A Closer Look at Adaptive Regret

Dmitry Adamskiy[1], Wouter M. Koolen[1], Alexey Chernov[2], and Vladimir Vovk[1]

[1] Computer Learning Research Centre and Department of Computer Science, Royal Holloway, University of London, Egham, Surrey, TW20 0EX, UK
[2] Department Mathematical Sciences, Durham University, Durham, DH1 3LE, UK

Abstract. For the prediction with expert advice setting, we consider methods to construct algorithms that have low adaptive regret. The adaptive regret of an algorithm on a time interval $[t_1, t_2]$ is the loss of the algorithm there minus the loss of the best expert. Adaptive regret measures how well the algorithm approximates the best expert locally, and it is therefore somewhere between the classical regret (measured on all outcomes) and the tracking regret, where the algorithm is compared to a good sequence of experts.
We investigate two existing intuitive methods to derive algorithms with low adaptive regret, one based on specialist experts and the other based on restarts. Quite surprisingly, we show that both methods lead to the same algorithm, namely Fixed Share, which is known for its tracking regret. Our main result is a thorough analysis of the adaptive regret of Fixed Share. We obtain the exact worst-case adaptive regret for Fixed Share, from which the classical tracking bounds can be derived. We also prove that Fixed Share is optimal, in the sense that no algorithm can have a better adaptive regret bound.

Keywords: Online learning, adaptive regret, Fixed Share, specialist experts

1 Introduction

This paper deals with the prediction with expert advice setting. Nature generates outcomes step by step. At every step Learner tries to predict the outcome. Then the actual outcome is revealed and the quality of Learner's prediction is measured by a loss function.

No assumptions are made about the nature of the data. Instead, at every step Learner is presented with the predictions of a pool of experts and he may base his predictions on these. The goal of Learner in the classical setting is to guarantee small regret, that is, to suffer cumulative loss that is not much larger than that of the best (in hindsight) expert from the pool. Several classical algorithms exist for this task, including the Aggregating Algorithm [13] and the Exponentially Weighted Forecaster [3]. In the standard log-loss game the regret incurred by those algorithms when competing with N experts is at most $\ln N$.

A common extension of the framework takes into account the fact that the best expert could change with time. In this case we may be interested in competing with the best *sequence* of experts from the pool. Known algorithms for this task include Fixed Share [8] and Mixing Past Posteriors [1].

In this paper we focus on the related task of obtaining small *adaptive* regret, a notion first considered in [11] and later studied in [7]. The adaptive regret of an algorithm on a time interval $[t_1, t_2]$ is the loss of the algorithm there, minus the loss of the best expert for that interval:

$$R_{[t_1,t_2]} \coloneqq L_{[t_1,t_2]} - \min_j L^j_{[t_1,t_2]}$$

The goal is now to ensure small regret on all intervals simultaneously. Note that adaptive regret was defined in [7] with a maximum over intervals, but we need the fine-grained dependence on the endpoint times to be able to prove matching upper and lower bounds.

Our results. The contribution of our paper is twofold.

1. We study two constructions to get adaptive regret algorithms from existing classical regret algorithms. The first one is a simple construction which originates in [5] and [4] and involves so called sleeping (specialist) experts, and the second one uses restarts, as proposed in [7]. Although conceptually dissimilar, we show that both constructions reduce to the Fixed Share algorithm with variable switching rate.
2. We compute the exact adaptive worst-case regret of Fixed Share and show that no algorithm can have better adaptive regret. We also derive the tracking regret bounds from the adaptive regret bounds, showing that the latter are in fact more fundamental.

Here is a sneak preview of the adaptive bounds we obtain, presented in a slightly relaxed form for simplicity. The refined statement can be found in Theorem 4 below. In the log-loss game for each of the following adaptive regret bounds there is an algorithm achieving it, simultaneously for all the intervals $[t_1, t_2]$:

$$\ln N + \ln t_2 \,, \tag{1a}$$

$$\ln N + \ln t_1 + \ln\ln t_2 + 2 \,, \tag{1b}$$

$$\ln N + 2\ln t_1 + 1 \,, \tag{1c}$$

where $\ln\ln 1$ is interpreted as 0.

Outline. The structure of the paper is as follows. In Section 2 we give the description of the protocol and review the standard algorithms. In Section 3 we study two intuitive ways of obtaining adaptive regret algorithms from classical algorithms. We show that curiously both these algorithms turn out to be Fixed Share. In Section 4 we study in detail the adaptive regret of Fixed Share.

2 Setup

We phrase our results in the setting defined in Protocol 1 which, for lack of a standard name, we call *mix loss*. We choose this fundamental setting because

Protocol 1 Mix loss prediction

for $t = 1, 2, \dots$ **do**

Learner announces probability vector $\boldsymbol{u}_t \in \triangle_N$

Reality announces loss vector $\boldsymbol{\ell}_t \in [-\infty, \infty]^N$

Learner suffers loss $\ell_t := -\ln \sum_n u_t^n e^{-\ell_t^n}$

end for

it is universal, in the sense that many other common settings reduce to it. For example probability forecasting, sequential investment and data compression are straightforward instances [3]. In addition, mix loss is the baseline for the wider class of *mixable loss functions*, which includes e. g. square loss [14]. Classical regret bounds transfer from mix loss to mixable losses almost by definition, and the same reasoning extends to adaptive regret bounds. In addition, mix loss results carry over in the usual modular ways (via Hoeffding and related bounds) to non-mixable games, which include the Hedge setting [6] and Online Convex Optimisation [16].

