Online Learning with Bounded Recall

Jon Schneider [*1] **Kiran Vodrahalli** [*1]

Abstract

We study the problem of full-information online learning in the "bounded recall" setting popular in the study of repeated games. An online learning algorithm $\mathcal{A}$ is M*-bounded-recall* if its output at time t can be written as a function of the M previous rewards (and not e.g. any other internal state of $\mathcal{A}$). We first demonstrate that a natural approach to constructing bounded-recall algorithms from mean-based no-regret learning algorithms (e.g., running Hedge over the last M rounds) fails, and that any such algorithm incurs constant regret per round. We then construct a stationary bounded-recall algorithm that achieves a per-round regret of $\Theta(1/\sqrt{M})$, which we complement with a tight lower bound. Finally, we show that unlike the perfect recall setting, any low regret bound bounded-recall algorithm must be aware of the ordering of the past M losses – any bounded-recall algorithm which plays a symmetric function of the past M losses must incur constant regret per round.

1. Introduction

Online learning is the study of online decision making, and is one of the cornerstones of modern machine learning, with numerous applications across computer science, machine learning, and the sciences.

Traditionally, most online learning algorithms depend (either directly or indirectly) on the entire history of rewards seen up to the present. For example, each round, the Multiplicative Weights Update method decides to play action i with probability proportional to some exponential of the cumulative reward of action i until the present (in fact, the rewards of the first round have as much impact on the action chosen at round t as the rewards of round $t-1$).

[*]Equal contribution [1]Google. Correspondence to: Jon Schneider <jschnei@google.com>, Kiran Vodrahalli <kirannv@google.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

Another natural setting for learning is to restrict the learner so that they can only use information they received from the past M rounds of play. This is a restriction that is known in the game theoretic literature as *bounded recall*, and was introduced by Aumann & Sorin (1989) in an attempt to design a model of repeated play that could lead to altruistic equilibria. However, there are multiple other reasons why bounded recall is interesting to study from a learning angle, including:

- **Recency bias and modelling human behavior**: It is increasingly common in the economics and social sciences literature to model rational agents as low-regret learners (e.g. Blum & Mansour (2007b); Nekipelov et al. (2015); Braverman et al. (2018)). However, the extent to which standard low-regret algorithms actually model human behavior (especially over longer timescales) is unclear. Indeed, "recency bias" is a well-known psychological phenomenon in human decision-making indicating that people do, in general, prioritize more recent information. Bounded recall dynamics could be a more accurate model of human behavior than general no-regret dynamics.
- **Designing adaptive learning algorithms:** On the flip side, one may wish to design learning algorithms that *do* prioritize information from more recent rounds. While there exist learning paradigms that accomplish this without imposing the bounded-recall constraint (notably, algorithms that minimize adaptive / dynamic regret: see Bousquet & Warmuth (2002); Zhang et al. (2018); Zheng et al. (2019)), it is also valuable to understand the black box procedure of running an arbitrary learning algorithm over a sliding window.
- **Sequence models:** Decisions made by attention-based sequence models (Vaswani et al., 2017), which are popularly employed by modern large language models (Achiam et al., 2023; Team et al., 2023), fall naturally into the bounded recall paradigm: these models have a fixed context window of past rounds (corresponding to tokens) that they can observe, but the models can also be used to operate on streams of tokens far exceeding their context window. As decisions are increasingly entrusted to sequence models, it is important to understand their theoretical capabilities as online learning

algorithms.

- **Privacy and data retention**: Finally, the rights of users to control the use of their data are becoming ever more common worldwide, with regulations like the GDPR's "right to be forgotten" (Voigt & Von dem Bussche, 2017) influencing how organizations store and think about data. For instance, it is increasingly recognized that it is not enough to simply remove explicit storage of data, but it is also necessary to remove the impact of this data on any models that were trained using it (motivating the study of machine unlearning, see e.g. Ginart et al. (2019)).

  Similarly, many organizations have a blanket policy of erasing any data that lies outside some retention window; such organizations must then train their model with data that lies within this window. Note that this is an example of a scenario where it is not sufficient to simply prioritize newer data; we must actively exclude data that is sufficiently old.

Motivated by these examples, in this paper we study *bounded-recall* learning algorithms: algorithms that can only use information received in the last M rounds when making their decisions. In particular, we study this problem in the classical full-information online learning model, where every round t (for T rounds) a learner must play a distribution x_t over d possible actions[1]. Upon doing this, the learner receives a reward vector r_t from the adversary, and receives an expected reward of $\langle x_t, r_t \rangle$. The learner's goal is to minimize their regret (the difference between their utility and the utility of the best fixed action).

It is well-known that without any restriction on the learning algorithm, there exist low-regret learning algorithms (e.g. Hedge) which incur at most $O(\sqrt{(\log d)/T})$ regret per round, and moreover this is tight; any algorithm must incur at least $\Omega(\sqrt{(\log d)/T})$ regret on some online learning instance. We begin by asking what regret bounds are possible for bounded-recall algorithms. In Section 3 we show that (in analogy with the unrestricted case), any bounded-recall learning algorithm must incur at least $\Omega(\sqrt{(\log d)/M})$ regret per round. Moreover, there is a very simple bounded-recall algorithm which achieves this regret bound: simply take an optimal unrestricted learning algorithm and restart it every M rounds (Algorithm 1).

However, this periodically restarting algorithm has some undesirable qualities – for example, even when playing it on time-invariant stochastic data, the performance of this algorithm periodically gets worse every M rounds right after it restarts (it also e.g. does not seem like a particularly natural algorithm for modeling human behavior). Given this, we introduce the notion of *stationary* bounded-recall algorithms, where the action taken by the learner at time t must be a fixed function of the rewards in rounds $t-M$ through $t-1$; in particular, this function cannot depend on the value t of the current round (and rules out the periodically restarting algorithm).

One of the most natural methods for constructing a stationary bounded-recall algorithm is to take an unrestricted low-regret algorithm $\mathcal{A}$ of your choice and each round, rerun it over only the last M rewards. Does this procedure result in a low-regret stationary bounded-recall algorithm? We prove a two-pronged result:

- If the low-regret algorithm $\mathcal{A}$ belongs to a class of algorithms known as *mean-based algorithms*, then there exists an online learning instance where the resulting bounded-recall algorithm incurs a *constant* (independent of M and T) regret per round.

  The class of mean-based algorithms includes most common online learning algorithms (including Hedge and FTRL); intuitively, it contains any algorithm which will play action i with high probability if historically, action i performs linearly better than any other action.

- However, there *does* exist a low-regret algorithm $\mathcal{A}$ such that the resulting stationary bounded-recall algorithm incurs an optimal regret of at most $O(\sqrt{(\log d)/M})$ per round.

  We call the resulting bounded-recall algorithm the "average restart" algorithm and it works as follows: first choose a random starting point s uniformly at random between $t-M$ and $t-1$. Then, run a low-regret algorithm of your choice (e.g. Hedge) only on the rewards from rounds s to $t-1$.

