anytime and agnostic to L (or B), while the tuning of SWA requires to know B and T (or to resort to a doubling trick, which performs poorly in practice).

Comparison to stationary stochastic bandits

The regret of FEWA and RAW-UCB match 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, our analysis 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. 1.3.1 are realized with constant functions, which is the standard stochastic bandits, for which MOSS-like (**audibert2009minimax**) 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.

1.3.2 Problem-dependent guarantees

Since our setting generalizes the stationary stochastic bandit setting, a natural question is whether we pay any price for this generalization. While the result of **levine2017rotting** suggested that learning in rotting bandits could be more difficult, in Thm. 1.3.1 we actually proved that FEWA and RAW-UCB nearly match the problem-independent regret rate $\widetilde{\mathcal{O}}(\sqrt{KT})$. We may wonder whether this is true for the *problem-dependent* regret as well.

R Consider a stationary stochastic bandit setting with expected rewards $\{\mu_i\}_i$ and $\mu_\star \triangleq \max_i \mu_i$. For $\pi \in \{\pi_{\mathrm{F}}, \pi_{\mathrm{R}}\}$, on the favorable event ξ_t^α with $\delta_t \geq 1/T^\alpha$, we can apply Lemmas 1.2.2 or 1.2.3 at the last time arm i is pulled (*i.e.* after $N_{i,T}^\pi - 1$ pulls) for $h = N_{i,T}^\pi - 1$,

$$\mu_\star - \mu_i \leq C_\pi c\left(N_{i,T}^\pi - 1, \delta_t\right) = C_\pi \sqrt{\frac{2\alpha\sigma^2\log(T)}{N_{i,T}^\pi - 1}}$$

$$\text{or equivalently, } N_{i,T}^\pi \leq 1 + \frac{2\alpha C_\pi^2 \sigma^2 \log(T)}{(\mu_\star - \mu_i)^2}. \tag{1.26}$$

Therefore, for $\alpha > 4$[1], our algorithms match the lower bound of **lai1985asymptotically** up to a constant factor αC_π^2.

With a similar reasonning than in Remark 1.3.2, we can show a similar bound on the number of overpulls $h_{i,T}^\pi$ of arm i in the general rested rotting bandits case. Indeed, we show in Lemma 1.3.7 that $h_{i,T}^\pi$ is smaller than a problem-dependent quantity $h_{i,T}^+$ which is itself smaller by construction than a function of "gaps" $\Delta_{i,h_{i,T}^+-1}$,

$$h_{i,T}^+ \triangleq \max\left\{ h \leq 1 + \frac{2\alpha C_\pi^2 \sigma^2 \log(T)}{\Delta_{i,h-1}^2} \right\} \quad \text{with} \quad \Delta_{i,h} \triangleq \min_{j \in \mathscr{K}} \mu_j\left(N_{j,T}^\star - 1\right) - \overline{\mu}_i^h\left(N_{i,t}^\star + h\right). \tag{1.27}$$

R Notice that for stationary bandits, we have for all h, $\Delta_{i,h} = \Delta_i = \mu_\star - \mu_i$. In fact, $\Delta_{i,h}$ extends the notion of gap to our non-stationary setting : it is the average gap between the smallest value pulled by the optimal policy and the average value of the h first overpulls of arm i. We also highlight that $h_{i,t}^+$ is always defined because $h = 1$ always verify the self-bounding property.

Moreover, on the favorable event ξ_t^α, we can show that the regret of $h_{i,T}^\pi$ overpulls of arm i is bounded by $\mathscr{O}(\sqrt{h_{i,T}^\pi})$ (see Lemma 1.3.5, in Subsection 1.3.3). Hence, we bound $h_{i,T}^\pi$ by $h_{i,T}^+$ and we use the self-bounding property in the definition of $h_{i,T}^+$ (Equation 1.27) to get a $\mathscr{O}(\log(T))$ problem-dependent bound for our algorithms on any rotting bandit scenario.

Theorem 1.3.2 For any rotting bandit scenario with means $\{\mu_i\}_i \in \mathscr{L}_L^K$ and any time horizon T, $\pi \in \{\pi_R, \pi_F\}$ run with $\alpha \geq 5$ suffers an expected regret of

$$\mathbb{E}[R_T(\pi)] \leq \sum_{i \in \mathscr{K}} \left(\frac{2\alpha C_\pi^2 \sigma^2 \log(T)}{\Delta_{i,h_{i,T}^+-1}} + \sqrt{2\alpha C_\pi^2 \sigma^2 \log(T)} + 3L \right).$$

R The problem-dependent guarantee of `RAW-UCB` is 4 times smaller than the guarantee of `FEWA`: this is the benefits of upper-confidence bound index policies over confidence bound filtering ones. However, for $\alpha = 5$, `RAW-UCB` is still at a factor $\alpha C_{\pi_R}^2 = 20$ of the lower bound of **lai1985asymptotically**. This is mostly due to our proof technique. Indeed, **auer2002finite** also use a similar high-probability proof for `UCB1` and also get a large factor compared to the lower bound and an over-conservative tuning of the confidence bounds[2]. Yet, even compared to `UCB1`, we have to use a more

[1] α should be large enough to control the cost of the unfavorable events, see Lemma 1.3.4.

[2] To make the results comparable to the one of **auer2002finite**, we need to replace $2\sigma^2$ by $1/2$ for sub-Gaussian noise.

conservative tuning of the confidence bounds. On the first hand, we use a larger number of estimators at each round : Kt^2 instead of Kt for UCB. Hence, after taking the union bound, we need to increase α by one to have the same probability of the unfavorable event than for UCB1 (see Prop. 1.1.5). On the other hand, for reward functions in $\mathscr{L}_L$, the maximal possible regret at round t is bounded by Lt which is larger than the constant cost L for the stationary case. Thus, we have to increase α by one to control the cost of the unfavorable event. Notice that it is a consequence of our extended setting : we would not need to increase α for reward functions in $\mathscr{B}_B$.

While we presented our Theorems 1.3.1 and 1.3.2 with $\alpha \geq 5$, we could have similar results for $\alpha > 4$ by replacing the additive term $3KL$ by $(1+\zeta(\alpha-3))KL$[3]. For bounded reward functions, we can further reduce $\alpha > 3$. It is still much larger confidence interval than $\delta_t \sim \frac{1}{t\log t^2}$ which is used in UCB with asymptotic-optimal tuning for gaussian stationary bandits with fixed (and known) variance (**lattimore2019bandit**). We further discuss the notion of asymptotic optimality in rotting bandits in Section **??**.

1.3.3 Proof

Notation

Sketch of the proof

In Lemma 1.3.3, we split the regret decomposition according to whether the overpulls has been done on the favorable event ξ_t^α or not.

In Lemma 1.3.4, we show that the part of the expected regret due to pulls under $\overline{\xi_t^\alpha}$ is bounded by a constant with respect to T for $\alpha > 4$. Indeed, while we have only trivial bounds on the quality of the pulls on these events, we can control their probabilities thanks to Proposition 1.2.1.

In Lemma 1.3.5, we show that for $h_{i,T}^\pi$ overpulls of arm i, we suffer no more than $\widetilde{\mathscr{O}}\left(\sqrt{h_{i,t}^\pi}\right)$ on the favorable event. Indeed, thanks to Lemma 1.2.2 or 1.2.3, we know that the cost of the h before last pulls is bounded by $h \cdot c(h,\delta_t) = \widetilde{\mathscr{O}}\left(\sqrt{h}\right)$.

The proof of Theorem 1.3.1 follows by noticing that $\sum_{i\in\mathscr{K}} h_{i,T}^\pi \leq T$ which leads to the $\widetilde{\mathscr{O}}\left(\sqrt{KT}\right)$ rate. Indeed, thanks to the concavity of the $\sqrt{\cdot}$ and to Jensen's inequality, we find that the worst allocation is $h_{i,T}^\pi = \frac{T}{K}$.

In Lemma 1.3.7, we construct a problem-dependent bound of $h_{i,T}^\pi$ which extends the notion of gap for rotting bandits using Lemma 1.2.2 or 1.2.3.

The proof of Theorem 1.3.2 follows by plugging this bound in the result of Lemma 1.3.5.

