I 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

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

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,

$$J_T(\pi) \triangleq \sum_{t=1}^{T} \mu_{\pi(t)}\left(N_{\pi(t),t-1}\right) = \sum_{i\in\mathscr{K}} \sum_{n=0}^{N_{i,T}^{\pi}-1} \mu_i(n). \tag{1.1}$$

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,

$$R_T(\pi) \triangleq J^\star - J_T(\pi). \tag{1.2}$$

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 1.1, we can conveniently rewrite the regret as

$$\begin{aligned} R_T(\pi) &= \sum_{i \in \mathscr{K}} \left(\sum_{n=0}^{N_{i,T}^{\star}-1} \mu_i(n) - \sum_{n=0}^{N_{i,T}^{\pi}-1} \mu_i(n) \right) \\ &= \sum_{i \in \mathrm{UP}} \sum_{n=N_{i,T}^{\pi}}^{N_{i,T}^{\star}-1} \mu_i(n) - \sum_{i \in \mathrm{OP}} \sum_{n=N_{i,T}^{\star}}^{N_{i,T}^{\pi}-1} \mu_i(n), \end{aligned} \tag{1.3}$$

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 Auer et al. (2002)) 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,

$$R_T(\pi) \leq \sum_{i \in \mathrm{OP}} \sum_{n=N_{i,T}^{\star}}^{N_{i,T}^{\pi}-1} \left(\mu_T^+(\pi) - \mu_i(n) \right). \tag{1.4}$$

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 Heidari et al. (2016), which has solved the offline problem and the online problem in the absence of noise, and Levine et al. (2017), which has presented the first learning policy with non trivial guarantees for rotting bandits with noise.

1.1.2 The offline problem (Heidari et al. 2016)

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 $\mathscr{A}_0$, Heidari et al. (2016))

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 — Heidari et al. (2016). 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 (Heidari et al. 2016)

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 ? Heidari et al. (2016) 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 $\mathscr{A}_2$, Heidari et al. (2016))

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 — Heidari et al. (2016). 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 1.4 applied to policy π_{G},

$$R_T(\pi_{\mathrm{G}}) \leq \sum_{i \in \mathrm{OP}} \sum_{n=N^\star_{i,T}}^{N^{\pi_{\mathrm{G}}}_{i,T}-1} \left(\mu_T^+(\pi_{\mathrm{G}}) - \mu_i(n)\right). \tag{1.5}$$

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 $n_i \leq N^{\pi_{\mathrm{G}}}_{i,T} - 2$,

$$\mu_i(n_i) \geq \mu_i(N^{\pi_{\mathrm{G}}}_{i,T} - 2) = \mu_i(N^{\pi_{\mathrm{G}}}_{i,t_i-1} - 1) \geq \mu_j(N^{\pi_{\mathrm{G}}}_{j,t_i-1} - 1). \tag{1.6}$$

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,

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

Gathering Equations 1.6 and 1.7, we have that

$$\forall n_i \leq N^{\pi_{\mathrm{G}}}_{i,T} - 2, \;\; \mu(n) \geq \mu_T^+(\pi_{\mathrm{G}}). \tag{1.8}$$

Therefore, we can upper-bound all the before last terms in each second sum in Equation 1.5 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 1.8. 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 — Heidari et al. (2016). 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 Heidari et al. (2016). 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 Heidari et al. (2016) 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 Levine et al. (2017) : 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 $\mathcal{O}(\sigma)$. 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 $\mathcal{O}(\sigma T)$ which could be much larger from the noise-free rate $\mathcal{O}(KL)$ depending on the parameters.

Levine et al. (2017) 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 $\mathcal{O}(h)$ 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 $\mathcal{O}(KLh)$ due to this bias. On the other hand, the learner will also take decision based on estimators with variance $\widetilde{\mathcal{O}}(\frac{\sigma}{\sqrt{h}})$ which may cost $\widetilde{\mathcal{O}}(\frac{\sigma T}{\sqrt{h}})$ on the long run. Choosing $h = \widetilde{\mathcal{O}}\left(\frac{\sigma T}{KL}\right)^{2/3}$, we get the regret rate of $\widetilde{\mathcal{O}}\left(L^{1/3}\sigma^{2/3}K^{1/3}T^{2/3}\right)$.