  The underlying full-horizon algorithm can be obtained by setting $M = T$. Interestingly, as far as we are aware this appears to be a novel low-regret (non-mean-based) algorithm, and may potentially be of use in other settings where mean-based algorithms are shown to fail (e.g. Braverman et al. (2018)).

- In contrast with standard, full-horizon multiplicative weights (which assign equal importance to all previous rounds), the "average restart" algorithm we construct treats the set of rounds it can observe highly asymmetrically, putting far more weight on more recent rounds. We show that this asymmetry is essential in the bounded recall setting: any symmetric, stationary bounded-recall algorithm must incur high per-round regret.

[1] Throughout this paper we focus entirely on the learning with experts setting (where the action set is the d-simplex), but all observations / theorems in this paper should extend easily to other full-information online learning settings, such as online convex optimization. It is an interesting question to understand whether it is possible to adapt these results for partial-information settings such as bandits.

Finally, we run simulations of the algorithms mentioned above on a variety of different online learning instances. We observe that in time-varying settings, the bounded-recall algorithms we develop often outperform their standard online learning counterparts.

Related work. The idea of bounded-recall dynamics in economics appears to have been introduced by either Aumann & Sorin (1989) or Lehrer (1988). Since then, there has been a large economic literature on studying games under bounded-recall dynamics and repeated games where agents have bounded memory (e.g., with strategy encoded by a finite-state machine). See Neyman (1997) for a partial survey of the area. See also Drenska & Calder (2023) for a related but different problem of getting low regret compared to a class of benchmarks which are bounded recall (instead of designing a learning algorithm which must be bounded recall). Qiao & Valiant (2021) have also considered a bounded-recall setup in the selective learning problem setting. Most relevant to this paper is the work of Zapechelnyuk (2008), who shows (in a similar proof to our Theorem 4.1) that "better-reply dynamics" in bounded recall settings can lead to high regret ("better-reply dynamics" are a generalization of regret matching algorithms, and are a more constrained class than the set of mean-based algorithms we study). The fact that PERIODICRESTART achieves low-regret is a folklore result in both online learning and repeated games. To the best of our knowledge, the algorithms AVERAGERESTART and AVERAGERESTARTFULLHORIZON have not been considered before in either literature.

We defer discussion of additional related work to Appendix A.

2. Model and Preliminaries

We consider a full-information online learning setting where a learner (running algorithm $\mathcal{A}$) must output a distribution $x_t \in \Delta([d])$ over d actions each round t for T rounds. Initially, an oblivious[2] adversary selects a sequence $\mathbf{r} \in [0,1]^{T\times d}$ of reward vectors $\{r_t\}_{t\in[T]}$ where $r_{t,i}$ represents the reward of action i in round t. Then, in each round t, the learner selects their distribution $x_t \in \Delta([d])$ as a function of the reward vectors $r_1, r_2, \ldots, r_{t-1}$ (we write this as $x_t = \mathcal{A}(r_1, r_2, \ldots, r_{t-1})$). The learner then receives utility $\langle x_t, r_t\rangle$, and observes the full reward vector r_t. We evaluate our performance via the *per-round regret*:

Definition 2.1 (Per-round Regret)**.** The per-round regret of an online learning algorithm $\mathcal{A}$ on a learning instance $\mathbf{r}$ is given by

$$\mathsf{Reg}(\mathcal{A};\mathbf{r}) := \frac{1}{T}\left(\max_{i\in[d]}\sum_{t=1}^{T} r_{t,i} - \sum_{t=1}^{T}\langle x_t, r_t\rangle\right).$$

In other words, $\mathsf{Reg}(\mathcal{A};\mathbf{r})$ represents the (amortized) difference in performance between algorithm $\mathcal{A}$ and the best action in hindsight on instance $\mathbf{r}$. Where $\mathbf{r}$ is clear from context we will omit it and write this simply as $\mathsf{Reg}(\mathcal{A})$.

Throughout this paper we will primarily be concerned with online learning algorithms that are *bounded-recall*; that is, algorithms whose decision can only depend on a subset of recent rewards. Formally, we define this as follows:

Definition 2.2 (Bounded-Recall Online Learning Algorithms)**.** An online learning algorithm $\mathcal{A}$ is *M-bounded-recall* if its output x_t at round t can be written in the form

$$x_t = f_t(r_{t-M}, \ldots, r_{t-1})$$

for some fixed function f_t depending only on $\mathcal{A}$ (here we take $r_i = 0$ when $i \le 0$); in other words, the output at time t depends only on t and the rewards from the past M rounds (the *history window*). If furthermore we have that f_t is independent of t, we say that $\mathcal{A}$ is a *stationary M-bounded-recall* algorithm.

In general, we will consider the setting where both M and T go to infinity, with $T \gg M$ (although many of our results still hold for $T = \Theta(M)$). We say a (M-bounded-recall) learning algorithm is *low-regret* if $\mathsf{Reg}(\mathcal{A}) = o(1)$ (here we allow $o(1)$ to depend on M; i.e., we will consider an algorithm with $\mathsf{Reg}(\mathcal{A}) = O(1/\sqrt{M})$ to be low-regret).

2.1. Mean-based learners

One of the most natural approaches to constructing a stationary bounded-recall learner $\mathcal{A}$ is to take a low-regret learning algorithm $\mathcal{A}'$ for the full-horizon setting (e.g. the Hedge algorithm) but only run it over the most recent M rounds. Later, we show that for a wide range of natural low-regret learning algorithms $\mathcal{A}'$ – e.g., Hedge, follow the perturbed/regularized leader, multiplicative weights, etc. – this results in a bounded-recall algorithm $\mathcal{A}$ with *linear regret*.

All these algorithms have the property that they are *mean-based* algorithms. Intuitively, an algorithm is mean-based if it approximately best responds to the history so far (i.e., if one arm i has historically performed better than all other arms, the learning algorithm should play i with weight near 1). Formally, we define this as follows.

Definition 2.3 (Mean-based algorithm)**.** For each $i \in [d]$ and $1 \le t \le T$, let $R_{t,i} = \sum_{s=1}^{t} r_{s,i}$. A learning algorithm

[2]Note that all results we obtain for our deterministic algorithms immediately extend to the adaptive setting, since in the full-information setting an oblivious adversary can already perfectly predict what we are going to play in any round.

$\mathcal{A}$ is γ-*mean-based* if, whenever $R_{t,i} - R_{t,j} > \gamma T$, $x_{t,j} < \gamma$. A learning algorithm is *mean-based* if it is γ-mean-based for some $\gamma = o(1)$.

Braverman et al. (2018) show that many standard low-regret algorithms (including Hedge, Multiplicative Weights, Follow the Perturbed Leader, EXP3, etc.) are mean-based.

