Learning the Learning Rate for Prediction with Expert Advice

Wouter M. Koolen
Queensland University of Technology and UC Berkeley
wouter.koolen@qut.edu.au

Tim van Erven
Leiden University, the Netherlands
tim@timvanerven.nl

Peter D. Grünwald
Leiden University and Centrum Wiskunde & Informatica, the Netherlands
pdg@cwi.nl

Abstract

Most standard algorithms for prediction with expert advice depend on a parameter called the learning rate. This learning rate needs to be large enough to fit the data well, but small enough to prevent overfitting. For the exponential weights algorithm, a sequence of prior work has established theoretical guarantees for higher and higher data-dependent tunings of the learning rate, which allow for increasingly aggressive learning. But in practice such theoretical tunings often still perform worse (as measured by their regret) than ad hoc tuning with an even higher learning rate. To close the gap between theory and practice we introduce an approach to learn the learning rate. Up to a factor that is at most (poly)logarithmic in the number of experts and the inverse of the learning rate, our method performs as well as if we would know the empirically best learning rate from a large range that includes both conservative small values and values that are much higher than those for which formal guarantees were previously available. Our method employs a grid of learning rates, yet runs in linear time regardless of the size of the grid.

1 Introduction

Consider a learner who in each round $t = 1, 2, \dots$ specifies a probability distribution $\boldsymbol{w}_t$ on K experts, before being told a vector $\boldsymbol{\ell}_t \in [0,1]^K$ with their losses and consequently incurring loss $h_t := \boldsymbol{w}_t \cdot \boldsymbol{\ell}_t$. Losses are summed up over trials and after T rounds the learner's cumulative loss $H_T = \sum_{t=1}^T h_t$ is compared to the cumulative losses $L_T^k = \sum_{t=1}^T \ell_t^k$ of the experts $k = 1, \dots, K$. This is essentially the framework of *prediction with expert advice* [1, 2], in particular the standard *Hedge setting* [3]. Ideally, the learner's predictions would not be much worse than those of the best expert, who has cumulative loss $L_T^* = \min_k L_T^k$, so that the *regret* $\mathcal{R}_T = H_T - L_T^*$ is small.

Follow-the-Leader (FTL) is a natural strategy for the learner. In any round t, it predicts with a point mass on the expert k with minimum loss L_{t-1}^k, i.e. the expert that was best on the previous $t-1$ rounds. However, in the standard game-theoretic analysis, the experts' losses are assumed to be generated by an adversary, and then the regret for FTL can grow linearly in T [4], which means that it is not learning. To do better, the predictions need to be less outspoken, which can be accomplished by replacing FTL's choice of the expert with minimal cumulative loss by the soft minimum $w_t^k \propto e^{-\eta L_{t-1}^k}$, which is known as the *exponential weights* or *Hedge* algorithm [3]. Here $\eta > 0$ is a regularisation parameter that is called the *learning rate*. As $\eta \to \infty$ the soft minimum approaches the exact minimum and exponential weights converges to FTL. In contrast, the lower η, the more the soft minimum resembles a uniform distribution and the more conservative the learner.

Let $\mathcal{R}_T^\eta$ denote the regret for exponential weights with learning rate η. To obtain guarantees against adversarial losses, several tunings of η have been proposed in the literature. Most of these may be understood by starting with the bound

$$\mathcal{R}_T^\eta \leq \frac{\ln K}{\eta} + \sum_{t=1}^{T} \delta_t^\eta, \tag{1}$$

which holds for any sequence of losses. Here $\delta_t^\eta \geq 0$ is the approximation error (called *mixability gap* by [5]) when the loss of the learner in round t is approximated by the so-called *mix loss*, which is a certain η-exp-concave lower bound (see Section 2.1). The analysis then proceeds by giving an upper bound $b_t(\eta) \geq \delta_t^\eta$ and choosing η to balance the two terms $\ln(K)/\eta$ and $\sum_t b_t(\eta)$. In particular, the bound $\delta_t^\eta \leq \eta/8$ results in the most conservative tuning $\eta = \sqrt{8\ln(K)/T}$, for which the regret is always bounded by $O(\sqrt{T\ln(K)})$; the same guarantee can still be achieved even if the horizon T is unknown in advance by using, for instance, the so-called doubling trick [4]. It is possible though to learn more aggressively by using a bound on δ_t^η that depends on the data. The first such improvement can be obtained by using $\delta_t^\eta \leq e^\eta \boldsymbol{w}_t \cdot \boldsymbol{\ell}_t$ and choosing $\eta = \ln(1 + \sqrt{2\ln(K)/L_T^*}) \approx \sqrt{2\ln(K)/L_T^*}$, where again the doubling trick can be used if L_T^* is unknown in advance, which leads to a bound of $O(\sqrt{L_T^* \ln(K)} + \ln K)$ [6, 4]. Since $L_T^* \leq T$ this is never worse than the conservative tuning, and it can be better if the best expert has very small losses (a case sometimes called the "low noise condition"). A further improvement has been proposed by Cesa-Bianchi *et al.* [7], who bound δ_t^η by a constant times the variance v_t^η of ℓ_t^k when k is distributed according to $\boldsymbol{w}_t$, such that $v_t^\eta = \boldsymbol{w}_t \cdot (\boldsymbol{\ell}_t - h_t)^2$. Rather than using a constant learning rate, at time t they play the Hedge weights $\boldsymbol{w}_t$ based on a time-varying learning rate η_t that is approximately tuned as $\sqrt{\ln(K)/V_{t-1}}$ with $V_t = \sum_{s \leq t} v_s^{\eta_s}$. This leads to a so-called *second-order* bound on the regret of the form