Let us introduce two standard algorithms in this setup. The *Aggregating Algorithm* [15] is parametrised by a prior distribution $\boldsymbol{u}_1$ on $[N]$ (where $[N]$ denotes the set $\{1, \dots, N\}$). It predicts in trial t with

$$u_t^n \;:=\; \frac{u_1^n e^{-\sum_{s<t} \ell_s^n}}{\sum_n u_1^n e^{-\sum_{s<t} \ell_s^n}}, \tag{2a}$$

which we may also maintain incrementally using the update rule

$$u_{t+1}^n \;=\; \frac{u_t^n e^{-\ell_t^n}}{\sum_n u_t^n e^{-\ell_t^n}}. \tag{2b}$$

For this algorithm with uniform prior $u_1^n = 1/N$, the classical regret bound states that for each expert j

$$\sum_{t=1}^T \ell_t - \sum_{t=1}^T \ell_t^j \;\leq\; \ln N.$$

Note that AA is minimax for classical mix loss regret since $\geq \ln N$ can be inflicted on any algorithm. The second algorithm, *Fixed Share* [8], in addition to a prior $\boldsymbol{u}_1$ requires a sequence of switching rates $\alpha_2, \alpha_3, \dots$ Intuitively, α_t is the probability of a switch in the sequence of "best-at-the-step" experts *before* trial t. The weights are now updated as

$$u_{t+1}^n \;:=\; \frac{\alpha_{t+1}}{N-1} + \left(1 - \frac{N}{N-1}\alpha_{t+1}\right) \frac{u_t^n e^{-\ell_t^n}}{\sum_n u_t^n e^{-\ell_t^n}}\,. \tag{3}$$

(We see that the Aggregating Algorithm is the special case when all α_t are 0.) The tracking regret bound for Fixed Share with uniform prior $\boldsymbol{u}_1$ and constant

$\alpha_t = \alpha$ switching rate states that for any reference sequence $j_1, \ldots, j_T$ of experts with m blocks (and hence $m-1$ switches)

$$\sum_{t=1}^T \ell_t - \sum_{t=1}^T \ell_t^{j_t} \;\leq\; \ln N + (m-1)\ln(N-1) - (m-1)\ln\alpha - (T-m)\ln(1-\alpha).$$

Having introduced the standard classical and tracking regret algorithms, we now turn to adaptive regret.

3 Intuitive algorithms with low adaptive regret

Two methods have been proposed in the literature that can be used to obtain adaptive regret bounds: specialist experts and restarts. We discuss both and show that each of them yields Fixed Share with a particular choice of time-dependent switching rate α_t.

3.1 Specialist experts

One way of getting an adaptive algorithm is the following. We create a pool of virtual experts. For each real expert n and time t, we include a virtual expert that mimics Learner's behaviour for the first $t-1$ trials (which is another way to say that this expert is a specialist [5] that abstains from prediction, or *sleeps*, during the first $t-1$ trials), and predicts as expert n from trial t onward. Then the classical regret w.r.t. this virtual expert on $[1, T]$ is the same as the adaptive regret w.r.t. the real expert n on $[t, T]$ because on the first $t-1$ steps the loss of the virtual expert equals Learner's loss. The natural idea is to feed all those virtual experts into the existing algorithm capable of obtaining good classical regret, the AA. For fixed t_2, the uniform prior on wake-up time $t_1 \leq t_2$ and expert n would lead to adaptive regret $\ln(Nt_2)$. It turns out that the same holds even without knowledge of t_2.

There is a snag, namely that in the prediction step you need to know the losses of the sleeping virtual specialists which are equal to the yet unknown loss of the Learner. However, it is possible to find a fixed point prediction which makes the AA loss exactly the same as if it took into account the sleeping experts making the same prediction. To avoid dealing with equations involving an infinite number of sleeping experts let us fix a time horizon $T > t$. Later we will see that this time horizon plays no role.

Let us denote by $w_t^{n,s}$ the probability assigned by the AA in trial t to the virtual specialist parametrised by real expert n and wake-up time s. Learner then will predict with weights $\boldsymbol{u}_t$ where $u_t^n = \sum_{s=1}^t w_t^{n,s} \Big/ \sum_{j=1}^N \sum_{\tau=1}^t w_t^{j,\tau}$. The desired fixed point property is achieved for this prediction:

$$\ell_t \;:=\; -\ln\left(\sum_{n=1}^N u_t^n e^{-\ell_t^n}\right) \;=\; -\ln\left(\sum_{n=1}^N \sum_{s=1}^t w_t^{n,s} e^{-\ell_t^n} + \sum_{n=1}^N \sum_{s=t+1}^T w_t^{n,s} e^{-\ell_t}\right).$$

That is, the loss ℓ_t of the prediction $\boldsymbol{u}_t$ in the game with N real experts equals the loss of the prediction $\boldsymbol{w}_t$ in the game with TN virtual specialists, where specialists that are still asleep are assumed to suffer Learner's loss ℓ_t.