We say a M-bounded-recall algorithm is M-*mean-based* if its output x_t in round t is of the form $\mathcal{A}_t(r_{t-M}, \ldots, r_{t-2}, r_{t-1})$ for some mean-based algorithm $\mathcal{A}_t$ (note that we allow for the choice of mean-based algorithm to differ from round to round; if we wish to construct a stationary M-bounded-recall algorithm, $\mathcal{A}_t$ should be the same for all t).

3. Benchmarks for bounded-recall learning

We begin by showing that all M-bounded-recall learning algorithms must incur at least $\Omega(\sqrt{(\log d)/M})$ regret per round. Intuitively, this follows for the same reason as the $\Omega(1/\sqrt{T})$ regret lower bounds for ordinary learning: a learner cannot distinguish between reward signals with mean $1/2$ and with mean $1/2 + \sqrt{1/M}$ with $o(M)$ samples.

Theorem 3.1 (Lower Bound)**.** *Fix an $M > 0$. Then for any M-bounded-recall learning algorithm $\mathcal{A}$ and $T > M$, there exists a distribution $\mathcal{D}$ over online learning instances $\mathbf{r}$ of length T with d actions such that*

$$\mathbb{E}_{\mathbf{r}\sim\mathcal{D}}[\mathsf{Reg}(\mathcal{A}; \mathbf{r})] \geq \Omega\left(\sqrt{\frac{\log d}{M}}\right).$$

Proof. See Appendix. □

Next, we show that we can achieve the regret bound of Theorem 3.1 with a very simple bounded-recall algorithm family which we call the PERIODICRESTART algorithm (Algorithm 1).

Algorithm 1 PERIODICRESTART

Require: Time horizon T, history window M, no-regret algorithm $\mathcal{A}$.
1: **for** $t = 1 \to T$ **do**
2: $\quad s := \lfloor t/M \rfloor \cdot M$
3: $\quad$ Play action $x_t = \mathcal{A}(r_s, r_{s+1}, \ldots, r_t)$.
4: **end for**

Note that in PERIODICRESTART, x_t depends on at most the previous M rounds and t. Note too that it is straightforward to implement PERIODICRESTART given an implementation of $\mathcal{A}$; it suffices to simply run $\mathcal{A}$, restarting its state to the initial state every M rounds (hence the name of the algorithm).

Theorem 3.2. *Assume the algorithm $\mathcal{A}$ has the property that $\mathsf{Reg}(\mathcal{A}; \mathbf{r}) \leq R(T, d)$ for any online learning instance $\mathbf{r} \in [0,1]^{T\times d}$. Then, for any instance $\mathbf{r} \in [0,1]^{T\times d}$*

$$\mathsf{Reg}(\text{PERIODICRESTART}; \mathbf{r}) \leq R(M, d).$$

Proof. We will show that the guarantees on $\mathcal{A}$ imply that the per-round regret of PERIODICRESTART over a segment of length M where $\mathcal{A}$ does not restart is at most $R(T, d)$ (from which the theorem follows). Let $i^* = \arg\max_{i\in[d]} \sum_t r_{t,i}$. Since $\mathsf{Reg}(\mathcal{A}; \mathbf{r}) \leq R(T, d)$, for any $0 \leq n \leq \lfloor T/M \rfloor$, it is the case that

$$\begin{aligned}\sum_{t=nM+1}^{(n+1)M} \langle x_t, r_t\rangle &= \sum_{t=nM+1}^{(n+1)M} \langle \mathcal{A}(x_{nM+1}, \ldots, x_t), r_t\rangle \\ &\geq \left(\sum_{t=nM+1}^{(n+1)M} r_{t,i^*}\right) - M \cdot R(M, d).\end{aligned}$$

Summing this over all $t \in [T]$, it follows that $\mathsf{Reg}(\text{PERIODICRESTART}; \mathbf{r}) \leq R(M, d)$, as desired.

□

As a corollary of Theorem 3.2, we see there exist bounded-recall algorithms with per-round regret of $O(\sqrt{(\log d)/M})$.

Corollary 3.3. *For any $M > 0$, $d \geq 2$, there exists a bounded-recall algorithm $\mathcal{A}$ such that for any online learning instance $\mathbf{r} \in [0,1]^{T\times d}$, $\mathsf{Reg}(\mathcal{A}; \mathbf{r}) \leq O(\sqrt{(\log d)/M})$.*

Proof. Run PERIODICRESTART with the HEDGE algorithm, which has the guarantee that $\mathsf{Reg}(\text{HEDGE}; \mathbf{r}) \leq \sqrt{\frac{\log d}{T}}$ for any $\mathbf{r} \in [0,1]^{T\times d}$ (see e.g. Arora et al. (2012)). □

4. Bounded-recall mean-based algorithms have high regret

As mentioned earlier, one of the most natural strategies for constructing a stationary bounded-recall algorithm is to run a no-regret algorithm $\mathcal{A}$ of your choice on the rewards from the past M rounds. In this section, we show that if $\mathcal{A}$ belongs to the large class of mean-based algorithms, this does not work – that is, we show any M-mean based algorithm incurs a constant amount of per-round regret.

Theorem 4.1. *Fix any $M > 0$, and let $\mathcal{A}$ be an M-mean-based algorithm. Then for any $T \geq 3M$, there exists an online learning instance $\mathbf{r} \in [0,1]^{2T}$ with two actions where $\mathsf{Reg}(\mathcal{A}; \mathbf{r}) \geq 1/18 - o(1)$. In particular, for sufficiently large T, $\mathsf{Reg}(\mathcal{A}; \mathbf{r}) = \Omega(1)$ (i.e., is at least a constant independent of T and M).*

The core idea behind the proof of Theorem 4.1 will be the following example, where we construct an instance of length $3M$ where any M-mean-based algorithm incurs regret at least $\Omega(M)$.

Lemma 4.2. *Fix any $M > 0$, let $T = 3M$, and let $\mathcal{A}$ be an M-mean-based algorithm. There exists an online learning instance $\mathbf{r} \in [0,1]^{2T}$ with two actions where $T \cdot \mathsf{Reg}(\mathcal{A}; \mathbf{r}) \geq M/6 - o(M)$.*

Proof. Consider the following instance $\mathbf{r}$:

- For $1 \leq t \leq M$, $r_t = (1, 0)$.
- For $M < t \leq 5M/3$, $r_t = (0, 1)$.
- For $5M/3 < t \leq 2M$, $r_t = (1, 0)$.
- For $2M < t \leq 3M$, $r_t = (0, 0)$.