[3] $\zeta(x) \triangleq \sum_n n^{-x}$

Full proof

Let $t_i^\pi(n)$ the function such that $t_i^\pi(n) = t$ when policy π selects arm i at time t for the n-th time. We call $\mu_T^+(\pi) \triangleq \max_{i\in\mathscr{K}} \mu_i\left(N_{i,T}^\pi\right)$, *i.e.* the largest available reward for π at round T+1.

Lemma 1.3.3 Let $h_{i,T}^\pi \triangleq |N_{i,T}^\pi - N_{i,T}^\star|$. For any policy π, the regret at round T is no bigger than

$$R_T(\pi) \leq \sum_{i\in\mathrm{OP}} \sum_{h=0}^{h_{i,T}^\pi - 1} \left[\xi^\alpha_{t_i^\pi(N_{i,T}^\star+h)}\right] \left(\mu_T^+(\pi) - \mu_i(N_{i,T}^\star + h)\right) + \sum_{t=1}^{T} \left[\overline{\xi_t^\alpha}\right] Lt.$$

We refer to the the first sum above as to A_π and to the second sum as to B.

Proof. We consider the regret at round T. We start from the upper bound in Equation 1.5,

$$R_T(\pi) \leq \sum_{i\in\mathrm{OP}} \sum_{h=0}^{h_{i,T}^\pi - 1} \left(\mu_T^+(\pi) - \mu_i(N_{i,T}^\star + h)\right). \tag{1.28}$$

Then, we need to separate overpulls that are done under ξ_t^α and under $\overline{\xi_t^\alpha}$. We introduce $t_i^\pi(n)$, the round at which π pulls arm i for the n-th time. We now make the round at which each overpull occurs explicit,

$$\begin{aligned}
R_T(\pi) &\leq \sum_{i\in\mathrm{OP}} \sum_{h=0}^{h_{i,T}^\pi - 1} \sum_{t=1}^{T} \left[t_i^\pi\left(N_{i,T}^\star + h\right) = t\right] \left(\mu_T^+(\pi) - \mu_i(N_{i,T}^\star + h)\right) \\
&\leq \underbrace{\sum_{i\in\mathrm{OP}} \sum_{h=0}^{h_{i,T}^\pi - 1} \sum_{t=1}^{T} \left[t_i^\pi\left(N_{i,T}^\star + h\right) = t \wedge \xi_t^\alpha\right] \left(\mu_T^+(\pi) - \mu_i(N_{i,T}^\star + h)\right)}_{A_\pi} \\
&+ \underbrace{\sum_{i\in\mathrm{OP}} \sum_{h=0}^{h_{i,T}^\pi - 1} \sum_{t=1}^{T} \left[t_i^\pi\left(N_{i,T}^\star + h\right) = t \wedge \overline{\xi_t^\alpha}\right] \left(\mu_T^+(\pi) - \mu_i(N_{i,T}^\star + h)\right)}_{B}.
\end{aligned}$$

For the analysis of the pulls done under ξ_t^α we do not need to know at which round it was done. Therefore,

$$A_\pi \leq \sum_{i\in\mathrm{OP}} \sum_{h=0}^{h_{i,T}^\pi - 1} \left[\xi^\alpha_{t(N_{i,t}^\star+h)}\right] \left(\mu_T^+(\pi) - \mu_i(N_{i,T}^\star + h)\right).$$

For `FEWA` or `RAW-UCB`, it is not easy to directly guarantee the low probability of overpulls (the second sum). Thus, we upper-bound the regret of each overpull at round t under $\overline{\xi_t^\alpha}$ by its maximum value Lt. While this is done to ease `FEWA` analysis, this is valid for any policy π. Then, noticing that we can have at most 1 overpull per round t, i.e., $\sum_{i\in\mathrm{OP}} \sum_{h=0}^{h_{i,T}^\pi - 1} \left[t_i^\pi\left(N_{i,T}^\star + h\right) = t\right] \leq 1$, we get

$$B \leq \sum_{t=1}^{T} \left[\overline{\xi_t^\alpha}\right] Lt \left(\sum_{i\in\mathrm{OP}} \sum_{h=0}^{h_{i,T}^\pi - 1} \left[t_i^\pi\left(N_{i,T}^\star + h\right) = t\right]\right) \leq \sum_{t=1}^{T} \left[\overline{\xi_t^\alpha}\right] Lt.$$

Therefore, we conclude that

$$R_T(\pi) \leq \underbrace{\sum_{i\in \mathrm{OP}} \sum_{h=0}^{h^{\pi}_{i,T}-1} \left[\xi^{\alpha}_{t_i^{\pi}(N^{\star}_{i,t}+h)}\right] \left(\mu_T^+(\pi) - \mu_i(N^{\star}_{i,T}+h)\right)}_{A_\pi} + \underbrace{\sum_{t=1}^{T} \left[\overline{\xi^{\alpha}_t}\right] Lt}_{B}.$$

■

Lemma 1.3.4 Let $\zeta(x) = \sum_n n^{-x}$. Thus, with $\delta_t = t^{-\alpha}$ and $\alpha > 4$, we can use Proposition 1.2.1 and get

$$\mathbb{E}[B] \triangleq \sum_{t=1}^{T} p\left(\overline{\xi^{\alpha}_t}\right) Lt \leq \sum_{t=1}^{T} KLt^{3-\alpha} \leq KL\zeta(\alpha-3).$$

In particular, for $\alpha \geq 5$, we have :

$$\mathbb{E}[B] \leq KL\zeta(2) \leq 2KL.$$

Lemma 1.3.5 We define $h^{\xi}_{i,T} \triangleq \max\left\{h \leq h_{i,T} |\ \xi^{\alpha}_{t_i^{\pi}(N^{\star}_{i,t}+h)}\right\}$, the largest number of overpulls of arm i pulled under ξ^{α}_t at round $t = t_i^{\pi}(N^{\star}_{i,t} + h^{\xi}_{i,T}) \leq T$. We also define $\mathrm{OP}_{\xi} \triangleq \left\{i \in \mathrm{OP} |\ h^{\xi}_{i,T} \geq 1\right\}$. For policy $\pi \in \{\pi_{\mathrm{R}}, \pi_{\mathrm{F}}\}$ with parameter α, A_π defined in Lemma 1.3.3 is upper-bounded by

$$\begin{aligned} A_\pi &\triangleq \sum_{i\in \mathrm{OP}} \sum_{h=0}^{h^{\pi}_{i,T}-1} \left[\xi^{\alpha}_{t_i^{\pi}(N^{\star}_{i,T}+h)}\right] \left(\mu_T^+(\pi) - \mu_i(N^{\star}_{i,T}+h)\right) \\ &\leq \sum_{i\in \mathrm{OP}_{\xi}} \left(C_\pi \sqrt{2\alpha\sigma^2\left(h^{\xi}_{i,T}-1\right)\log(T)} + C_\pi\sqrt{2\alpha\sigma^2\log(T)} + L\right). \end{aligned}$$

Proof. First, we define $h^{\xi}_{i,T} \triangleq \max\left\{h \leq h_{i,T} |\ \xi^{\alpha}_{t_i^{\pi}(N^{\star}_{i,t}+h)}\right\}$, the largest number of overpulls of arm i pulled at round $t_i \triangleq t_i^{\pi}(N^{\star}_{i,t} + h^{\xi}_{i,T}) \leq T$ under ξ^{α}_t. Now, we upper-bound A_π by including all the overpulls of arm i until the $h^{\xi}_{i,T}$-th overpull, even the ones under $\overline{\xi^{\alpha}_t}$,

$$\begin{aligned} A_\pi &\triangleq \sum_{i\in \mathrm{OP}} \sum_{h=0}^{h^{\pi}_{i,T}-1} \left[\xi^{\alpha}_{t_i^{\pi}(N^{\star}_{i,t}+h)}\right] \left(\mu_T^+(\pi) - \mu_i(N^{\star}_{i,T}+h)\right) \\ &\leq \sum_{i\in \mathrm{OP}_{\xi}} \sum_{h=0}^{h^{\xi}_{i,T}} \left(\mu_T^+(\pi) - \mu_i(N^{\star}_{i,T}+h)\right), \end{aligned}$$