At first glance, it is very inefficient to maintain weights of TN specialists. However, we do not need to, since we may merge the weights of all awake specialists associated to the same real expert, resulting in Algorithm 1. To verify this, denote this merged (unnormalised) weight in trial t by v_t^n for each real expert n. The merged (unnormalised) weight v_{t+1}^n of this real expert n in the next trial $t+1$ consists of the prior weight of the newly awaken virtual specialist plus v_t^n, the sum of the old weights, each multiplied by the same factor $e^{(\ell^t - \ell_t^n)}$ (as they were all awake). Thus we can update the sum directly, and this is reflected by our update rule.

Note that for simplicity, we have taken the prior on experts and wake-up times independent, i. e.

$$p^{(n,t)} \;=\; p(t)\,.$$

Also note that there is no need for the priors $p^{(n,t)}$ to normalise.

Algorithm 1 Adaptive Aggregating Algorithm

Input: Prior nonnegative weights $p(t)$, $t = 1, 2, \ldots$

$v_1^n := p(1), n = 1, \ldots, N$

for $t = 1, 2, \ldots$ **do**

Play weights $u_t^n := \frac{v_t^n}{\sum_{k=1}^N v_t^k}$

Read the experts losses ℓ_t^n, $n = 1, \ldots, N$

Set $v_{t+1}^n := p(t+1) + v_t^n \frac{e^{-\ell_t^n}}{\sum_{k=1}^N u_t^k e^{-\ell_t^k}}$, $n = 1, \ldots, N$

end for

Now we will see that Algorithm 1 turns out to be Fixed Share with variable switching rate. In the rest of this section we derive this. Let $P(t) = \sum_{s=1}^t p(s)$.

Fact 1. *The update step of Algorithm 1 preserves the following:*

$$\sum_n v_t^n = \sum_n \sum_{s \le t} p(s) = NP(t)\,.$$

Proof. This follows immediately from expanding the one-step update rule:

$$\begin{aligned}
\sum_n v_{t+1}^n \;&=\; \sum_n p(t+1) + \sum_n v_t^n \frac{e^{-\ell_t^n}}{\sum_k u_t^k e^{-\ell_t^k}} \\
&=\; \sum_n p(t+1) + \sum_n v_t^n \frac{e^{-\ell_t^n}}{\sum_k \frac{v_t^k}{\sum_j v_t^j} e^{-\ell_t^k}} \\
&=\; Np(t+1) + \sum_n v_t^n \overset{\text{Induction}}{=} NP(t+1)\,.
\end{aligned}$$

□

We now show that Algorithm 1 can be seen as Fixed Share (and vice versa).

Lemma 2. *Say that α_t is the probability of a Fixed Share switch before trial t, and $p(t)$ is the prior weight of specialist waking up in trial t in Algorithm 1. Then the following conversion preserves behaviour*

$$p(t) \;=\; \frac{\frac{N}{N-1}\alpha_t}{\prod_{s=2}^{t}(1-\frac{N}{N-1}\alpha_s)}, \qquad \alpha_t \;=\; \frac{N-1}{N}\frac{p(t)}{\sum_{s=1}^{t}p(s)},$$

where we use the convention that $\alpha_1 = \frac{N-1}{N}$.

Proof. Let us rewrite the update step of Algorithm 1 for the normalised weights.

$$\begin{aligned}
u_{t+1}^n &= \frac{v_{t+1}^n}{\sum_k v_{t+1}^k} = \frac{p(t+1)}{NP(t+1)} + \frac{1}{NP(t+1)}v_t^n\frac{e^{-\ell_t^n}}{\sum_k u_t^k e^{-\ell_t^k}} \\
&= \frac{p(t+1)}{NP(t+1)} + \frac{1}{NP(t+1)}NP(t)u_t^n\frac{e^{-\ell_t^n}}{\sum_k u_t^k e^{-\ell_t^k}} \\
&= \frac{\alpha_{t+1}}{N-1} + \frac{P(t+1)-p(t+1)}{P(t+1)}u_t^n\frac{e^{-\ell_t^n}}{\sum_k u_t^k e^{-\ell_t^k}} \\
&= \frac{\alpha_{t+1}}{N-1} + \left(1-\frac{N}{N-1}\alpha_{t+1}\right)u_t^n\frac{e^{-\ell_t^n}}{\sum_k u_t^k e^{-\ell_t^k}}.
\end{aligned}$$

We see that the weight update is the update of the Fixed Share algorithm with variable switching rate α_t. □

The idea to use specialist experts for obtaining adaptive bounds was introduced in [5]. There a virtual specialist is created for every interval $[t_1, t_2]$ which leads to redundancy and suboptimal bounds. Their adaptive regret bounds sport a term which exceeds $2\ln t_2$ whereas our bounds (1) have at most a single $\ln t_2$.

3.2 Restarts

A second intuitive method to obtain adaptive regret bounds, called Follow the Leading History (FLH), was introduced in [7]. One starts with a base algorithm that ensures low classical regret. FLH then obtains low adaptive regret by restarting a copy of this base algorithm each trial, and aggregating the predictions of these copies. To get low adaptive regret w.r.t. N experts, it is natural to take the AA as the base algorithm. We now show that FLH with this choice equals Fixed Share with switching rate $\alpha_t = \frac{N-1}{Nt}$.