For each t, let $R_{t,1} = \sum_{s=t-M}^{t-1} r_{t,1}$ and let $R_{t,2} = \sum_{s=t-M}^{t-1} r_{t,2}$ (letting $r_t = 0$ for $t \leq 0$). Since $\mathcal{A}$ is M-mean-based, there exists a γ (which is $o(1)$ w.r.t. T) such that if $R_{t,1} - R_{t,2} > \gamma T$, then $x_{t,1} \geq 1 - \gamma$, and likewise if $R_{t,2} - R_{t,1} > \gamma T$, then $x_{t,2} \geq 1 - \gamma$. Let $\Delta_t = R_{t,1} - R_{t,2}$. We then have that:

- For $1 \leq t \leq M$, $\Delta_t = t$.
- For $M < t \leq 5M/3$, $\Delta_t = M - 2(t - M)$.
- For $5M/3 < t \leq 2M$, $\Delta_t = -M/3$.
- For $2M < t \leq 8M/3$, $\Delta_t = -M/3 + (t - 2M)$.
- For $8M/3 < t \leq 3M$, $\Delta_t = M/3 - (t - 8M/3)$.

For a visualization, see Figure 1.

Now, let $\mathcal{P} = \{t \in [T] \mid \Delta_t \geq \gamma T\}$, let $\mathcal{N} = \{t \in [T] \mid \Delta_t \leq -\gamma T\}$, and let $\mathcal{Z} = \{t \in [T] \mid t \notin \mathcal{P} \cup \mathcal{N}\}$. As previously discussed, for $t \in \mathcal{P}$, $x_{t,1} \geq 1 - \gamma$, and for $t \in \mathcal{N}$, $x_{t,2} \geq 1 - \gamma$. It follows that the total reward obtained by $\mathcal{A}$ over $\mathbf{r}$ is at most

$$\sum_{t=1}^{T} \langle r_t, x_t \rangle \leq \left(\sum_{t \in \mathcal{P}} r_{t,1} + \sum_{t \in \mathcal{N}} r_{t,2} \right) + \gamma T + |\mathcal{Z}|. \quad (1)$$

From our characterization above of Δ_t, we know that:

$$\mathcal{P} = \left[\gamma T, \frac{3M}{2} - \frac{\gamma}{2} T \right] \bigcup \left[\frac{7M}{3} + \gamma T, 3M - \gamma T \right],$$

and

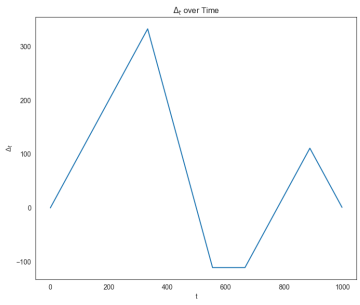


Figure 1. A plot of Δ_t over time, as used in Lemma 4.2.

$$\mathcal{N} = \left[\frac{3M}{2} + \frac{\gamma}{2} T, \frac{7M}{3} - \gamma T \right].$$

Combining this with the description of the instance $\mathcal{A}$, we can see that

$$\sum_{t \in \mathcal{P}} r_{t,1} = M - \gamma T, \ \sum_{t \in \mathcal{N}} r_{t,2} = \frac{M}{6} - \frac{\gamma}{2} T.$$

and that $|\mathcal{Z}| \leq 5\gamma T$. Expression (1) for the reward of the learner then becomes

$$\sum_{t=1}^{T} \langle r_t, x_t \rangle \leq \frac{7M}{6} + \frac{9\gamma}{2} T = \frac{7M}{6} + o(T). \quad (2)$$

On the other hand, the optimal action in hindsight is action 1, and $\sum_{t=1}^{T} r_{t,1} = \frac{4M}{3}$. It follows that

$$T \cdot \mathsf{Reg}(\mathcal{A}; \mathbf{r}) = \sum_{t=1}^{T} r_{t,1} - \sum_{t=1}^{T} \langle r_t, x_t \rangle \geq \frac{M}{18} - o(M).$$

□

Remark 4.3. It is valuable to compare the example in Lemma 4.2 to the example provided by Zapechelnyuk (2008) in their Theorem 4 (proving high-regret against better-reply dynamics). Like us, their example can be broken down in to a small number of phases of distinct behavior; however, they require the adversary to adapt to the learner's actions, and their example does not clearly generalize to mean-based algorithms.

We now apply Lemma 4.2 to prove Theorem 4.1.

Proof of Theorem 4.1. Let $\mathbf{r}^{(M)} \in [0,1]^{2 \cdot (3M)}$ be the counterexample constructed in Lemma 4.2, and let $N = \lfloor T/3M \rfloor$. Construct $\mathbf{r} \in [0,1]^T$ by concatenating N

copies of $\mathbf{r}^{(M)}$ and setting all other rewards to 0 (i.e., $\mathbf{r}_{t,i} = \mathbf{r}^{(M)}_{(t \bmod 3M),i}$ for $t \le 3MN$, $\mathbf{r}_{t,i} = 0$ for $t > 3MN$).

Fix any $1 \le n \le N$, and consider the regret incurred by $\mathcal{A}$ on rounds $t \in [(n-1) \cdot 3M + 1, n \cdot 3M]$ (the nth copy of $\mathbf{r}^{(M)}$). Since $\mathbf{r}^{(M)}$ ends with M zeros, $\mathcal{A}$ will behave identically on these rounds as it would in the first T rounds. Thus, by Lemma 4.2, $\mathcal{A}$ incurs regret at least $M/6 - o(M)$ on these rounds, and at least $NM/6 - o(T)$ regret in total. The per-round regret of $\mathcal{A}$ is thus at least $\mathsf{Reg}(\mathcal{A}; \mathbf{r}) \ge \frac{1}{T}\left(\frac{NM}{6} - o(T)\right) \ge \frac{1}{18} - o(1)$. □

5. Stationary Bounded-Recall Algorithms

5.1. Averaging over restarts

In the previous section, we showed that running a mean-based learning algorithm over the restricted history does not result in a low-regret stationary bounded-recall learning algorithm. But are there *non-mean-based* algorithms which lead to low-regret stationary bounded-recall algorithms? Do low-regret stationary bounded-recall algorithms exist at all?

In this section, we will show that the answer to both of these questions is yes. We begin by constructing a stationary M-bounded-recall algorithm called the AVERAGERESTART algorithm (Algorithm 3) which incurs per-round regret of at most $O(\sqrt{(\log d)/M})$ (matching the lower bound of Theorem 3.1). Intuitively, AVERAGERESTART randomizes over several different versions of PERIODICRESTART – in particular, one can view the output of AVERAGERESTART as first randomly sampling a starting point s uniformly over the last M rounds, and outputting the action that $\mathcal{A}$ would play having only seen rewards r_s through r_{t-1} (if $s \le 0$ or $s > T$, we assume that $r_s = 0$). Note that this algorithm is stationary M-bounded-recall, as no step of this algorithm depends specifically on the round t.