where $\mathrm{OP}_{\xi} \triangleq \left\{i \in \mathrm{OP} |\ h^{\xi}_{i,T} \geq 1\right\}$. We can therefore split the second sum of $h^{\xi}_{i,T}$ term above into two parts. The first part corresponds to the first $h^{\xi}_{i,T} - 1$ (possibly zero) terms (overpulling differences) and the second part to the last $(h^{\xi}_{i,T} - 1)$-th one. Recalling that at

round t_i, arm i was selected under $\xi^{\alpha}_{t_i}$, we apply Lemma 1.2.2 or 1.2.3 to bound the regret caused by previous overpulls of i (possibly none),

$$A_\pi \leq \sum_{i \in \mathrm{OP}_\xi} \mu_T^+(\pi) - \mu_i \left(N^\star_{i,T} + h^\xi_{i,T} - 1\right) + C_\pi \left(h^\xi_{i,T} - 1\right) c\left(h^\xi_{i,T} - 1, \delta_{t_i}\right) \tag{1.29}$$

$$\leq \sum_{i \in \mathrm{OP}_\xi} \mu_T^+(\pi) - \mu_i \left(N^\star_{i,T} + h^\xi_{i,T} - 1\right) + C_\pi \left(h^\xi_{i,T} - 1\right) c\left(h^\xi_{i,T} - 1, \delta_T\right) \tag{1.30}$$

$$\leq \sum_{i \in \mathrm{OP}_\xi} \mu_T^+(\pi) - \mu_i \left(N^\star_{i,T} + h^\xi_{i,T} - 1\right) + C_\pi \sqrt{2\alpha\sigma^2 \left(h^\xi_{i,T} - 1\right) \log(T)}, \tag{1.31}$$

The second inequality is obtained because δ_t is decreasing and $c(.,\delta)$ is decreasing as well. The last inequality is the definition of confidence interval in Proposition 1.2.1. If $N^\star_{i,T} = 0$ and $h^\xi_{i,T} = 1$ then

$$\mu_T^+(\pi) - \mu_i(N^\star_{i,T} + h^\xi_{i,T} - 1) = \mu_T^+(\pi) - \mu_i(0) \leq L,$$

since $\mu_T^+(\pi) \leq \max_{j \in \mathscr{K}} \mu_j(0)$ and $\max_{j \in \mathscr{K}} \mu_j(0) - \mu_i(0) \leq L$ because $\{\mu_i\}_{i \in \mathscr{K}} \in \mathscr{L}_L^K$ (Def. 1.1.1). Otherwise, we can decompose

$$\mu_T^+(\pi) - \mu_i(N^\star_{i,T} + h^\xi_{i,T} - 1) = \underbrace{\mu_T^+(\pi) - \mu_i(N^\star_{i,T} + h^\xi_{i,T} - 2)}_{A_1} + \underbrace{\mu_i(N^\star_{i,T} + h^\xi_{i,T} - 2) - \mu_i(N^\star_{i,T} + h^\xi_{i,T} - 1)}_{A_2}.$$

For term A_1, since this $h^\xi_{i,T}$-th overpull is done under $\xi^{\alpha}_{t_i}$, by Lemma 1.2.2 or 1.2.3 we have that

$$A_1 = \mu_T^+(\pi) - \overline{\mu}_i^1(N^\star_{i,T} + h^\xi_{i,T} - 1) \leq 1c(1, \delta_{t_i}) \leq 2c(1, \delta_T) \leq C_\pi \sqrt{2\alpha\sigma^2 \log(T)}.$$

The second difference, $A_2 = \mu_i(N^\star_{i,T} + h^\xi_{i,T} - 2) - \mu_i(N^\star_{i,T} + h^\xi_{i,T} - 1)$ cannot exceed L, since by the assumptions of our setting (Def. 1.1.1), the maximum decay in one round is bounded. Therefore, we further upper-bound Equation 1.31 as

$$A_\pi \leq \sum_{i \in \mathrm{OP}_\xi} \left(C_\pi \sqrt{2\alpha\sigma^2 \left(h^\xi_{i,T} - 1\right) \log(T)} + C_\pi \sqrt{2\alpha\sigma^2 \log(T)} + L \right). \tag{1.32}$$

■

Theorem 1.3.6 For any rotting bandit scenario with means $\{\mu_i\}_i \in \mathscr{L}_L^K$ and any time horizon T, $\pi \in \{\pi_R, \pi_F\}$ run with $\alpha \geq 5$ suffers an expected regret of

$$\mathbb{E}[R_T(\pi)] \leq C_\pi \sqrt{2\alpha\sigma^2 \log(T)} \left(\sqrt{KT} + K\right) + 3KL.$$

Proof. In Lemma 1.3.3, we split the regret in two parts. The first one B corresponds to the regret due to unfavorable events $\overline{\xi_t^\alpha}$. We do not derive any guarantee of our algorithms on these events but their probabilities can be controlled thanks to parameter α. Hence, for $\alpha > 4$, we show in Lemma 1.3.4 that the part of the expected regret due to unfavorable events can be bounded by a constant w.r.t. T. Yet, we choose $\alpha \geq 5$ to have a small constant.

The second one A_π corresponds to the regret due to favorable events ξ_t^α which can be bounded for our two algorithms (`FEWA` and `RAW-UCB`) thanks to Lemma 1.3.5. In order to get a problem-independent upper bound, we need to replace $h^\xi_{i,T}$ by a problem-independent quantity. Starting from Lemma 1.3.5,

$$A_\pi \leq \sum_{i \in \mathrm{OP}_\xi} \left(C_\pi \sqrt{2\alpha\sigma^2 \left(h^\xi_{i,T} - 1\right) \log(T)} + C_\pi \sqrt{2\alpha\sigma^2 \log(T)} + L \right).$$

Since $\mathrm{OP}_\xi \subseteq \mathrm{OP}$, we can upper-bound the number of terms in the above sum by K. Next, the total number of overpulls $\sum_{i \in \mathrm{OP}} h_{i,T}$ cannot exceed T. As square-root function is concave we can use Jensen's inequality. Moreover, we can deduce that the worst allocation of overpulls is the uniform one, i.e., $h_{i,T} = T/K$,

$$\begin{aligned} A_\pi &\leq K(C_\pi \sqrt{2\alpha\sigma^2 \log(T)} + L) + C_\pi \sqrt{2\alpha\sigma^2 \log(T)} \sum_{i \in \mathrm{OP}} \sqrt{(h_{i,T} - 1)} \\ &\leq K(C_\pi \sqrt{2\alpha\sigma^2 \log(T)} + L) + C_\pi \sqrt{2\alpha\sigma^2 KT \log(T)}. \end{aligned} \tag{1.33}$$

Therefore, using Lemma 1.3.3 together with Equations 1.33 and Lemma 1.3.4, we bound the total expected regret as

$$\mathbb{E}[R_T(\pi)] \leq C_\pi \sqrt{2\alpha\sigma^2 \log(T)} \left(\sqrt{KT} + K\right) + 3KL. \tag{1.34}$$

■

Lemma 1.3.7 Let $\mu_T^- \triangleq \min_{i \in \mathscr{K}^\star} \mu_i \left(N^\star_{i,T} - 1\right)$ with $\mathscr{K}^\star \triangleq \left\{ i \in \mathscr{K} \,|\, N^\star_{i,T} \geq 1 \right\}$; and $\Delta_{i,h} \triangleq \mu_T^- - \overline{\mu}_i^h \left(N^\star_{i,t} + h\right)$. $h^\xi_{i,T}$ defined in Lemma 1.3.3 is upper-bounded by a problem-dependent quantity,

$$h^\xi_{i,T} \leq h^+_{i,T} \triangleq \max \left\{ h \leq T \,\middle|\, h \leq 1 + \frac{2\alpha C_\pi^2 \sigma^2 \log(T)}{\Delta^2_{i,h-1}} \right\} \leq 1 + \frac{2\alpha C_\pi^2 \sigma^2 \log(T)}{\Delta^2_{i,h^+_{i,T}-1}}.$$

Proof. We want to bound $h^\xi_{i,T}$ with a problem dependent quantity $h^+_{i,T}$. We remind the reader that for arm i at round T, the $h^\xi_{i,T}$-th overpull is pulled under $\xi^\alpha_{t_i}$ at round t_i.