For each n, s and $t \geq s$, let $p_t^{n|s}$ denote the weight allocated to expert n by the copy of the AA started at time s. By definition $p_s^{n|s} = 1/N$, and these weights evolve according to (2b). We denote by p_t^s the weight allocated by FLH in trial $t \geq s$ to the copy of AA started at time s. In [7], these weights are defined as follows. Initially $p_1^1 = 1$ and subsequently

$$p_{t+1}^s \;=\; \left(1-\frac{1}{t+1}\right)\frac{p_t^s e^{-\left(-\ln\sum_n p_t^{n|s}e^{-\ell_t^n}\right)}}{\sum_{r=1}^{t} p_t^r e^{-\left(-\ln\sum_n p_t^{n|r}e^{-\ell_t^n}\right)}}, \qquad p_{t+1}^{t+1} \;=\; \frac{1}{t+1}.$$

Lemma 3. *For mix loss, FLH with AA as the base algorithm issues the same predictions as Fixed Share with learning rate* $\alpha_t = \frac{N-1}{Nt}$.

Proof. We prove by induction on t that the FS and FLH weights coincide:

$$u_t^n \; = \; \sum_{s=1}^{t} p_t^{n|s} p_t^s.$$

The base case $t = 1$ is obvious. For the induction step we expand

$$\begin{aligned}
\sum_{s=1}^{t+1} p_{t+1}^{n|s} p_{t+1}^s \; &= \; \sum_{s=1}^{t} p_{t+1}^{n|s} p_{t+1}^s + p_{t+1}^{t+1}/N \\
= \; \left(1 - \frac{1}{t+1}\right) \sum_{s=1}^{t} &\left(\frac{p_t^{n|s} e^{-\ell_t^n}}{\sum_n p_t^{n|s} e^{-\ell_t^n}} \frac{p_t^s \left(\sum_n p_t^{n|s} e^{-\ell_t^n}\right)}{\sum_{r=1}^t p_t^r \left(\sum_n p_t^{n|r} e^{-\ell_t^n}\right)} \right) + \frac{1}{N(t+1)} \\
&= \; \left(1 - \frac{1}{t+1}\right) \frac{\sum_{s=1}^t p_t^s p_t^{n|s} e^{-\ell_t^n}}{\sum_{r=1}^t \sum_n p_t^r p_t^{n|r} e^{-\ell_t^n}} + \frac{1}{N(t+1)} \\
&\overset{\text{Induction}}{=} \; \left(1 - \frac{1}{t+1}\right) \frac{u_t^n e^{-\ell_t^n}}{\sum_n u_t^n e^{-\ell_t^n}} + \frac{1}{N(t+1)} \; = \; u_{t+1}^n,
\end{aligned}$$

and find the Fixed Share update equation (3) for switching rate $\alpha_t = \frac{N-1}{Nt}$. □

4 The adaptive regret of Fixed Share

We have seen in the previous section that both intuitive approaches to obtain algorithms with low adaptive regret result in Fixed Share. We take this convergence to mean that Fixed Share is the most fundamental adaptive algorithm. The tracking regret for Fixed Share is already well-studied. In this section we thoroughly analyse the adaptive regret of Fixed Share. We obtain the worst-case adaptive regret for mix loss. This result implies the known tracking regret bounds.

We also show an information-theoretic lower bound for mix loss that must hold for any algorithm, and which is tight for Fixed Share. This proves that Fixed Share is a Pareto-optimal algorithm for the mix loss game, in the sense that no other algorithm can guarantee essentially better adaptive regret.

4.1 The exact worst-case adaptive regret for mix loss

In this section we first compute the exact worst-case adaptive regret of Fixed Share with arbitrary switching rate α_t. Then we obtain certain regret bounds of interest, including the tracking regret bound, for particular choices of α_t.

Theorem 4. *The worst-case adaptive regret of Fixed Share with N experts on interval $[t_1, t_2]$ equals*

$$-\ln\left(\frac{\alpha_{t_1}}{N-1}\prod_{t=t_1+1}^{t_2}(1-\alpha_t)\right). \tag{4}$$

Proof. The proof consists of two parts. First we claim that the worst-case data for the interval $[t_1, t_2]$ in the setting of Protocol 1 is rather simple: on the interval there is one *good* expert (all others get infinite losses) and on the single trial before the interval (if $t_1 > 1$) this expert suffers infinite loss while others do not. The proof of this can be found in Appendix A.

Now we will compute the regret on this data. The regret of Fixed Share on the interval $[t_1, t_2]$ is $-\ln$ of the product of the weights put on the good expert (say, j) on this interval:

$$R^{\mathrm{FS}}_{[t_1,t_2]} = -\ln \prod_{t_1 \le t \le t_2} u_t^j .$$

It is straightforward to derive $u_{t_1}^j$ from (3):

$$u_{t_1}^j = \frac{\alpha_{t_1}}{N-1} \qquad \text{and} \qquad u_t^j = 1-\alpha_t \qquad \text{for } t \in [t_1+1, t_2]$$

from which the statement follows. □

Example 1: constant switching rate. This is the original Fixed Share [8].