Algorithm 2 AVERAGERESTART

Require: Time horizon T, history window M, no-regret algorithm $\mathcal{A}$.

for $t = 0 \to T$ **do**
  for $m = 1 \to M$ **do**
    $x_t^{(m)} := \mathcal{A}(r_{t-m}, r_{t-m+1}, \ldots, r_{t-1})$ (where we let $r_t = 0$ for $t \le 0$).
  end for
  Play action $x_t = \frac{1}{M}\sum_{m=1}^{M} x_t^{(m)}$.
end for

Theorem 5.1. *Assume the algorithm $\mathcal{A}$ has the property that $\mathsf{Reg}(\mathcal{A}; \mathbf{r}) \le R(T, d)$ for any online learning instance $\mathbf{r} \in [0,1]^{T \times d}$. Then, for any instance $\mathbf{r} \in [0,1]^{T \times d}$*

$$\mathsf{Reg}(\text{AVERAGERESTART}; \mathbf{r}) \le R(M, d)$$

Algorithm 3 RANDOMIZEDAVERAGERESTART

Require: Time horizon T, history window M, no-regret algorithm $\mathcal{A}$.

for $t = 0 \to T$ **do**
  Sample $j \in [M]$ uniformly at random.
  $x_t^{(j)} := \mathcal{A}(r_{t-j}, r_{t-j+1}, \ldots, r_{t-1})$ (where we let $r_t = 0$ for $t \le 0$).
  Play action $x_t := x_t^{(j)}$.
end for

Proof. See Appendix. □

By standard concentration arguments, the corresponding randomized algorithm RANDOMIZEDAVERAGERESTART also has low regret with high probability.

Corollary 5.2. *Fix $\delta > 0$, and define $\mathcal{A}$, R, and $\mathbf{r}$ as in Theorem 5.1. Then, with probability at least $1 - \delta$ (over the randomness due to the algorithm in all rounds),*

$$\mathsf{Reg}(\text{RANDOMIZEDAVERAGERESTART}; \mathbf{r}) \le R(M, d) + \sqrt{T \log(1/\delta)}.$$

As with PERIODICRESTART, by choosing $\mathcal{A}$ to be HEDGE, we obtain a stationary M-bounded-recall algorithm with regret $O(\sqrt{(\log d)/M})$.

Corollary 5.3. *For any $M > 0$, $d \ge 2$, there exists a stationary bounded-recall algorithm $\mathcal{A}$ such that for any online learning instance $\mathbf{r} \in [0,1]^{T \times d}$, $\mathsf{Reg}(\mathcal{A}; \mathbf{r}) \le O(\sqrt{(\log d)/M})$.*

5.2. Averaging restarts over the entire time horizon

Earlier, we asked whether there were non-mean-based learning algorithms which give rise to stationary bounded-recall learning algorithms with regret $o(1)$. While it is not immediately obvious from the description of AVERAGERESTART, this algorithm is indeed of this form. We call the corresponding (non-bounded-recall) learning algorithm AVERAGERESTARTFULLHORIZON (Algorithm 4). In particular, the action x_t output by AVERAGERESTART at time t is given by

$$x_t = \text{AVERAGERESTARTFULLHORIZON}(r_{t-M}, \ldots, r_{t-1}).$$

where AVERAGERESTARTFULLHORIZON is initialized with time horizon M.

Interestingly, if $\mathcal{A}$ is a low-regret algorithm, so is AVERAGERESTARTFULLHORIZON. This follows directly from Theorem 5.1.

Algorithm 4 AVERAGERESTARTFULLHORIZON

Require: Time horizon T, no-regret algorithm $\mathcal{A}$.
for $t = 0 \to T$ **do**
  for $\tau = 1 \to T$ **do**
    $x_t^{(\tau)} := \mathcal{A}(r_{t-\tau}, r_{t-\tau+1}, \dots, r_{t-1})$ (where we let $r_t = 0$ for $t \le 0$).
  end for
  Play action $x_t = \frac{1}{T}\sum_{\tau=1}^{T} x_t^{(\tau)}$.
end for

Theorem 5.4. *Assume the algorithm $\mathcal{A}$ has the property that $\mathsf{Reg}(\mathcal{A}; \mathbf{r}) \le R(T, d)$ for any online learning instance $\mathbf{r} \in [0,1]^{T \times d}$. Then, for any instance $\mathbf{r} \in [0,1]^{T \times d}$*

$$\mathsf{Reg}(\text{AVERAGERESTARTFULLHORIZON}; \mathbf{r}) \le R(T, d).$$

Proof. Set $M = T$ in the proof of Theorem 5.1. (In particular, the proof of Theorem 5.1 applies for any choice of M and T, not only $T \gg M$). □

5.3. The necessity of asymmetry

One basic (and somewhat surprising) property of many mean-based algorithms is that they treat all rounds in the past symmetrically: permuting the order of the past rounds does not affect the action chosen by Hedge or FTRL. Mathematically, this is equivalent to saying that the function $\mathcal{A}(r_1, r_2, \dots, r_{t-1})$ is a symmetric function in its arguments.

Likewise, we can say that a bounded-recall algorithm is *symmetric* if its action x_t at time t is a symmetric function of the rewards $r_{t-M}, \dots, r_{t-1}$ (i.e., the functions f_t in Definition 2.2 are symmetric). Given the prevalence of symmetric no-regret learning algorithms, it is natural to ask whether there exist any symmetric no-regret bounded-recall learning algorithms. Notably, both the periodic restart algorithm and average restart algorithm are not symmetric (they both put significantly more weight on recent rewards).

In this section, we prove that there does not exist a bounded-recall learning algorithm that is all three of stationary, symmetric, and no-regret. We will do this by showing that any such learning algorithm must act "similarly" to a mean-based algorithm. This will allow us to use Lemma 4.2 to show that any symmetric bounded-recall algorithm must have high-regret.

The key lemma we employ is the following.

Lemma 5.5. *Let $\mathcal{A}$ be a stationary, symmetric M-bounded-recall no-regret learning algorithm, and choose a sufficiently large $T \ge 10M/\epsilon^2$ such that $\mathsf{Reg}(\mathcal{A}) \le \epsilon/10$. For any $0 \le n \le M$, let $p(n)$ be the probability that $\mathcal{A}$ plays arm 1 if, of the M previous rewards $r_{t-M}, \dots r_{t-1}$, exactly n of them are equal to $(1,0)$ and exactly $M-n$ of them are equal to $(0,1)$.*

Then, if $n \le (1-\epsilon)(M/2)$, $p(n) \le \epsilon$, and if $n \ge (1+\epsilon)(M/2)$, $p(n) \ge 1-\epsilon$.

Proof sketch. The main idea is to construct a sequence of rewards $\mathbf{r}$ where any segment of M consecutive rewards in $\mathbf{r}$ has the property that exactly n of them equal $(1,0)$, and $M-n$ of them equal $(0,1)$. It is possible to construct such a sequence by taking a segment of M rewards with this property and repeatedly cycling it.