Therefore, Lemma 1.2.2 or 1.2.3 applies and we have

$$\begin{aligned}\overline{\mu}_i^{h_{i,T}^{\xi}-1}\left(N_{i,T}^{\star}+h_{i,T}^{\xi}-1\right) &\geq \mu_T^{+}(\pi)-C_\pi c\left(h_{i,T}^{\xi}-1,\delta_{t_i}\right)\\ &\geq \mu_T^{+}(\pi)-C_\pi c\left(h_{i,T}^{\xi}-1,\delta_T\right)\\ &\geq \mu_T^{+}(\pi)-C_\pi\sqrt{\frac{2\alpha\sigma^2\log(T)}{h_{i,T}^{\xi}-1}},\end{aligned}$$

Hence, we have that

$$h_{i,T}^{\xi}\leq 1+\frac{2\alpha C_\pi^2\sigma^2\log(T)}{\left(\mu_T^{+}(\pi)-\overline{\mu}_i^{h_{i,T}^{\xi}-1}\left(N_{i,T}^{\star}+h_{i,T}^{\xi}-1\right)\right)^2}. \tag{1.35}$$

Yet, this upperbound still depends on random quantities such as $\mu_T^{+}(\pi)$ or $h_{i,T}^{\xi}$ on the denominator. Consider the smallest value collected by the optimal policy,

$$\mu_T^{-}\triangleq\min_{i\in\mathscr{K}^\star}\mu_i\left(N_{i,T}^{\star}-1\right)\text{ with }\mathscr{K}^\star\triangleq\left\{i\in\mathscr{K}\,|N_{i,T}^{\star}\geq 1\right\}.$$

It is the T-th largest value among the KT possible ones. Therefore, since $\overline{\mu}_i^{h_{i,T}^{\xi}-1}\left(N_{i,T}^{\star}+h_{i,T}^{\xi}-1\right)$ is an average of overpulls value, which are smaller or equal to μ_T^{-}.

$$\mu_T^{-}\geq\overline{\mu}_i^{h_{i,T}^{\xi}-1}\left(N_{i,T}^{\star}+h_{i,T}^{\xi}-1\right),$$

Moreover, $\mu_T^{-}>\mu_T^{+}(\pi)$ implies that the regret is 0. Indeed, in that case $\mu_T^{+}(\pi)$ - the pull with the largest value among the remaining values at the end of the game for π - is *strictly smaller* than μ_T^{-} - the T-th largest reward sample. Therefore, π has collected the T largest value and has zero regret. Hence, we focus on the case $\mu_T^{-}\leq\mu_T^{+}(\pi)$, for which the regret may not be zero. In that case, we can upperbound the RHS term Equation 1.35 by replacing the random quantity $\mu_T^{+}(\pi)$ by the smaller quantity μ_T^{-}. Hence,

$$h_{i,T}^{\xi}\leq 1+\frac{2\alpha C_\pi^2\sigma^2\log(T)}{\left(\mu_T^{+}(\pi)-\overline{\mu}_i^{h_{i,T}^{\xi}-1}\left(N_{i,T}^{\star}+h_{i,T}^{\xi}-1\right)\right)^2}\leq 1+\frac{2\alpha C_\pi^2\sigma^2\log(T)}{\Delta^2_{i,h_{i,T}^{\xi}-1}},$$

with $\Delta_{i,h}\triangleq\mu_T^{-}-\overline{\mu}_i^{h}\left(N_{i,t}^{\star}+h\right)$, the difference between the lowest mean value of the arm pulled by $\pi^\star$ and the average of the h first overpulls of arm i. Yet, this self-bounding property of $h_{i,T}^{\xi}$ is not a proper problem-dependent upper bound. We will consider the largest h which satisfies this self-bounding property, i.e.

$$h_{i,T}^{+}\triangleq\max\left\{h\leq T\,\middle|\,h\leq 1+\frac{2\alpha C_\pi^2\sigma^2\log(T)}{\Delta^2_{i,h-1}}\right\}.$$

We have that,

$$h^{\xi}_{i,T} \leq h^{+}_{i,T} \leq 1 + \frac{2\alpha C_\pi^2 \sigma^2 \log(T)}{\Delta^2_{i,h^{+}_{i,T}-1}}.$$

■

Theorem 1.3.8 For any rotting bandit scenario with means $\{\mu_i\}_i \in \mathscr{L}_L^K$ and any time horizon T, $\pi \in \{\pi_R, \pi_F\}$ run with $\alpha \geq 5$ suffers an expected regret of

$$\mathbb{E}[R_T(\pi)] \leq \sum_{i \in \mathscr{K}} \left(\frac{2\alpha C_\pi^2 \sigma^2 \log(T)}{\Delta_{i,h^{+}_{i,T}-1}} + \sqrt{2\alpha C_\pi^2 \sigma^2 \log(T)} + 3L \right).$$

Proof. We use Lemmas 1.3.5 and Lemma 1.3.7 to bound A_π (see Lemma 1.3.3). Indeed, since the square-root function is increasing, we can upper-bound the result in Lemma 1.3.5 by replacing $h^{\xi}_{i,T}$ by its upper bound in Lemma 1.3.7

$$\begin{aligned} A_\pi &\leq \sum_{i \in \mathrm{OP}_\xi} \left(C_\pi \sqrt{2\alpha\sigma^2 \log(T)} \left(1 + \sqrt{h^{+}_{i,T} - 1}\right) + L \right) \\ &\leq \sum_{i \in \mathrm{OP}_\xi} \left(C_\pi \sqrt{2\alpha\sigma^2 \log(T)} \left(1 + \frac{C_\pi \sqrt{2\alpha\sigma^2 \log(T)}}{\Delta_{i,h^{+}_{i,T}-1}}\right) + L \right). \end{aligned}$$

Notice that the quantity $\mathrm{OP}_\xi \subset \mathscr{K}$. Therefore, we have

$$A_\pi \leq \sum_{i \in \mathscr{K}} \left(\frac{2\alpha C_\pi^2 \sigma^2 \log(T)}{\Delta_{i,h^{+}_{i,T}-1}} + \sqrt{2\alpha C_\pi^2 \sigma^2 \log(T)} + L \right). \tag{1.36}$$

Using Lemmas 1.3.3, 1.3.4, and Equation 1.36 we get

$$\begin{aligned} \mathbb{E}[R_T(\pi)] &= \mathbb{E}[A_\pi] + \mathbb{E}[B] \\ &\leq \sum_{i \in \mathscr{K}} \left(\frac{2\alpha C_\pi^2 \sigma^2 \log(T)}{\Delta_{i,h^{+}_{i,T}-1}} + \sqrt{2\alpha C_\pi^2 \sigma^2 \log(T)} + L \right) + 2KL \\ &\leq \sum_{i \in \mathscr{K}} \left(\frac{2\alpha C_\pi^2 \sigma^2 \log(T)}{\Delta_{i,h^{+}_{i,T}-1}} + \sqrt{2\alpha C_\pi^2 \sigma^2 \log(T)} + 3L \right). \end{aligned}$$

■

1.4 Efficient algorithms

1.4.1 The numerical cost of adaptive windows

In the three last sections, we presented two adaptive windows algorithms whose significantly improved over state-of-the-art algorithms, both theoretically and experimentally.

Yet, we highlight that these improvements are computationally expensive. Indeed, at each round t, we store, update and compare $\mathcal{O}(t)$ statistics.

The full update of the statistics can be done at a worst case cost of $\mathcal{O}(t)$. Indeed, each statistics $\widehat{\mu}_i^h$ can be refreshed with a $\mathcal{O}(1)$ operation :

$$\widehat{\mu}_i^{h+1}(n+1) = \frac{h}{h+1}\widehat{\mu}_i^h(n) + \frac{1}{h+1}o_t\,.$$

The comparison part in both FEWA and RAW-UCB is also a $\mathcal{O}(t)$ operations. In FEWA, we do a scan based on $\widehat{\mu}_i^h$ for all $i \in \mathcal{K}_h$ with increasing h. Hence, the total number of unitary operation is in $\mathcal{O}(t)$ in the worst case, as it scales with the number of statistics. RAW-UCB computes one UCB for each of the $\mathcal{O}(t)$ statistics. For each arm, it selects the minimum UCB as index, which can be done with complexity $\mathcal{O}(t)$. Finally, finding the largest index is an $\mathcal{O}(K)$ operations. Therefore, we can conclude,

Proposition 1.4.1 FEWA and RAW-UCB have a $\mathcal{O}(t)$ worst-case complexity per round t in time and memory.