Corollary 5. *Fixed Share with constant switching rate $\alpha_t = \alpha$ for $t > 1$ (recall that $\alpha_1 = \frac{N-1}{N}$) has worst-case adaptive regret equal to*

$$\begin{aligned} &\ln(N-1) - \ln\alpha - (t_2 - t_1)\ln\big(1-\alpha\big) && \textit{for } t_1 > 1, \textit{ and} \\ &\ln N - (t_2 - 1)\ln\big(1-\alpha\big) && \textit{for } t_1 = 1. \end{aligned}$$

A slightly weaker upper bound was obtained in [2]. The clear advantage of our analysis with equality is that we can obtain the standard Fixed Share tracking regret bound by summing the above adaptive regret bounds on individual intervals. Comparing with the best sequence of experts S on the interval $[1, T]$ with m blocks, we obtain the bound

$$L^{\mathrm{FS}}_{[1,T]} - L^{S}_{[1,T]} \le \ln N + (m-1)\ln(N-1) - (m-1)\ln\alpha - (T-m)\ln(1-\alpha),$$

which is exactly the Fixed Share standard bound. So we see that the reason why Fixed Share can effectively compete with switching sequences is that it can, in fact, effectively compete with an expert on any interval, that is, has small adaptive regret.

Example 2: slowly decreasing switching rate. The idea of slowly decreasing the switching rate was considered in [12] in the context of source coding, and later analysed for expert switching in [10]; we saw in Section 3.2 that it also underlies Follow the Leading History of [7]. It results in tracking regret bounds that are almost as good as the bounds for constant α *with optimally tuned* α. These tracking bounds are again implied by the following corresponding adaptive regret bound.

Corollary 6. *Fixed Share with switching rate* $\alpha_t = 1/t$ *(except for* $\alpha_1 = \frac{N-1}{N}$*) has worst-case adaptive regret*

$$-\ln\left(\frac{1}{(N-1)t_1}\prod_{t=t_1+1}^{t_2}\frac{t-1}{t}\right) = \ln(N-1) + \ln t_2 \qquad \text{for } t_1 > 1, \text{ and} \tag{5a}$$

$$-\ln\left(\frac{1}{N}\prod_{t=2}^{t_2}\frac{t-1}{t}\right) = \ln N + \ln t_2 \qquad \text{for } t_1 = 1. \tag{5b}$$

Example 3: quickly decreasing switching rate. The bounds we have obtained so far depend on t_2 either linearly or logarithmically. To get bounds that depend on t_2 sub-logarithmically, or even not at all, one may instead decrease the switching rate faster than $1/t$, as analysed in [12,9]. To obtain a controlled trade-off, we consider setting the switching rate to $\alpha_t = \frac{1}{t\ln t}$, except for $\alpha_1 = \frac{N-1}{N}$. This leads to adaptive regret at most

$$\ln(N-1) + \ln t_1 + \ln\ln t_1 - \sum_{t=t_1+1}^{t_2}\ln\left(1 - \frac{1}{t\ln t}\right) \leq \ln(N-1) + \ln t_1 + \ln\ln t_2 + 1.28 \tag{6a}$$

when $t_1 > 1$ and

$$\ln N - \sum_{t=2}^{t_2}\ln\left(1 - \frac{1}{t\ln t}\right) \leq \ln N + \ln\ln t_2 + 1.65 \tag{6b}$$

when $t_1 = 1$ (remember that $\ln\ln 1$ is understood to be 0). The constant 1.28 in (6a) is needed because t_1 and t_2 can take small values; e.g., if we only consider $t_1 \geq 10$, we can replace 1.28 by 0.05, and we can replace 1.28 by an arbitrarily small $\delta > 0$ if we only consider $t_1 \geq c$ for a sufficiently large c.

The dependence on t_2 in (6) is extremely mild. We can suppress it completely by increasing the dependence on t_1 just ever so slightly. If we set $\alpha_t = t^{-1-\epsilon}$, where $\epsilon > 0$, then the sum of the series $\sum_{t=1}^{\infty}\alpha_t$ is finite and the bound becomes

$$\ln(N-1) + (1+\epsilon)\ln t_1 + c_\epsilon \qquad \text{for } t_1 > 1, \text{ and} \tag{7a}$$

$$\ln N + c_\epsilon \qquad \text{for } t_1 = 1, \tag{7b}$$

where $c_\epsilon = -\sum_{t=2}^{\infty} \ln(1 - t^{-1-\epsilon})$. It is clear that the bound (7a) is far from optimal when t_1 is large: c_ϵ can be replaced by a quantity that tends to 0 as $O(t_1^{-\epsilon})$ as $t_1 \to \infty$. In particular, for $\epsilon = 1$ we have the bound

$$\ln N + 2 \ln t_1 + \ln 2.$$

An interesting feature of this switching rate is that for the full interval $[t_1, t_2] = [1, T]$ the bound differs from the standard AA bound only by an additive term less than 1. In words, the overhead for small adaptive regret is negligible.

4.2 Lower bounds on adaptive regret

One may wonder how good this worst-case adaptive regret bound for Fixed Share is, if we compare to some other algorithm. We now argue that it cannot be improved. First we show an information-theoretic lower bound on the adaptive regret of any algorithm. Then we show that Fixed Share meets this bound.