Algorithm 3 SWA (Levine et al. 2017)

Require: $\mathcal{K}$, h
Initialize $\widehat{\mu}_i^h \leftarrow +\infty$ for all $i \in \mathcal{K}$
Initialize $\mathbf{H}(i) \leftarrow []$ for all $i \in \mathcal{K}$
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 $i_t \in \arg\max_{i \in \mathcal{K}} \widehat{\mu}_i^{h\,a}$; 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 (Garivier and Moulines 2011), which slides for all arms at every round.

Analysis

The analysis of Levine et al. (2017) 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

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

and the average of the associated means as

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

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

$$\xi_{\text{SWA}} \triangleq \left\{\forall t \in \{Kh+1,\dots,T\}, \forall i \in \mathscr{K}, \forall n \in \{h,\dots,t-1\}, \left|\widehat{\mu}_i^h(n) - \overline{\mu}_i^h(n)\right| \leq c(h,\delta_T)\right\} \tag{1.9}$$

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 Levine et al. (2017) 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 — Levine et al. (2017). 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,

$$\mathbb{E}\left[R_T(\pi_{\text{SWA}})\right] \leq \mathbb{E}\left[\mathbb{1}\left[\xi_{\text{SWA}}\right] R_T(\pi_{\text{SWA}})\right] + KB. \tag{1.10}$$

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 1.4 applied to policy `SWA`,

$$R_T(\pi_{\text{SWA}}|\xi_{\text{SWA}}) \leq \mathbb{1}\left[\xi_{\text{SWA}}\right] \sum_{i \in \text{OP}} \sum_{n=N^\star_{i,T}}^{N_{i,T}^{\pi_{\text{SWA}}}-1} \left(\mu_T^+(\pi_{\text{SWA}}) - \mu_i(n)\right). \tag{1.11}$$

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 $n_i \leq N_{i,T}^{\pi_{\text{SWA}}} - (h+1)$,

$$\begin{aligned}
\mu_i(n_i) &\geq \mu_i(N_{i,T}^{\pi_{\text{SWA}}} - (h+1)) \\
&\geq \overline{\mu}_i^h(N_{i,t_i-1}^{\pi_{\text{SWA}}}) \\
&\geq \widehat{\mu}_i^h(N_{i,t_i-1}^{\pi_{\text{SWA}}}) - c(h,\delta_T) \\
&\geq \widehat{\mu}_j^h(N_{j,t_i-1}^{\pi_{\text{SWA}}}) - c(h,\delta_T) \\
&\geq \overline{\mu}_j^h(N_{j,t_i-1}^{\pi_{\text{SWA}}}) - 2c(h,\delta_T).
\end{aligned} \tag{1.12}$$

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,

$$\overline{\mu}_j^h(N_{j,t_i-1}^{\pi_{\text{SWA}}}) \geq \mu_j(N_{j,T}^{\pi_{\text{SWA}}}) = \mu_T^+(\pi_{\text{SWA}}). \tag{1.13}$$

Gathering Equations 1.12 and 1.13, we have that

$$\forall n_i \leq N_{i,T}^{\pi_{\text{SWA}}} - (h+1), \ \left(\mu_T^+(\pi_{\text{SWA}}) - \mu_i(n_i)\right) \leq 2c(h, \delta_T). \tag{1.14}$$

Therefore, in Equation 1.11, we can split the sum on $N_{i,T}^{\pi_{\text{SWA}}} - h$. Hence,

$$\begin{aligned}
R_T(\pi_{\text{SWA}}|\xi_{\text{SWA}}) \leq & \mathbb{1}\left[\xi_{\text{SWA}}\right] \sum_{i \in \text{OP}} \sum_{n=N_{i,T}^\star}^{N_{i,T}^{\pi_{\text{SWA}}}-1} \left(\mu_T^+(\pi_{\text{SWA}}) - \mu_i(n)\right) \\
= & \mathbb{1}\left[\xi_{\text{SWA}}\right] \sum_{i \in \text{OP}} \sum_{n=N_{i,T}^\star}^{N_{i,T}^{\pi_{\text{SWA}}}-(h+1)} \left(\mu_T^+(\pi_{\text{SWA}}) - \mu_i(n)\right) \\
& + \mathbb{1}\left[\xi_{\text{SWA}}\right] \sum_{i \in \text{OP}} \sum_{n=N_{i,T}^{\pi_{\text{SWA}}}-h}^{N_{i,T}^{\pi_{\text{SWA}}}-1} \left(\mu_T^+(\pi_{\text{SWA}}) - \mu_i(n)\right) \\
\leq & 2Tc(h, \delta_T) + KhB.
\end{aligned} \tag{1.15}$$

In the last inequality, we used Equation 1.14 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 1.15 in Equation 1.10 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 — Levine et al. (2017). 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 Levine et al. (2017) 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 $\frac{B}{\sigma}$). Levine et al. (2017) 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 Levine et al. (2017) (without the small modification of the tuning h) .