R SWA (h) has a $\mathcal{O}(h)$ worst-case complexity in time and memory because the sliding-window mechanism need to store and update $\mathcal{O}(h)$ statistics to always have the average of the h last sample ready. Hence, when it is optimally tuned for the minimax bound, SWA has a $\mathcal{O}\left(T^{2/3}\right)$ per round complexity. As often in non-stationary bandits, it may be possible to replace sliding window statistics by discounted statistics. Such modification often leads to slightly worse theoretical regret rate but to a much better $\mathcal{O}(K)$ complexity.

Hence, handling a large number of windows, which is the main strength of our algorithms to achieve a lower regret, is a significant drawback when it comes to design fast algorithm. Therefore, it is an open question whether one can enjoy the benefits of adaptive windows without suffering large time and space complexity.

1.4.2 The efficient update trick

We detail EFF_UPDATE, an update scheme to handle efficiently statistics of different windows. A similar yet different approach has appeared independently in the context of streaming mining (**bifet2007learning**). EFF_UPDATE is built around two main ideas.

First, at any time t we can avoid using $\left\{\widehat{\mu}_i^h\right\}_h$ for all possible windows h starting from 1 with an increment of 1. In fact, both statistics $\widehat{\mu}_i^h$ and constructed confidence levels $c(h,\delta_t)$ have very close value for successive h as h becomes large :

$$\widehat{\mu}_i^{h+1}(n) = \widehat{\mu}_i^h(n) + \mathcal{O}\left(\frac{\sigma+L}{h}\right),$$
$$c(h+1,\delta_t) = c(h,\delta_t) + \mathcal{O}\left(\frac{\sigma}{h^{3/2}}\right).$$

Hence, in both FEWA and RAW-UCB, we compute a lot of very similar quantities. Instead, we could use fewer statistics which are significantly different : $\left\{\widehat{\mu}_i^h(N_{i,t-1}^\pi)\right\}_{h\in H_{i,m}}$, where the window h is dispatched on a geometric grid,

$$H_{i,m}\left(N_{i,t-1}^\pi\right) \triangleq \left\{h_j \in \left\{1,\dots,N_{i,t-1}^\pi\right\} \mid h_{j+1} = \left\lceil m\cdot h_j\right\rceil \text{ and } h_1 = 1\right\} \quad \text{with } m>1.$$

When there is no confusion, we drop the dependency in $N_{i,t-1}^\pi$. This modification alone is not enough to reduce both the time and space complexity. Indeed, updating $\widehat{\mu}_i^h$ requires to replace the h-th last sample by the new one o_t. Hence, we need to store all the collected statistics to be able to update all the $\widehat{\mu}_i^h$ for all h with $\mathcal{O}(1)$ complexity. Therefore, in EFF_UPDATE, we will use $\mathcal{O}(K\log(t))$ *delayed* statistics that we can update with $\mathcal{O}(K\log(t))$ space and time complexity.

EFF_UPDATE (Alg. 0) takes as input the new observation o_t that the learner gets at the N_i-th pull of arm i; the geometric window grid $H_{i,m}$ tuned with an hyperparameter $m>1$, and for each window h_j in this grid, three different numbers $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$, $p_i^{h_j}$, $n_i^{h_j}$. $\left\{\widehat{\mu}_{i,\texttt{eff}}^{h_j}\right\}_{i,h_j}$ represents the set of *current* statistics of window size h_j that will be used instead of $\left\{\widehat{\mu}_i^h\right\}_{i,h}$ in our efficient algorithms. We also store a pending statistic $p_i^{h_j}$ and a count $n_i^{h_j}$ which are used in the sparse update procedure of $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$. EFF_UPDATE outputs an updated set of statistics.

Algorithm 8 EFF_UPDATE

Require: o_t, $H_{i,m} \leftarrow \left\{h_j < \lceil m \cdot N_i \rceil \mid h_{j+1} = \lceil m \cdot h_j \rceil \text{ with } h_1 = 1\right\}$, $\left\{\left\{\widehat{\mu}_{i,\texttt{eff}}^{h_j}, p_i^{h_j}, n_i^{h_j}\right\}\right\}_{h_j \in H_{i,m}}$

1: **if** $N_i = \max(H_{i,m})$ **then** ▷ Create a new triplet with window $h_j = \lceil m \cdot N_i \rceil$
2: $\quad H_{i,m} \leftarrow H_{i,m} \cup \{\lceil m \cdot N_i \rceil\}$
3: $\quad p_i^{\lceil m \cdot N_i \rceil} = p_i^{N_i}$
4: $\quad n_i^{\lceil m \cdot N_i \rceil} \leftarrow n_i^{N_i}$
5: $\quad \widehat{\mu}_{i,\texttt{eff}}^{\lceil m \cdot N_i \rceil} \leftarrow \texttt{None}$
6: **end if**
7: $p_i^1 \leftarrow o_t$ ▷ Update the first triplet with o_t
8: $n_i^1 \leftarrow 1$
9: $\widehat{\mu}_{i,\texttt{eff}}^1 \leftarrow o_t$
10: **for** $h_j \in H_{i,m} \setminus \{1\}$ **do** ▷ Update the other pending statistics $p_i^{h_j}$ and $n_i^{h_j}$
11: $\quad p_i^{h_j} \leftarrow p_i^{h_j} + o_t$
12: $\quad n_i^{h_j} \leftarrow n_i^{h_j} + 1$
13: **end for**
14: **for** $h_j \in \text{SORT\_DESC}(H_{i,m} \setminus \{1\})$ **do**
15: $\quad$ **if** $n_i^{h_j} = h_j$ **then**
16: $\quad\quad \widehat{\mu}_{i,\texttt{eff}}^{h_j} \leftarrow p_i^{h_j}/h_j$ ▷ Replace the current statistic $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$
17: $\quad\quad p_i^{h_j} = p_i^{h_{j-1}}$ ▷ Refresh the pending statistics
18: $\quad\quad n_i^{h_j} \leftarrow n_i^{h_{j-1}}$
19: $\quad$ **end if**
20: **end for**

Ensure: $\left\{\left\{\widehat{\mu}_{i,\texttt{eff}}^{h_j}, p_i^{h_j}, n_i^{h_j}\right\}\right\}_{h_j \in H_{i,m}}$

The core of EFF_UPDATE is divided in four parts: 1) From Lines 1 to 6, we create new statistics at a logarithmic rate with respect to the growth of N_i; 2) From Lines 7 to 9, we update the statistics of window $h_1 = 1$; 3) From Lines 10 to 13, we update the other pending statistics and count; 4) From Lines 14 to 20, we eventually update $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$ and refresh the correspounding pending statistic and count. The remaining details are quite technical. Thus, we first give the high-level properties that are ensured by the recursive usage of EFF_UPDATE. Then, we prove them by going through the algorithm line by line.

Proposition 1.4.2 $\left\{\left\{\widehat{\mu}_{i,\texttt{eff}}^{h_j}, p_i^{h_j}, n_i^{h_j}\right\}\right\}_{h_j \in H_{i,m}}$, constructed recursively with EFF_UPDATE with initial value $\left\{\left\{\widehat{\mu}_{i,\texttt{eff}}^1 : \texttt{None}, p_i^1 : 0, n_i^1 : 0\right\}\right\}$ have the following properties :

1. $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$ is the average of exactly h_j consecutive samples among the $2h_j - 1$ last ones.
2. The delay between two updates of $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$ is in $\left\{\left\lceil \frac{m-1}{m} h_j \right\rceil, \ldots, h_j - 1\right\}$.
3. When $m = 2$, $h_j = 2^{j-1}$. Moreover, for $j \geq 2$, $\widehat{\mu}_{i,\texttt{eff}}^{h_j}(n)$ is updated every 2^{j-2} rounds (and every rounds for $j = 1$).

4. $p_i^{h_j}$ is the sum of the $n_i^{h_j}$ last samples.
5. $n_i^{h_j} < h_j$ for $j > 2$. Also, $n_i^1 \leq 1$.
6. $\left\{n_i^{h_j}\right\}_{h_j}$ is an non-decreasing sequence with respect to h_j (or j).