For such a sequence of rewards, $\mathcal{A}$ is forced to select arm 1 with probability exactly $p(n)$ for each round $t > M$. Examining the regret of $\mathcal{A}$ on this sequence leads to the lemma statement; we defer details to the Appendix. □

Lemma 5.5 can be thought of as satisfying the following "weak" form of the mean-based condition: whenever most of the previous rewards are $(1,0)$, $\mathcal{A}$ must put the majority of its weight on arm 1, and whenever most of the previous rewards are $(0,1)$, $\mathcal{A}$ must put the majority of its weight on arm 2. But since the counterexample in Lemma 4.2 primarily uses rewards of the form $(1,0)$ and $(0,1)$, this weak condition is sufficient to prove an analogue of Theorem 4.1.

Theorem 5.6. *Let $\mathcal{A}$ be an M-mean-based symmetric, stationary, learning algorithm. Then, for any sufficiently large T there exists an online learning instance $\mathbf{r} \in [0,1]^{2T}$ with two actions where $\mathsf{Reg}(\mathcal{A}; \mathbf{r}) = \Omega(1)$.*

Proof. We first claim that if we take T large enough so that Lemma 5.5 is satisfied, then $\mathcal{A}$ will incur regret at least $(1 - O(\epsilon)) \cdot (M/18) - o(M)$ on a segment of $3M$ rounds constructed as in Lemma 4.2. To see this, note that the analysis of Lemma 4.2 holds essentially as written, with the exception that we can replace the set $\mathcal{P}$ with $\mathcal{P}_\epsilon = \{t \in [T] | \Delta_t \ge \gamma T + \epsilon(M/2)\}$, with the guarantee that $x_{t,1} \ge 1 - \gamma - \epsilon$ for $t \in \mathcal{P}_\epsilon$. Likewise, we can replace $\mathcal{N}$ with the analogous set $\mathcal{N}_\epsilon$ with the analogous guarantee. Both these changes change the LHS of (2) by at most $O(\epsilon)$, and we therefore arrive at the above regret bound.

(There is one subtle issue with the above argument, which is that for the first M rounds of this segment, some of the rewards in the history window are $(0,0)$. But for these first M rounds, the mean-based algorithm plays optimally, so misplaying here cannot decrease our regret.)

Now, by repeating the same concatenation argument as in the proof of Theorem 4.1, we can show that $\mathcal{A}$ must incur a per-round-regret of at least $1/18 - O(\epsilon) - o(1)$ on $\lfloor T/3M \rfloor$ concatenated instances of the above segment. □

6. Simulations

We conduct some experiments to assess the performance of the bounded-recall learning algorithms we introduce in some simple settings. In particular, we consider Algorithms 1 (PERIODICRESTART), 3 (AVERAGERESTART), 4 (AVERAGERESTARTFULLHORIZON), and an M-bounded-recall mean-based algorithm, all based off of the Multiplicative Weights Update algorithm with fixed learning rate $\eta = 1/2$. We also simulate the classic full-horizon Multiplicative Weights algorithm with the same parameters.

We assess performance in two synthetic environments. In the first environment, we simulate periodic drifting rewards, where the expected reward of arm 1 at time t has mean $|\sin(\pi/6 + t \cdot \pi/\phi)|$, for a period ϕ uniformly chosen from the set $\{T/20, T/10, T/5, T/2\}$. All algorithms we simulate in the first environment have $M = 0.15T$. In the second environment, we simulate the adversarial example from the proof of Lemma 4.2 (with $M = T/3$). In Figure 2, we plot the cumulative regret over time for both environments, averaged over ten independent runs with $T = 1000$.

As we might expect, the bounded-recall algorithms outperform the full-horizon algorithms in the first environment, by virtue of being able to quickly respond to the recent past. In the second environment, we see that, confirming the claims of Theorem 4.1, the choice of bounded-recall algorithm can have a large impact on regret: although PERIODICRESTART and AVERAGERESTART end with significant negative regret, the naive mean-based bounded-recall algorithm ends up with significant positive regret (proportional to T). Interestingly, the bounded-recall no-regret algorithms also outperform the full-horizon no-regret algorithms here (which both end with approximately zero regret).

7. Conclusion and Future Work

In this paper, we initiated the study of bounded-recall online learning and produced novel bounded-recall no-regret algorithms, as well as provided linear regret lower bounds against a natural class of bounded-recall learners.

There are many avenues for future work:

- It would be interesting to consider bounded-recall extensions to other notions of regret (e.g., swap regret (Blum & Mansour, 2007a));
- Our stationary bounded-recall algorithms are more computationally demanding than other no-regret algorithms – understanding the computational limitations of these methods would be useful as well;
- Obtaining a theoretical characterization of performance improvements for bounded-recall methods relative to classic online learners in general non-stationary settings would be quite interesting as well.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Arora, S., Hazan, E., and Kale, S. The multiplicative weights update method: a meta-algorithm and applications. *Theory of computing*, 8(1):121–164, 2012.

Aumann, R. J. and Sorin, S. Cooperation and bounded recall. *Games and Economic Behavior*, 1(1):5–39, 1989.

Blum, A. and Mansour, Y. From external to internal regret. *Journal of Machine Learning Research*, 8(6), 2007a.

Blum, A. and Mansour, Y. Learning, regret minimization, and equilibria. 2007b.

Bousquet, O. and Warmuth, M. K. Tracking a small set of experts by mixing past posteriors. *Journal of Machine Learning Research*, 3(Nov):363–396, 2002.

Braverman, M., Mao, J., Schneider, J., and Weinberg, M. Selling to a no-regret buyer. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pp. 523–538, 2018.

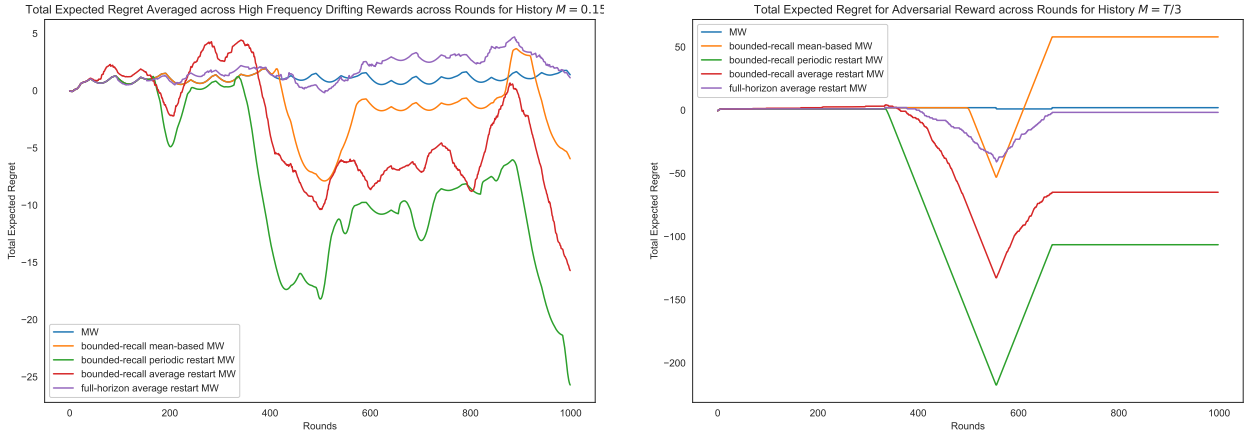
Cesa-Bianchi, N., Mansour, Y., and Stoltz, G. Improved second-order bounds for prediction with expert advice. *Machine Learning*, 66(2):321–352, 2007.