Theorem 7. *Let $\phi(t_1, t_2, N)$ be the worst-case adaptive regret of any algorithm. Then for all T and for all N*

$$\sum_{m=1}^{T} \sum_{1=t_1<\ldots<t_{m+1}=T+1} N(N-1)^{m-1} e^{-\sum_{j=1}^{m} \phi(t_j, t_{j+1}-1, N)} \;\leq\; 1. \qquad (8)$$

Proof. Fix an algorithm, time horizon T and expert count N. For any sequence $\boldsymbol{e} \in \{1, \ldots, N\}^T$ we define the loss pattern $(\ell_t^n)_{t\in[T]}^{n\in[N]}$ by

$$\ell_t^n \;=\; -\ln \mathbf{1}_{\{n=e_t\}}$$

(where $\mathbf{1}_{\{n=e_t\}} = 1$ if $n = e_t$ and $\mathbf{1}_{\{n=e_t\}} = 0$ otherwise). Let $L(\boldsymbol{e})$ be the loss of the algorithm on this loss pattern. Define the weight $w(\boldsymbol{e}) = e^{-L(\boldsymbol{e})}$. Clearly w is a probability distribution on $[N]^T$. Now let $t_2 < \ldots < t_m$ enumerate the internal block start indices $\big\{t \in \{2, \ldots, T\} \;\big|\; e_{t-1} \neq e_t\big\}$, and for the boundary set $t_1 = 1$ and $t_{m+1} = T + 1$. Since ϕ is the worst-case adaptive regret, and the loss of the best expert on each block is 0, we must have

$$L(\boldsymbol{e}) \;\leq\; \sum_{j=1}^{m} \phi(t_j, t_{j+1} - 1, N).$$

The theorem is obtained by negating and exponentiating this inequality, summing it over $[N]^T$, and grouping the contributions of sequences that agree on their block start indices. □

This bound is worthwhile because it is tight as we will see momentarily. It is however somewhat esoteric to interpret. It may be readily relaxed to imply for example that the bounds in (1) are tight, to a certain accuracy.

We will be interested in the performance guarantees that are *separable*, i.e., in upper bounds on $\phi(t_1, t_2, N)$ of the form $A(t_1) + B(t_2)$ (the number N of experts is fixed and omitted from our notation).

Corollary 8. *Suppose $\phi(t_1, t_2, N) \le A(t_1) + B(t_2)$ for all t_1 and t_2. Then for all T,*

$$\ln N - A(1) - B(T) + \sum_{t=2}^{T} \ln\left(1 + (N-1)e^{-A(t)-B(t-1)}\right) \;\le\; 0. \qquad (9)$$

Proof. Substitute the constraint on ϕ into (8). □

The following corollary shows that the stronger form (5) of (1a) is essentially tight: we cannot improve the right-hand side of (5a) by a constant, even for large t_1 and t_2, unless (5b) is relaxed drastically (it is not sufficient to replace $\ln N$ by D and ignore all $t_2 < T_0$ for arbitrarily large D and T_0).

Corollary 9. *Fix the number of experts N, a constant $C < \ln(N-1)$, and arbitrarily large positive integer constants D and T_0. No algorithm has worst-case adaptive regret*

$$\phi(t_1, t_2, N) \;\le\; C + \ln t_2 + \infty 1_{\{t_2 < T_0\}} + D 1_{\{t_1 = 1\}} + \infty 1_{\{1 < t_1 \le T_0\}}. \qquad (10)$$

Proof. Setting

$$A(t) = \begin{cases} D & \text{if } t = 1 \\ 0 & \text{if } t > T_0 \\ \infty & \text{otherwise} \end{cases} \quad \text{and} \quad B(t) = \begin{cases} \ln t + C & \text{if } t \ge T_0 \\ \infty & \text{otherwise} \end{cases}$$

on the right-hand side of (9) we obtain

$$\ln N - D - \ln T - C + \sum_{t=T_0+1}^{T} \ln\left(1 + (N-1)\frac{e^{-C}}{t-1}\right) \ge -\ln T + \frac{N-1}{e^C}\ln T - O(1)$$

which tends to ∞ as $T \to \infty$ (the inequality follows from the inequality $\ln(1+x) \ge x - x^2$, where $x \ge -1/2$). This contradicts (9). □

Our next corollary is about the tightness of (1b) and its elaboration (6) (see also the discussion following (6)).

Corollary 10. *Fix the number of experts N, a constant $C < \ln(N-1)$, and positive integer D and T_0. No algorithm has worst-case adaptive regret*

$$\phi(t_1, t_2, N) \;\le\; C + \ln t_1 + \ln\ln t_2 + \infty 1_{\{t_2 < T_0\}} + D 1_{\{t_1 = 1\}} + \infty 1_{\{1 < t_1 \le T_0\}}. \qquad (11)$$

Proof. Setting

$$A(t) = \begin{cases} D & \text{if } t = 1 \\ \ln t & \text{if } t > T_0 \\ \infty & \text{otherwise} \end{cases} \quad \text{and} \quad B(t) = \begin{cases} \ln\ln t + C & \text{if } t \ge T_0 \\ \infty & \text{otherwise} \end{cases}$$

on the right-hand side of (9) we now have

$$\ln N - D - \ln\ln T - C + \sum_{t=T_0+1}^{T} \ln\left(1 + (N-1)\frac{e^{-C}}{t\ln(t-1)}\right)$$
$$\geq -\ln\ln T + \frac{N-1}{e^C}\ln\ln(T-1) - O(1) \to \infty \qquad (T \to \infty). \qquad \square$$

Finally, we explore the tightness of (1c) (and its elaboration given later in the paper: see (7) and the discussion afterwards).