Proof. The three last properties are trivially true at the initialization. Thus, we show by induction that they remain true after updates.

Proof of 4.

At Lines 3 and 4, we create a new pending statistics and count by intializing them with other statistics and count. Hence, because of the recursion hypothesis, all the pending statistics $p_i^{h_j}$ (including the created one) contains the sum of the $n_i^{h_j}$ *before last* pulls. At Lines 7 and 8, we update p_i^1 with the last sample and set n_i^1 to 1. At Lines 11 and 12, we add the last sample to $p_i^{h_j}$ (which was containing the before last samples) and increase the count by 1. Hence, at the end of Line 12, all the $p_i^{h_j}$ contains the sum of the last $n_i^{h_j}$ samples. Thus, refreshing $p_i^{h_j}$ and $n_i^{h_j}$ with $p_i^{h_{j-1}}$ and $n_i^{h_{j-1}}$ keeps this property true (Lines 17 and 18).

Proof of 5.

For $j \geq 2$, $n_i^{h_j}$ is created at Line 4 with $n_i^{N_i} = N_i - 1 < \lceil m \cdot N_i \rceil = h_j$ ($m > 1$). Indeed, at the beginning of N_i-th update $\widehat{\mu}_{i,\texttt{eff}}^{N_i}$ is not created yet and $p_i^{h_j}$ contains the $n_i^{h_j} = N_i - 1$ before last pulls. Then, $n_i^{h_j}$ ($j \geq 2$) is increased by one at each update at Line 12. When it reaches $n_i^{h_j} = h_j$ (Line 15), it is replaced by the precedent count $n_i^{h_{j-1}} < h_j$ (Line 12). Indeed, for $j > 3$, $n_i^{h_{j-1}} < h_{j-1} < h_j$ before the updating scheme (recursion hypothesis). After the increment by one at Line 12, we still have $n_i^{h_{j-1}} \leq h_{j-1} < h_j$. For $j = 2$, $n_i^{h_{j-1}} = n_i^1 \leq 1 < h_2$.

Proof of 6.

At Line 4, we create a new pending count corresponding to the largest h_j and we initialize it with the precedent largest count. At Lines 8 and 12, we set $n_i^1 = 1$ and increase all the other $n_i^{h_j}$ by one. This operation preserves the non-decreasing property of the ordered set. Last, at Line 18, we set few counts $n_i^{h_j}$ to the precedent value $n_i^{h_{j-1}}$- which also preserves the non-decreasing property of the ordered set.

Proof of 1 and 2.

Thanks to Property 4, we know that $p_i^{h_j}$ is the sum of the $n_i^{h_j}$ last sample. It is still true at the end of Line 12 (see the proof). Then, at Line 16, and given the condition in Line 15,

we set $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$ with the average of the last h_j sample. Then, $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$ is not updated untill the condition at Line 15 is fulfilled again. Given that $n_i^{h_j}$ is refreshed with a quantity larger or equal to 1 and smaller or equal to h_{j-1} at Line 18. Then, it is increased by one at each update. we know that $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$ will be updated at least every $h_j - 1$, and at most every $h_j - h_{j-1}$ round. Hence, considering the worst possible delay we can conclude : $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$ is the average of exactly h_j consecutive samples among the $2h_j - 1$ last ones. Last, considering that $h_{j-1} \leq h_j/m$, we conclude that the minimal delay is larger or equal to $\frac{m-1}{m} h_j$.

Proof of 3.

When $m = 2$, it is easy to find by induction that,

$$h_{j+1} = \lceil m \cdot h_j \rceil = 2h_j = 2^j.$$

For $j = 1$, $\widehat{\mu}_{i,\texttt{eff}}^{1}$ is updated at every update at Line 9. By induction on $j \geq 2$, $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$ is updated (Line 16) for the first time after $h_j = 2^{j-1} = 4 \cdot 2^{j-3}$ pulls. Therefore, it is also an updating pull for $\widehat{\mu}_{i,\texttt{eff}}^{h_{j-1}}$ (by the induction hypothesis) and n_j is set with $n_{j-1} = 2^{j-2}$ at Line 18. Notice that we sort $H_{i,m}$ in the decreasing order at Line 14, hence n_j is updated with n_{j-1} before it is itself updated with n_{j-2}. Hence, $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$ is updated again in $h_j - 2^{j-2} = 2^{j-2}$ pulls, *i.e.* after $6 \cdot 2^{j-3}$ pulls of arm i. Again, n_j is set with $n_{j-1} = 2^{j-2}$ (because it is an updating pull for $\widehat{\mu}_{i,\texttt{eff}}^{h_{j-1}}$). By induction, we see that the k-th update happen at pull $(k+1) \cdot 2^{j-2}$, *i.e.* every 2^{j-2} pulls. ■

R At Line 18, we refresh $n_i^{h_j}$ with $n_i^{h_{j-1}}$ which is often larger than 1. Indeed, we could refresh $p_i^{h_j}$ and $n_i^{h_j}$ at 0. Yet, in order to reduce the delay in the update, we use the variable available in the memory which contains the sums of h last sample, with the largest $h < h_j$. According to Properties 4, 5 and 6, this quantity is $p_i^{h_{j-1}}$. Therefore, while we were not able to prove that it reduces

Notice that we sort $H_{i,m}$ in the decreasing order at Line 14 to minimize the delay: if there is two consecutive updates of $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$ and $\widehat{\mu}_{i,\texttt{eff}}^{h_{j+1}}$ at the same run of `EFF_UPDATE`, doing a backward loop guarantees to refresh $n_i^{h_{j+1}}$ with a larger value than with a forward loop.

1.4.3 `EFF-FEWA` and `EFF-RAW-UCB`

`EFF-FEWA` and `EFF-RAW-UCB` are the two efficient versions of our initial algorithms. With an hyperparameter $m > 1$, they use `EFF_UPDATE` instead of `UPDATE` (Lines 4 and 18 in `FEWA` and Lines 4 and 9 in `RAW-UCB`). Therefore, they use $\left\{\widehat{\mu}_{i,\texttt{eff}}^{h_j}\right\}_{i,h_j \in H_{i,m}}$ instead of $\left\{\widehat{\mu}_i^h\right\}_{i,h \leq N_{i,t-1}}$.

More precisely, in FEWA, we replace the increment $h \leftarrow h+1$ by $h \leftarrow \lceil m \cdot h \rceil$ at Line 12. Hence, the next set is not called $\mathscr{K}_{h+1}$ but $\mathscr{K}_{\lceil m \cdot h \rceil}$ (Line 11 in FEWA and Line 6 in FILTER). Finally, at Lines 13 and 14, the condition is not $N_{i_t} = h$ but $N_{i_t} \leq h$. In the FILTER procedure, we also change $\widehat{\mu}_i^h$ by $\widehat{\mu}_{i,\texttt{eff}}^h$ at Lines 2 and 4. In RAW-UCB, we only change the $h \leq N_i$ by $h_j \in H_{i,m}$ and $\widehat{\mu}_i^h$ by $\widehat{\mu}_{i,\texttt{eff}}^h$ in the index computation at Line 7.

Proposition 1.4.3 EFF-FEWA and EFF-RAW-UCB tuned with hyperparmaeter m have a $\mathscr{O}(K \log_m(t))$ worst-case time and space complexity at round t.

Proof. The total number of statistics for each arm i at round t is bounded by $\mathscr{O}(\log_m(t))$. Indeed,

$$t \geq N_{i,t-1} \geq h_j \geq m^{j-1} \implies j \leq 1 + \log_m(t).$$

Moreover, in EFF_UPDATE we use 3 numbers for each $\left\{\widehat{\mu}_{i,\texttt{eff}}^{h_j}\right\}_j$. Hence, the space complexity scales with

$$\sum_{i \in \mathscr{K}} |H_{i,m}| = \sum_{i \in \mathscr{K}} \mathscr{O}(\log_m(t)) = \mathscr{O}(K \log_m(t)).$$

The time complexity of EFF_UPDATE scales with the number of statistics in arm i_t, *i.e.* at most $\mathscr{O}(\log_m(t))$. The indexes computation of EFF-RAW-UCB find the minimum of K sets with cardinality $\mathscr{O}(\log_m(t))$, while finding the maximum among these indexes is a $\mathscr{O}(K)$ operation. Thus, the worst-case time complexity is $\mathscr{O}(K \log_m(t))$. EFF-FEWA uses at most $\mathscr{O}(\log_m(t))$ times the procedure FILTER whose inner complexity scales with $|\mathscr{K}_h| \leq K$. Therefore, in the worst case, the time complexity of EFF-FEWA at round t is bounded by $\mathscr{O}(K \log_m(t))$. ■

1.4.4 Analysis

The analysis of RAW-UCB (respectively FEWA) only uses Proposition 1.2.1 and Lemma 1.2.3 (respectively 1.2.2). We will derive analoguous results for EFF-RAW-UCB and EFF-FEWA, which allows us to reproduce very similar upper-bounds on the regret.