Algorithm 4 wSWA (Levine et al. 2017)

Require: α, σ, $T_0 \leftarrow 1$
1: $T \leftarrow T_0$
2: $h \leftarrow \left\lceil \alpha \left(\frac{4\sigma T}{K}\right)^{2/3} \left(\log\left(\sqrt{2}T\right)\right)^{1/3} \right\rceil$
3: **for** $t \leftarrow 1, 2, \ldots, T$ **do**
4: RUN SWA (h)
5: **end for**
6: CLEAN SWA'S MEMORY
7: 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, Levine et al. (2017) 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, Heidari et al. (2016) show that in the absence of noise, the regret is lower bounded by $\mathscr{O}(KL)$. Second, we recall We start 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, Levine et al. (2017) 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

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.

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 5 FEWA

Require: $\mathscr{K}$, σ, δ_0, α
1: Initialize $\mathbf{H}(i) \leftarrow []$ for all $i \in \mathscr{K}$
2: **for** $t \leftarrow 1, 2, \ldots, K$ **do** ▷ *Pull each arm once*
3: PULL $i_t \leftarrow t$; RECEIVE o_t
4: $\mathbf{H}(i) \leftarrow \mathbf{H}(i)$.append$(o_t)$
5: $N_{i_t} \leftarrow 1$
6: **end for**
7: **for** $t \leftarrow K+1, K+2, \ldots$ **do**
8: $\delta_t \leftarrow \delta_0/(t^\alpha)$
9: $h \leftarrow 1$ ▷ *initialize bandwidth*
10: $\mathscr{K}_1 \leftarrow \mathscr{K}$ ▷ *initialize with all the arms*
11: $i_t \leftarrow$ none
12: **while** i_t is none **do**
13: $\mathscr{K}_{h+1} \leftarrow$ FILTER$(\mathscr{K}_h, h, \delta_t)$
14: $h \leftarrow h+1$
15: **if** $\exists i \in \mathscr{K}_h$ such that $N_{i_t} = h$ **then**
16: PULL $i_t \in \{i \in \mathscr{K}_h | N_{i_t} = h\}$[a]; RECEIVE o_t
17: **end if**
18: **end while**
19: $\mathbf{H}(i) \leftarrow \mathbf{H}(i)$.append$(o_t)$
20: $N_{i_t} \leftarrow N_{i_t} + 1$
21: **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, and a confidence parameter δ as input and returns an updated set of arms $\mathscr{K}_{h+1}$. For each arm i that has been pulled n times, the algorithm constructs an estimate $\widehat{\mu}_i^h(n)$ that averages the $h \leq n$ most recent rewards observed from i. The subroutine FILTER discards all the arms whose mean estimate (built with window h) from $\mathscr{K}_h$ 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. **??**).

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 15 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 $c(h, \sigma, \delta_t)$ is large for small h (i.e., as few samples are used in constructing $\widehat{\mu}_i^h(N_{i,t})$, the estimation error may be high). Thus, FEWA keeps refining $\mathscr{K}_h$ for larger windows in the

Algorithm 6 FILTER

Require: $\mathscr{K}_h$, h, δ_t, σ
1: $c(h,\delta_t) \leftarrow \sqrt{(2\sigma^2/h)\log(1/\delta_t)}$
2: **for** $i \in \mathscr{K}_h$ **do**
3: $\quad \widehat{\mu}_i^h \leftarrow \text{MEAN}(\mathbf{H}(i_t)[-h:])$
4: **end for**
5: $\widehat{\mu}_{\max}^h \leftarrow \max_{i\in\mathscr{K}_h} \widehat{\mu}_i^h$
6: **for** $i \in \mathscr{K}_h$ **do**
7: $\quad \Delta_i \leftarrow \widehat{\mu}_{\max}^h - \widehat{\mu}_i^h$
8: $\quad$ **if** $\Delta_i \leq 2c(h,\delta_t)$ **then**
9: $\quad\quad$ add i to $\mathscr{K}_{h+1}$
10: $\quad$ **end if**
11: **end for**
Ensure: $\mathscr{K}_{h+1}$

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 15). At this point, increasing h would not bring any additional evidence that could refine $\mathscr{K}_h$ further (recall that $\widehat{\mu}_i^h(N_{i,t})$ is not defined for $h > N_{i,t}$). Finally, FEWA selects the active arm $i(t)$ whose number of samples matches the current window, i.e., the least pulled arm in $\mathscr{K}_h$. The set of available rewards and the number of pulls are then updated accordingly.