Corollary 11. *Fix the number of experts N, a constant $\epsilon > 0$, and a constant $a < \sum_{t=2}^{\infty} \ln(1 + t^{-1-\epsilon})$. No algorithm has worst-case adaptive regret*

$$\phi(t_1, t_2, N) \ \leq \ \begin{cases} \ln N + a & \text{if } t_1 = 1 \\ \ln(N-1) + (1+\epsilon)\ln t_1 & \text{otherwise.} \end{cases} \tag{12}$$

Proof. Setting

$$A(t) = \begin{cases} \ln N + a & \text{if } t = 1 \\ \ln(N-1) + (1+\epsilon)\ln t & \text{otherwise} \end{cases}$$

and $B(t) = 0$ on the right-hand side of (9) now gives

$$\ln N - \ln N - a + \sum_{t=2}^{T} \ln\left(1 + (N-1)e^{-\ln(N-1)-(1+\epsilon)\ln t}\right) > 0$$

for a sufficiently large T. $\square$

4.3 Fixed Share has optimal adaptive worst-case regret

We now prove that Fixed Share is optimal, in the sense that no algorithm can have a worst-case adaptive regret that is nowhere worse.

Corollary 12. *Fix any switching rate $(\alpha_t)_{t\geq 1}$, and let $\phi(t_1, t_2, N)$ be the worst-case adaptive regret of FS. Then (8) holds with equality.*

Proof. Plug the worst-case adaptive regret (4) into the sum (8). $\square$

5 Conclusion

We examined the problem of guaranteeing small adaptive regret for the setting of prediction with expert advice. In the first part we considered two techniques to obtain adaptive algorithms: using virtual specialist experts and restarting classical algorithms. We showed that both can be viewed as Fixed Share with a variable switching rate. In the second part we computed the exact worst-case adaptive regret for Fixed Share, thus tightening the existing upper bounds. So much, in fact, that by summing these worst-case regrets over a partition of the interval $[1, T]$ we recover the standard Fixed Share tracking bound. This formally establishes the complete congruence between adaptive and tracking performance, which was intuitive but not apparent from previously obtained adaptive bounds.

We then showed that Fixed Share is Pareto-optimal, in the sense that no algorithm can ensure better adaptive regret. We presented an information-theoretic lower bound on the worst-case adaptive regret of any algorithm, and showed that it holds with equality for Fixed Share.

Open problem Whereas upper bounds readily transfer to mixable losses, obtaining adaptive regret lower bounds for mixable losses is much more tricky. It is fair to call the lower bound argument in [15] for classical regret complicated, and this would be a special case for adaptive regret lower bounds.

Acknowledgments First author supported by Veterinary Laboratories Agency (VLA) of Department for Environment, Food and Rural Affairs (Defra). Second author supported by NWO Rubicon grant 680-50-1010. Third author supported by EPSRC grant EP/I030328/1.

References

1. Bousquet, O., Warmuth, M.K.: Tracking a small set of experts by mixing past posteriors. Journal of Machine Learning Research 3, 363–396 (2002)
2. Cesa-Bianchi, N., Gaillard, P., Lugosi, G., Stoltz, G.: A new look at shifting regret. CoRR abs/1202.3323 (2012)
3. Cesa-Bianchi, N., Lugosi, G.: Prediction, Learning, and Games. Cambridge University Press (2006)
4. Chernov, A., Vovk, V.: Prediction with expert evaluators' advice. In: Gavaldà, R., Lugosi, G., Zeugmann, T., Zilles, S. (eds.) ALT. Lecture Notes in Computer Science, vol. 5809, pp. 8–22. Springer (2009)
5. Freund, Y., Schapire, R.E., Singer, Y., Warmuth, M.K.: Using and combining predictors that specialize. In: Proc. 29th Annual ACM Symposium on Theory of Computing. pp. 334–343. ACM (1997)
6. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55, 119–139 (1997)
7. Hazan, E., Seshadhri, C.: Efficient learning algorithms for changing environments. In: ICML. p. 50 (2009)
8. Herbster, M., Warmuth, M.K.: Tracking the best expert. Machine Learning 32(2), 151–178 (1998)
9. Koolen, W.M.: Combining Strategies Efficiently: High-quality Decisions from Conflicting Advice. Ph.D. thesis, Institute of Logic, Language and Computation (ILLC), University of Amsterdam (Jan 2011)
10. Koolen, W.M., de Rooij, S.: Combining expert advice efficiently. In: Servedio, R., Zang, T. (eds.) Proceedings of the 21st Annual Conference on Learning Theory (COLT 2008). pp. 275–286 (Jun 2008)

11. Littlestone, N., Warmuth, M.K.: The weighted majority algorithm. Inf. Comput. 108(2), 212–261 (1994)
12. Shamir, G.I., Merhav, N.: Low complexity sequential lossless coding for piecewise stationary memoryless sources. IEEE Trans. Info. Theory 45, 1498–1519 (1999)
13. Vovk, V.: Aggregating strategies. In: Proceedings of the Third Annual Workshop on Computational Learning Theory. pp. 371–383. Morgan Kaufmann (1990)
14. Vovk, V.: Competitive on-line statistics. International Statistical Review 69, 213–248 (2001)
15. Vovk, V.: A game of prediction with expert advice. Journal of Computer and System Sciences 56, 153–173 (1998)
16. Zinkevich, M.: Online convex programming and generalized infinitesimal gradient ascent. In: Proc. 20th Int. Conference on Machine Learning (ICML '03). pp. 928–936 (2003)

A Worst-case adaptive regret data for Fixed Share

In this subsection we prove that the worst-case data for Fixed Share has the following form. On the interval $[t_1, t_2]$ we are interested in all but one expert suffer infinite loss and on the step preceding t_1 (if $t_1 \neq 1$) this one expert suffers infinite loss himself. The construction is iterative and we start constructing the data from the end of the interval.