A favorable event for efficiently updated adaptive windows

Proposition 1.4.4 For any round t and confidence $\delta_t \triangleq 2t^{-\alpha}$, let

$$\xi_{t,\texttt{eff}}^{\alpha} \triangleq \left\{ \forall i \in \mathscr{K},\ \forall n \leq t-1,\ \forall h_j \in H_{i,m}(n), \left| \widehat{\mu}_{i,\texttt{eff}}^{h_j}(n) - \overline{\mu}_{i,\texttt{eff}}^{h_j}(n) \right| \leq c(h_j, \delta_t) \right\}$$

be the event under which the estimates at round t are all accurate up to $c(h,\delta_t) \triangleq \sqrt{2\sigma^2 \log(2/\delta_t)/h}$. Then, for a policy π which pulls each arms once at the beginning, and for all $t > K$,

$$\mathbb{P}\left[\overline{\xi_{t,\texttt{eff}}^{\alpha}}\right] \leq 3Kt\delta_t = 6Kt^{1-\alpha}.$$

Proof. As in Propositions 1.1.5 and 1.2.1, we have to count the number of statistics that are required to hold in the confidence region. Calling $u_j(t)$ the number of update of statistics $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$ after t pulls, we have

$$\begin{aligned}\mathbb{P}\left[\overline{\xi^{\alpha}_{t,\texttt{eff}}}\right] &\leq \sum_{i\in\mathscr{K}} \sum_{j=1}^{\lfloor \log_2(t)\rfloor} u_j(t)\delta_t \\ &\leq \sum_{i\in\mathscr{K}} \left(t-1+\sum_{j=2}^{\lfloor \log_2(t)\rfloor} \frac{t-1}{2^{j-2}}\right)\delta_t \\ &\leq 3Kt\delta_t\end{aligned}$$

In the second inequality, we use Property 3 in Proposition 1.4.2: statistics $\widehat{\mu}_{i,\texttt{eff}}^{h_j}(n)$ is only updated every 2^{j-2} pulls for $j \geq 2$ (and every pull for $j = 1$). ■

Lemma 1.4.5 At round t on favorable event ξ_t^{α}, if arm i_t is selected by `EFF-RAW-UCB` (m=2), for any $h \leq N_{i,t-1}$, the average of its h last pulls cannot deviate significantly from the best available arm at that round, i.e.,

$$\overline{\mu}_{i_t}^{h}(t-1,\pi) \geq \max_{i\in\mathscr{K}} \mu_i(t,N_{i,t-1}) - \frac{2\sqrt{2}}{\sqrt{2}-1} c(h,\delta_t).$$

Proof. We denote by $\overline{\mu}_i^{hh'}(t-1,\pi)$ and $\widehat{\mu}_i^{hh'}(t-1,\pi)$ the true mean and empirical average associated to the $h'-h$ samples between the h-th last one (included) and the h'-th last one

(excluded). Let j_h such that : $2^{j_h} - 1 \leq h < 2^{j_h+1} - 1$.

$$\begin{aligned}
\overline{\mu}_{i_t}^{h}(t-1,\pi) &\geq \overline{\mu}_{i_t}^{2^{j_h}-1}(t-1,\pi) \\
&= \sum_{j=0}^{j_h-1} \frac{2^j}{2^{j_h}-1} \overline{\mu}_{i_t}^{2^j 2^{j+1}}(t-1,\pi) \\
&\geq \sum_{j=0}^{j_h-1} \frac{2^j}{2^{j_h}-1} \left(\widehat{\mu}_{i,\texttt{eff}}^{h_j} - c(2^j,\delta_t) \right) \\
&\geq \min_j \left(s_{i_t j}^c + c(2^j,\delta_t) \right) - \sum_{j=0}^{j_h-1} \frac{2^{j+1}}{2^{j_h}-1} c(2^j,\delta_t) \\
&= \min_j \left(s_{i_t j}^c + c(2^j,\delta_t) \right) - \frac{2c(1,\delta_t)}{2^{j_h}-1} \sum_{j=0}^{j_h-1} 2^{\frac{j}{2}} \\
&= \min_j \left(s_{i_t j}^c + c(h_j,\delta_t) \right) - \frac{2c(1,\delta_t)}{2^{j_h}-1} \frac{2^{\frac{j_h}{2}}-1}{\sqrt{2}-1} \\
&\geq \min_j \left(s_{i_t j}^c + c(h_j,\delta_t) \right) - \frac{2\sqrt{2}c(2^{j_h+1},\delta_t)}{\sqrt{2}-1} \\
&\geq \min_j \left(s_{i_t j}^c + c(h_j,\delta_t) \right) - \frac{2\sqrt{2}c(h_j,\delta_t)}{\sqrt{2}-1} \\
&\geq \max_{i \in \mathscr{K}} \mu_i(t, N_{i,t-1}) - \frac{2\sqrt{2}c(h_j,\delta_t)}{\sqrt{2}-1}
\end{aligned}$$

The first inequality is due to the decreasing nature of the reward. The second inequality is because, on ξ_t, s_{ij}^c concentrates near a value which is smaller than $\overline{\mu}_i^{2^j 2^{j+1}}(t-1,\pi)$ because it is an average from a sequence of consecutive reward which is newer than $2h_j \leq 2^{j+1}$ (when $m \leq 2$). The third inequality holds by selecting the minimum. The fourth one is standard algebra. The fifth one hold because $c(\cdot,\delta_t)$ decreases with h and $h_j < 2^{j_h+1}$. For the last one, we use the concentration on ξ_t and the decreasing assumption.

1.4.5 Experimental Result

■

1.5 Linear rotting bandits are impossible to learn

1.5.1 Linear rested rotting bandits

In this section, we present our rotting linear bandit framework which recovers 1) the linear model of as soon as the reward is stationnary; and 2) the rotting multi-armed bandits model as soon as $\mathscr{X}$ contains exclusively canonical basis vectors.

We introduce d non-increasing and L-Lipschitz functions $\mu_i : \mathbb{R} \to \mathbb{R}$. These functions satisfies Assumption **??**, but while there were K reward functions defined on $\mathbb{N}$ in the rotting MAB model, we now have d functions defined on $\mathbb{R}$. Indeed, in the linear setup the number of reward parameter is d and we expect this value to replace K in the regret bound.

We call $N_{i,t} \triangleq \sum_{t'=0}^{t}(X_{t'})_i$, which quantifies the amount of pull of direction i. We then define the reward :

$$o_t(X) = \sum_{i \leq d} \int_{N_{i,t}}^{N_{i,t+1}} \mu_i(x)dx. + \eta_t = \int_{\boldsymbol{N}_t}^{\boldsymbol{N}_{t+1}} \boldsymbol{\mu}(\boldsymbol{n})^\intercal d\boldsymbol{n} + \eta_t$$

The total reward can thus be writen:

$$J(\pi, T) = \int_{\boldsymbol{0}}^{\boldsymbol{N}_T} \boldsymbol{\mu}(\boldsymbol{n})^\intercal d\boldsymbol{n}.$$

Hence, we found a model which extends both rotting MAB model (when the actions are encoded by canonical vectors) and linear bandit model (when the reward is stationnary, i.e. $\boldsymbol{\mu}$ is a constant vector function). Moreover, for any vector $\boldsymbol{X}$, the reward associated to $\boldsymbol{X}$ is decreasing along the pulls while the cumulative reward is totally determined by the number of pull $\boldsymbol{N_T}$ and the knowledge of $\boldsymbol{\mu}$.