De Rooij, S., Van Erven, T., Grünwald, P. D., and Koolen, W. M. Follow the leader if you can, hedge if you must. *The Journal of Machine Learning Research*, 15(1):1281–1316, 2014.

Drenska, N. and Calder, J. Online prediction with history-dependent experts: The general case. *Communications on Pure and Applied Mathematics*, 76(9):1678–1727, 2023.

Erven, T., Koolen, W. M., Rooij, S., and Grünwald, P. Adaptive hedge. *Advances in Neural Information Processing Systems*, 24, 2011.

Ginart, A., Guan, M., Valiant, G., and Zou, J. Y. Making ai forget you: Data deletion in machine learning. *Advances in Neural Information Processing Systems*, 32, 2019.



(a) Left

(b) Right

Figure 2. (Left) We plot the total regret of the algorithms over time over a uniform average of high-frequency drifting scenarios where the periods of the mean reward of arm 1 are $T/20, T/10, T/5$, and $T/2$ and arm 2 flips an unbiased coin for reward $\{\pm 1\}$ – the bounded-recall algorithms significantly outperform the classic no-regret algorithms. (Right) We plot the total regret of the algorithms over time for one block of the adversarial rewards case (see the construction in Lemma 4.2) – observe that the mean-based bounded-recall learner attains regret on order $M/6$ (here, $M = T/3$), while our no-regret bounded-recall learners all outperform Multiplicative Weights.

Hazan, E. and Kale, S. Extracting certainty from uncertainty: Regret bounded by variation in costs. *Machine learning*, 80(2):165–188, 2010.

Huyen, C. Data distribution shifts and monitoring, Feb 2022. URL https://huyenchip.com/.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

Koolen, W. M., Grünwald, P., and Van Erven, T. Combining adversarial guarantees and stochastic fast rates in online learning. *Advances in Neural Information Processing Systems*, 29, 2016.

Lécuyer, M., Spahn, R., Vodrahalli, K., Geambasu, R., and Hsu, D. Privacy accounting and quality control in the sage differentially private ml platform. In *Proceedings of the 27th ACM Symposium on Operating Systems Principles*, pp. 181–195, 2019.

Lehrer, E. Repeated games with stationary bounded recall strategies. *Journal of Economic Theory*, 46(1):130–144, 1988.

Mourtada, J. and Gaïffas, S. On the optimality of the hedge algorithm in the stochastic regime. *Journal of Machine Learning Research*, 20:1–28, 2019.

Nekipelov, D., Syrgkanis, V., and Tardos, E. Econometrics for learning agents. In *Proceedings of the sixteenth acm conference on economics and computation*, pp. 1–18, 2015.

Neyman, A. *Cooperation, repetition, and automata*. Springer, 1997.

Qiao, M. and Valiant, G. Exponential weights algorithms for selective learning. In *Conference on Learning Theory*, pp. 3833–3858. PMLR, 2021.

Rabanser, S., Günnemann, S., and Lipton, Z. Failing loudly: An empirical study of methods for detecting dataset shift. *Advances in Neural Information Processing Systems*, 32, 2019.

Roughgarden, T. Online learning and the multiplicative weights algorithm, February 2016.

Sugiyama, M. and Kawanabe, M. *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press, 2012.

Team, G., Anil, R., Borgeaud, S., Wu, Y., Alayrac, J.-B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A. M., Hauth, A., et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Voigt, P. and Von dem Bussche, A. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676): 10–5555, 2017.

Wiles, O., Gowal, S., Stimberg, F., Alvise-Rebuffi, S., Ktena, I., Cemgil, T., et al. A fine-grained analysis on distribution shift. *arXiv preprint arXiv:2110.11328*, 2021.

Wu, Y. *Learning to Predict and Make Decisions under Distribution Shift*. PhD thesis, Carnegie Mellon University, 2021.

Zapechelnyuk, A. Better-reply dynamics with bounded recall. *Mathematics of Operations Research*, 33(4):869–879, 2008.

Zhang, L., Yang, T., Zhou, Z.-H., et al. Dynamic regret of strongly adaptive methods. In *International conference on machine learning*, pp. 5882–5891. PMLR, 2018.

Zheng, K., Luo, H., Diakonikolas, I., and Wang, L. Equipping experts/bandits with long-term memory. *Advances in Neural Information Processing Systems*, 32, 2019.

A. Related Work

A.1. Adaptive Multiplicative Weights

A lot of existing work focuses on understanding the behavior of Multiplicative Weights under various non-adversarial assumptions and properties of the loss sequence (Cesa-Bianchi et al., 2007; Hazan & Kale, 2010), as well as on coming up with adaptive algorithms which perform better than Multiplicative Weights in both worst-case and average-case settings (Erven et al., 2011; De Rooij et al., 2014; Koolen et al., 2016; Mourtada & Gaïffas, 2019). There is a particular emphasis on trading off between favorable performance for Follow-the-Leader and Multiplicative Weights in various average case settings. Our work connects to this literature by introducing bounded-recall algorithms which empirically outperform both MW and FTL for a class of drifting and periodic rewards.

A.2. Private Learning and Discarding Data

Bounded-recall online learning algorithms can be viewed as private with respect to data sufficiently far in the past – such data is not taken into account in the prediction. This approach to achieving privacy is similar in principle to *federated learning* (Kairouz et al., 2021), which ensures that by decentralizing the data store, many parties participating in the model training will simply never come into contact with certain raw data points, thus mitigating privacy risks. Similarly, bounded-recall approaches also provide an alternate angle on privacy-preserving ML systems which must store increasing amounts of streaming data (and thereby must adaptively set their privacy costs to avoid running out of privacy budget, see Lécuyer et al. (2019)). Additionally, the bounded-recall approach to privacy yields algorithms for which data deletion (Ginart et al., 2019) is efficient – thereby ensuring "the right to be forgotten."

A.3. Shifting Data Distributions

Bounded-recall algorithms are a natural approach to online learning over non-stationary time series, which is a common problem in practical industry settings (Huyen, 2022) and which has been studied for many years (Sugiyama & Kawanabe, 2012; Wiles et al., 2021; Wu, 2021; Rabanser et al., 2019). In particular, one can view bounded-recall online learners as adapting to non-stationary structure in the data – thus, one may not need to go through the whole process of detecting a distribution shift and then deciding to re-train – ideally the learning algorithm is adaptive and automatically takes such eventualities into account. Our proposed bounded-recall methods are one step in this direction.