Lemma 13. *For any history prior to the step t_2 the adaptive regret $R^j_{[t_1,t_2]}$ w.r.t. expert j on the interval $[t_1, t_2]$ is maximised with $\ell^k_{t_2} = \infty$ for $k \neq j$.*

Proof. Let us differentiate the adaptive regret w.r.t. $\ell^k_{t_2}$:

$$\frac{\partial R^j_{[t_1,t_2]}}{\partial \ell^k_{t_2}} = \frac{u^k_{t_2} e^{-\ell^k_{t_2}}}{\sum u^i_{t_2} e^{-\ell^i_{t_2}}} - \mathbf{1}_{\{j=k\}}$$

□

We can see that it is positive for all $k \neq j$ and becomes zero for $k = j$ when we plug in $\ell^k_{t_2} = \infty$ for those.

Lemma 14. *Fix an comparator expert j. Let $t \in [t_1, t_2]$. Suppose that the losses for steps $s = t+1, \ldots, t_2$ satisfy $\ell^k_s = \infty$ for $k \neq j$. Then the adaptive regret $R^j_{[t_1,t_2]}$ is maximised with $\ell^k_t = \infty$ for $k \neq j$.*

Proof. Let us start with showing that the if on the steps $t+1$ and $t+2$ the data is organised as we want to, that is j-th expert is good and all others suffer infinite loss, then Learner's loss on step $t+2$ is not dependent on what happens at time t and before. This follows immediately from (3), as

$$\ell_{t+2} = -\ln(1 - \alpha_{t+2}).$$

Now let us differentiate the adaptive regret $R^j_{[t_1,t_2]}$ w.r.t. ℓ^k_t assuming that the future losses are set up as we want. Let us show that the derivatives w.r.t. ℓ^k_t where $k \neq j$ are all positive. For those,

$$\frac{\partial R^j_{[t_1,t_2]}}{\partial \ell^k_t} = \frac{\partial \ell_t}{\partial \ell^k_t} + \frac{\partial \ell_{t+1}}{\partial \ell^k_t}$$

Expanding the second one gives (as before, $k \neq j$):

$$\begin{aligned}\frac{\partial \ell_{t+1}}{\partial \ell^k_t} &= \frac{\partial}{\partial \ell^k_t} - \ln\left(\frac{\alpha_{t+1}}{N-1} + (1 - \frac{N}{N-1}\alpha_{t+1})u^j_t e^{\ell_t - \ell^j_t}\right) \\ &= -\frac{(1 - \frac{N}{N-1}\alpha_{t+1})u^j_t e^{\ell_t - \ell^j_t}\frac{\partial}{\partial \ell^k_t}\ell_t}{\frac{\alpha_{t+1}}{N-1} + (1 - \frac{N}{N-1}\alpha_{t+1})u^j_t e^{\ell_t - \ell^j_t}}\end{aligned}$$

So we see that $\frac{\partial R^j_{[t_1,t_2]}}{\partial \ell^k_t} = \frac{\partial \ell_t}{\partial \ell^k_t}\left(\frac{\frac{\alpha_{t+1}}{N-1}}{\frac{\alpha_{t+1}}{N-1} + (1 - \frac{N}{N-1}\alpha_{t+1})u^j_t e^{\ell_t - \ell^j_t}}\right) > 0$. So our worst-case pattern of losses extends one trial backwards. □

Finally, we need to state the almost obvious fact that in order to maximise the adaptive regret we need to insert an infinite loss for the comparator expert right before the start of the interval, thus killing all the previous weight on him.

Lemma 15. *Fix a comparator expert j. Suppose that the losses for steps $s = t_1, \ldots, t_2$ satisfy $\ell^k_s = \infty$ for $k \neq j$. Then the adaptive regret $R^j_{[t_1,t_2]}$ is maximised with $\ell^j_{t-1} = \infty$.*

Proof. As before, the adaptive regret on steps starting from $t_1 + 1$ does not depend on $\ell^k_{t_1-1}$. So let us show that $\frac{\partial R^j_{[t_1,t_2]}}{\partial \ell^j_{t_1-1}} > 0$. We can reuse the proofs of previous lemmas for that:

$$\begin{aligned}\frac{\partial R^j_{[t_1,t_2]}}{\partial \ell^j_{t_1-1}} &= \frac{\partial \ell_{t_1}}{\partial \ell^j_{t_1-1}} \\ &= -\frac{(1 - \frac{N}{N-1}\alpha_{t_1})u^j_{t_1-1}e^{\ell_{t_1-1} - \ell^j_{t_1-1}}}{\frac{\alpha_{t_1}}{N-1} + (1 - \frac{N}{N-1}\alpha_{t_1})u^j_{t_1-1}e^{\ell_{t_1-1} - \ell^j_{t_1-1}}}\frac{\partial\left(\ell_{t_1-1} - \ell^j_{t_1-1}\right)}{\partial \ell^j_{t_1-1}} > 0,\end{aligned}$$

since $\frac{\partial\left(\ell_{t_1-1} - \ell^j_{t_1-1}\right)}{\partial \ell^j_{t_1-1}}$ is negative as follows from the proof of Lemma 13. □