Active set

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

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

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

Lemma 1.2.1 On the favorable event ξ_t, if an arm i passes through a filter of window h at round t, i.e., $i \in \mathscr{K}_h$, then the average of its h last pulls satisfies

$$\overline{\mu}_i^h(N_{i,t}^{\pi_F}) \geq \mu_t^+(\pi_F) - 4c(h,\delta_t). \tag{1.16}$$

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

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

Require: $(\delta_t)_{t\geq 1}$
1: pull each arm once
2: initialize $N_{i,K} \leftarrow 1$
3: **for** $t \leftarrow K+1, K+2, \ldots$ **do**
4: $\quad i_t \leftarrow \arg\max_i \mathrm{ind}(i,t,\delta_t)$ ▷ *cf.* (1.18)
5: $\quad$ receive reward o_t
6: $\quad N_{i_t,t} \leftarrow N_{i_t,t-1} + 1 \quad N_{j,t} \leftarrow N_{j,t-1}, \quad \forall j \neq i_t$
7: **end for**

Figure 1.1: The RAW-UCB algorithm

While the inequality above for i_t^* trivially satisfies Eq. 1.16, Lem. 1.2.3 is proved by integrating the possible errors introduced by the filter in selecting active arms due to the error of the empirical estimates.

Corollary 1.2.2 Let $i \in \mathrm{OP}$ be an arm overpulled by FEWA at round t and $h_{i,t} \triangleq N_{i,t}^{\pi_F} - N_{i,t}^{\pi^\star} \geq 1$ be the difference in the number of pulls w.r.t. the optimal policy $\pi^\star$ at round t. On the favorable event ξ_t, we have

$$\mu_t^+(\pi_F) - \overline{\mu}_i^{h_{i,t}}(N_{i,t}) \leq 4c(h_{i,t},\delta_t). \tag{1.17}$$

1.2.3 The RAW-UCB algorithm

A general index policy

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

$$\mathrm{ind}(i,t,\delta_t) \triangleq \min_{h \leq N_{i,t-1}} \widehat{\mu}_i^h(t-1,\pi) + c(h,\delta_t). \tag{1.18}$$

We set $\delta_t \triangleq \frac{1}{t^\alpha}$ 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.

Lemma 1.2.3 At round t on favorable event ξ_t, if arm i_t is selected, 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.,

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

This fundamental guarantee is comparable with Corollary about the algorithm FEWA. FEWA uses the same statistics than RAW-UCB but in a rather complex expanding filtering mechanism. 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.

1.3 Linear rotting bandits are impossible to learn

1.3.1 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.

1.4 The non-optimality of the greedy oracle policy

Theorem 1.4.1 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 1.4.2 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 1.4.3 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 1.4.4 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_1^2, we can express the regret as a function of $N_{1,T}$ or $N_{2,t}$.

$$R_1(\pi,T) \triangleq J_1(\pi_O,T) - J_1(\pi_{t_f},T) = LT - L(T-N_{2,T}) - N_{2,T}\frac{3L}{4} = \frac{LN_{2,T}}{4} \quad (1.19)$$

$$R_2(\pi,T) \triangleq J_2(\pi_O,T) - J_2(\pi_{t_f},T) = \frac{3LT}{4} - \frac{LT}{2} - (T-N_{1,T})\frac{L}{4} = \frac{LN_{1,T}}{4} \quad (1.20)$$

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

$$N_{1,t_f} + \frac{N_{2,t_f}}{2} \geq \frac{T}{2} - F \quad (1.21)$$

$$N_{1,t_f} + \frac{N_{2,t_f}}{2} \leq \frac{T}{2} - F + 1 \quad (1.22)$$

$$N_{1,t_f} + N_{2,t_f} = t_f \quad (1.23)$$

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

$$N_{1,T} \geq N_{1,t_f} \geq T - t_f - 2F \quad (1.24)$$

$$N_{2,T} \geq N_{2,t_f} \geq 2t_f - T + 2(F-1) \quad (1.25)$$

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 $\alpha > 1$.