However, one can note that the number of pulls $N_{i,t}$ in the rotting MAB setup has two meaningfull equivalent in the rotting linear setup : $\sum_t x_{i,t}$ and $\sum_t x_{i,t}^2$. The first one is useful in the integral to have linear dependence of the reward with X. The second is usefull from an information theoretic point of view (least square regression) to quantify how much we pulled each direction.

Bandits problems are often considered as the RL problems without state. However, in this setup we do have a state as the next reward depends on the matrix A_T. In the rotting MAB framework, we overcome this issue by showing that the greedy oracle strategy is optimal. Hence, there is no need for planning and the stochastic learning problem is reduced to a pure exploration-exploitation problem where one needs to determine the action which currently performs the best. Therefore, we would like to show that the greedy oracle policy (ie. the policy which selects $\int_{\boldsymbol{N}_t}^{\boldsymbol{N}_{t+1}} \boldsymbol{\mu}(\boldsymbol{n})^\intercal d\boldsymbol{n}$) is optimal in the rotting linear bandit problem. In the next section, we will show that the greedy oracle policy is not optimal and hence that there is no anytime optimal policy.

1.6 The non-optimality of the greedy oracle policy

Theorem 1.6.1 The greedy oracle strategy π_G is not optimal. More precisely, for any horizon T, there exists a reward vector function $\vec{\mu}$ such as the performance compared to the optimal policy for horizon $T \geq 2$ π_{O_T} is :

$$J(\pi_{O_T}, T) - J(\pi_G, T) \geq \frac{L(T-1)}{8}$$

Proof. We consider $d = 2$, $\mathscr{X} = \{X_1, X_2\}$ with $X_1 = (1,0)^\intercal$ and $X_2 = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})^\intercal$. For any horizon T, we consider the following reward functions :

$$\mu_1(x) = L \text{ if } x < \frac{T}{2} \text{ else } 0 \qquad \text{and} \qquad \mu_2(x) = \frac{L}{2}.$$

The greedy strategy will therefore select X_1 until $\lfloor \frac{T}{2} \rfloor$ and then X_2 untill the end of the game. Hence :

$$J(\pi_G, T) = \int_0^{\lfloor \frac{T}{2} \rfloor + \lceil \frac{T}{2} \rceil / 2} \mu_1(x)dx + \int_0^{\lceil \frac{T}{2} \rceil / 2} \mu_2(x)dx = \frac{T}{2}L + \left\lceil \frac{T}{2} \right\rceil \frac{L}{4} \leq \frac{5LT + L}{8}.$$

We now consider the policy π_2 which always selects arm 2. At the end of the game, it gathers the reward :

$$J(\pi_2, T) = \int_0^{\frac{T}{2}} \mu_1(x)dx + \int_0^{\frac{T}{2}} \mu_2(x)dx = \frac{T}{2}L + \frac{T}{2}\frac{L}{2} = \frac{3LT}{4}$$

Hence, since optimal policy $\pi_T^\star$ has larger reward than π_2 at horizon T (by definition), we have that

$$J(\pi_T^\star, T) - J(\pi_G, T) \geq J(\pi_2, T) - J(\pi_G, T) \geq \frac{L(T-1)}{8}$$

■

Hence the greedy policy can be as bad as 8th the regret of the worst performance possible on the problem sets. This is surprising as the greedy oracle strategy was optimal for the rotting MAB problem. One can note that the vectors used in the proof have the same L_2-norm and that the vector function $\vec{\mu}$ is bounded in $[0, L]^2$. The overall setup is simple. We do not need complex decays nor vectors with different "pulling amount" to have a suboptimal performance of the greedy policy. The suboptimality comes from the fact that we do not have access to all the canonical vectors. Hence, when the greedy algorithm has collected all the reward it can get from direction 1, it will start focusing on collecting reward in the second direction. When it pulls the second vector to take advantage of the second direction it also pulls the first direction which is now useless. Here comes some "regret" : the algorithm could have started directly collecting direction 2 as it would have got all the direction 1 benefits anyway. The following corollary underlines that the failure of the greedy oracle strategy implies the necessity of planning.

Lemma 1.6.2 For a cumulative reward exploration exploitation problem, the only possible anytime optimal oracle strategy is the greedy oracle one.

Proof. Let's assume π_{O_a} an anytime optimal strategy which does not select the greedy action at time T. Let's consider π_{G_T} a strategy which copies π_{O_a} for the $T-1$ round and is greedy at round T.

$$J(\pi_{O_a}, T) - J(\pi_{G_T}, T) = r_T(\pi_{O_a}(T)) - r_T(\pi_{G_T}(T)) < 0$$

where the last inequality comes from the fact that $r_T(\pi_{O_a}(T))$ is below the best reward available for that time. ■

Corollary 1.6.3 There is no anytime optimal oracle strategy for the rotting linear bandit model.

What is the regret of a short-sighted oracle strategy which sees F steps in the future (ie . knows μ_i from 0 up to $a^2_{ii,t} + F \max_d X^2_{d,i}$?)

Theorem 1.6.4 Any strategy which can anticipate the future up to F steps in advance has a worst case regret which scales at least with $O(T-2F)$. More precisely :

$$\max_{\mu} R(\pi, T) \geq \frac{L(T-2F)}{12} - \frac{L}{6}$$

Proof. We still consider $d = 2$, $\mathscr{X} = \{X_1, X_2\}$ with $X_1 = (1,0)^\intercal$ and $X_2 = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})^\intercal$. For any horizon T, we consider the following reward functions :

$$\mu^1_1(x) = L \quad \text{and} \quad \mu^2_1(x) = L \text{ if } x < \frac{T}{2} \text{ else } 0 \quad \text{and} \quad \mu_2(x) = \frac{L}{2}.$$

The optimal strategy associated to μ^1_1 (respectively μ^2_1) is $\pi_O \triangleq \pi_1$ (resp. $\pi_O \triangleq \pi_2$), the policy which always pulls the first (resp. second) arm, and it gathers the cumulative reward $J_1(\pi_O, T)$ (resp. $J_2(\pi_O, T)$). We have by simple calculations:

$$J_1(\pi_O, T) = LT \quad \text{and} \quad J_2(\pi_O, T) = \frac{3TL}{4}$$

Depending on whether μ_1 is μ^1_1 or μ^2_1, we can express the regret as a function of $N_{1,T}$ or $N_{2,t}$.

$$R_1(\pi, T) \triangleq J_1(\pi_O, T) - J_1(\pi_{t_f}, T) = LT - L(T - N_{2,T}) - N_{2,T}\frac{3L}{4} = \frac{LN_{2,T}}{4} \tag{1.37}$$

$$R_2(\pi, T) \triangleq J_2(\pi_O, T) - J_2(\pi_{t_f}, T) = \frac{3LT}{4} - \frac{LT}{2} - (T - N_{1,T})\frac{L}{4} = \frac{LN_{1,T}}{4} \tag{1.38}$$

We call t_f the first time such that $||\varepsilon_1||_{A_{t_f}} \geq \frac{T}{2} - F$. After t_f the learner entirely knows which reward functions she faced and before t_f the two reward functions are undistinguishable to the learner. Note that for any policy, $||\varepsilon_1||_{A_T} \geq \frac{T}{2} > \frac{T}{2} - F$, hence t_f exists for any policy. We have that

$$N_{1,t_f} + \frac{N_{2,t_f}}{2} \geq \frac{T}{2} - F \tag{1.39}$$

$$N_{1,t_f} + \frac{N_{2,t_f}}{2} \leq \frac{T}{2} - F + 1 \tag{1.40}$$

$$N_{1,t_f} + N_{2,t_f} = t_f \tag{1.41}$$

Hence, we have the following lowerbound for $N_{i,T}$:

$$N_{1,T} \geq N_{1,t_f} \geq T - t_f - 2F \tag{1.42}$$
$$N_{2,T} \geq N_{2,t_f} \geq 2t_f - T + 2(F-1) \tag{1.43}$$

Hence, worst case regret is :

$$\max_{\mu} R(\pi,T) \geq \max(R_1(\pi,T), R_2(\pi,T)) = \frac{L}{4} \max_{0 \leq t_f \leq T} (T - t_f - 2F, 2t_f - T + 2(F-1)) \geq \frac{L(T-2F)}{12} - \frac{L}{6}$$

■

Note we can slightly modify the proof to get $O(T - \alpha F)$ for $\alpha > 1$.