B. Omitted Proofs

B.1. Proof of Theorem 3.1

Proof of Theorem 3.1. Our hard example is simple: we make use of standard hard examples used in regret lower bounds for the classical online learning setting (see e.g. Roughgarden (2016)), and simply append to such an example of length M a block of M 0 rewards for all actions (effectively resetting the internal state of any bounded-recall algorithm, and resulting in 0 regret during that block). This trick allows us to only consider the regret on blocks of size M, and since the blocks are repeated, each block has the same best action in hindsight. Then taking the full block of length $2M$ together, we get an average regret lower bound for each block of size $\frac{1}{2}\Omega(\sqrt{M \log d})$. Adding up the regret lower bounds, we get a total regret lower bound for any bounded-recall online learning algorithm with past window of size M to be $\Omega\left(\frac{T}{2M} \cdot \sqrt{M \log d}\right) = \Omega\left(\sqrt{\frac{T^2 \log d}{M}}\right)$, or $\Omega\left(\sqrt{\frac{\log d}{M}}\right)$ on average, as desired. □

B.2. Proof of Theorem 5.1

Proof of Theorem 5.1. We will proceed by first proving the statement for the deterministic algorithm, the proof for the randomized algorithm follows directly. Intuitively, we will decompose the output of AVERAGERESTART as a uniform combination of M copies of PERIODICRESTART (one for each offset modulo M of reset location); since PERIODICRESTART has per-round regret $R(M, d)$, so will AVERAGERESTART.

Let $i^* = \arg\max_{i \in [d]} \sum_t r_{t,i}$. For any $t \in [-(M-1), T-1]$ and $m \in [M]$, let $y_t^{(m)} = x_{t+m}^{(m)} = \mathcal{A}(r_t, r_{t+1}, \ldots, r_{t+m-1})$. Now, note that

$$\begin{aligned}
\sum_{t=1}^{T}\langle x_t, r_t\rangle &= \sum_{t=1}^{T}\frac{1}{M}\sum_{m=1}^{M}\langle x_t^{(m)}, r_t\rangle \\
&= \sum_{t=1}^{T}\frac{1}{M}\sum_{m=1}^{M}\langle y_{t-m}^{(m)}, r_t\rangle \\
&= \sum_{t'=-M+1}^{T-1}\frac{1}{M}\sum_{m=1}^{M}\langle y_{t'}^{(m)}, r_{t'+m}\rangle \\
&= \sum_{t'=-M+1}^{T-1}\frac{1}{M}\sum_{m=1}^{M}\langle \mathcal{A}(r_{t'}, r_{t'+1}, \ldots, r_{t'+m-1}), r_{t'+m}\rangle \\
&\geq \sum_{t'=-M+1}^{T-1}\frac{1}{M}\left(\sum_{m=1}^{M} r_{t'+m,i^*} - M\cdot R(M,d)\right) \\
&= \left(\sum_{t=1}^{T} r_{t,i^*}\right) - (T+M)R(M,d).
\end{aligned}$$

Here we have twice used the fact that $r_s = 0$ for $s \leq 0$ and $s > T$; once for rewriting the sum in t in terms of t' (the terms that do not appear in the original sum have $t' + m \notin [1, T]$, so the inner product evaluates to 0), and once again when we rewrite the sum in t' in terms of T (each $r_{t,i}$ for $1 \leq t \leq T$ appears exactly M times; other $r_{t,i}$ for $t \notin [1, T]$ appear a variable number of times, but they all equal 0). Since $M \leq T$, it follows that $\mathsf{Reg}(\textsc{AverageRestart}; \mathbf{r}) \leq R(M, d)$.

□

B.3. Proof of Corollary 5.2

Proof of Corollary 5.2. To extend this proof to the randomized variant of the algorithm, note that the expected reward of the randomized algorithm in round t is equal to the reward of the deterministic algorithm via linearity of expectation:

$$\mathop{\mathbb{E}}_{j\sim\text{Unif}([M])}\left[\sum_{t=1}^{T}\langle x_t^{(j)}, r_t\rangle\right] = \sum_{t=1}^{T}\mathop{\mathbb{E}}_{j\sim\text{Unif}([M])}\left[\langle x_t^{(j)}, r_t\rangle\right] = \sum_{t=1}^{T}\frac{1}{M}\sum_{m=1}^{M}\langle x_t^{(m)}, r_t\rangle.$$

Now, let $X_t = \langle x_t^{(j)}, r_t\rangle$ be the r.v. corresponding to the randomized algorithm's reward in round t, and $\overline{X}_t = \mathbb{E}[X_t] = \frac{1}{M}\sum_{m=1}^{M}\langle x_t^{(m)}, r_t\rangle$ the reward of the deterministic algorithm in round t. Since $x_t \in \Delta_d$ and $r_t \in [0,1]^d$, X_t lies in $[0, 1]$, so by Hoeffding's inequality

$$\Pr\left[\sum X_t - \sum \overline{X}_t \geq -\sqrt{T\log(1/\delta)}\right] \geq 1 - \delta,$$

as desired. □

B.4. Proof of Lemma 5.5

Proof of Lemma 5.5. We will assume that $n \leq (1-\epsilon)(M/2)$ (the other case can be handled symmetrically). Fix any sequence $\mathbf{r}_M$ of M rewards, n of which are $(1, 0)$ and $M - n$ of which are $(0, 1)$. Form a sequence $\mathbf{r}$ of T rewards by repeating $\mathbf{r}_M$ multiple times.

Note that $\mathbf{r}$ has the property that any segment of M consecutive rewards contains exactly n $(1, 0)$s and $M - n$ $(0, 1)$s. So in each round after round M, the algorithm $\mathcal{A}$ will play arm 1 with probability $p(n)$. By doing this, they receive reward at most:

$$\text{Reward}(\mathcal{A}) \leq M + \left(p(n)\frac{n}{M} + (1-p(n))\cdot\left(1 - \frac{n}{M}\right)\right)(T - M),$$

(since they receive at most reward 1 each round for the first M rounds). Since $n < M/2$, the best fixed arm in hindsight is arm 2, so the optimal static adversary receives reward at least

$$\mathrm{Opt}(\mathcal{A}) \geq \left(1 - \frac{n}{M}\right)(T - M).$$

It follows that the regret of $\mathcal{A}$ on this sequence is at least

$$\begin{aligned}
\mathsf{Reg}(\mathcal{A}; \mathbf{r}) &= \frac{1}{T}(\mathrm{Opt}(\mathcal{A}) - \mathrm{Reward}(\mathcal{A})) \\
&\geq \frac{1}{T}\left[(T - M)\left(1 - \frac{2n}{M}\right)p(n) - M\right] \\
&\geq (1 - 0.1\epsilon^2)(\epsilon)p(n) - 0.1\epsilon \\
&\geq 0.5\epsilon p(n) - 0.1\epsilon^2
\end{aligned}$$

Since $\mathsf{Reg}(\mathcal{A}) \leq 0.1\epsilon^2$, this implies that $p(n) \leq (0.2\epsilon^2)/(0.5\epsilon) \leq \epsilon$, as desired. □