$$\mathcal{R}_T = O\left(\sqrt{V_t \ln(K)} + \ln K\right), \tag{2}$$

which, as Cesa-Bianchi *et al.* show, implies

$$\mathcal{R}_T = O\left(\sqrt{\frac{L_T^*(T - L_T^*)}{T}\ln(K)} + \ln K\right) \tag{3}$$

and is therefore always better than the tuning in terms of L_T^* (note though that (2) can be much stronger than (3) on data for which the exponential weights rapidly concentrate on a single expert, see also [8]). The general pattern that emerges is that the better the bound on δ_t^η, the higher η can be chosen and the more aggressive the learning. De Rooij *et al.* [5] take this approach to its extreme and do not bound δ_t^η at all. In their *AdaHedge* algorithm they tune $\eta_t = \ln(K)/\Delta_{t-1}$ where $\Delta_t = \sum_{s \leq t} \delta_s^{\eta_s}$, which is very similar to the second-order tuning of Cesa-Bianchi *et al.* and indeed also satisfies (2) and (3). Thus, this sequence of prior works appears to have reached the limit of what is possible based on improving the bound on δ_t^η. Unfortunately, however, if the data are not adversarial, then even second-order bounds do not guarantee the best possible tuning of η for the data at hand. (See the experiments that study the influence of η in [5].) In practice, selecting η_t to be the best-performing learning rate so far (that is, running FTL at the meta-level) appears to work well [9], but this approach requires a computationally intensive grid search over learning rates [9] and formal guarantees can only be given for independent and identically distributed (IID) data [10]. A new technique based on speculatively trying out different η was therefore introduced in the *FlipFlop* algorithm [5]. By alternating learning rates $\eta_t = \infty$ and η_t that are very similar to those of AdaHedge, FlipFlop is both able to satisfy the second-order bounds (2) and (3), and to guarantee that its regret is never much worse than the regret $\mathcal{R}_T^\infty$ for Follow-the-Leader:

$$\mathcal{R}_T = O\big(\mathcal{R}_T^\infty\big). \tag{4}$$

Thus FlipFlop covers two extremes: on the one hand it is able to compete with η that are small enough to deal with the worst case, and on the other hand it can compete with $\eta = \infty$ (FTL).

Main Contribution We generalise the FlipFlop approach to cover a large range of η in between. As before, let $\mathcal{R}_T^\eta$ denote the regret of exponential weights with fixed learning rate η. We introduce the *learning the learning rate* (LLR) algorithm, which satisfies (2), (3) and (4) and in addition guarantees a regret satisfying

$$\mathcal{R}_T = O\left(\ln(K)\left(\ln\tfrac{1}{\eta}\right)^{1+\varepsilon}\mathcal{R}_T^\eta\right) \qquad \text{for all } \eta \in [\eta_{t^*}^{\mathrm{ah}}, 1] \tag{5}$$

for any $\varepsilon > 0$. Thus, LLR performs almost as well as the learning rate $\hat{\eta}_T \in [\eta_{t^*}^{\mathrm{ah}}, 1]$ that is optimal with hindsight. Here the lower end-point $\eta_{t^*}^{\mathrm{ah}} \geq (1 - o(1))\sqrt{\ln(K)/T}$ (as follows from (28) below) is a data-dependent value that is sufficiently conservative (i.e. small) to provide second-order guarantees and consequently worst-case optimality. The upper end-point 1 is an artefact of the analysis, which we introduce because, for general losses in $[0, 1]^K$, we do not have a guarantee in terms of $\mathcal{R}_T^\eta$ for $1 < \eta < \infty$. For the special case of binary losses $\boldsymbol{\ell}_t \in \{0, 1\}^K$, however, we can say a bit more: as shown in Appendix B of the supplementary material, in this special case the LLR algorithm guarantees regret bounded by $\mathcal{R}_T = O(K\mathcal{R}_T^\eta)$ for all $\eta \in [1, \infty]$.

The additional factor $\ln(K)\ln^{1+\varepsilon}(1/\eta)$ in (5) comes from a prior on an exponentially spaced grid of η. It is logarithmic in the number of experts K, and its dependence on $1/\eta$ grows slower than $\ln^{1+\varepsilon}(1/\eta) \leq \ln^{1+\varepsilon}(1/\eta_{t^*}^{\mathrm{ah}}) = O(\ln^{1+\varepsilon}(T))$ for any $\varepsilon > 0$. For the optimally tuned $\hat{\eta}_T$, we have in mind regret that grows like $\mathcal{R}_T^{\hat{\eta}_T} = O(T^\alpha)$ for some $\alpha \in [0, 1/2]$, so an additional polylog factor seems a small price to pay to adapt to the right exponent α.

Although $\eta \geq \eta_{t^*}^{\mathrm{ah}}$ appear to be most important, the regret for LLR can also be related to $\mathcal{R}_T^\eta$ for lower η:

$$\mathcal{R}_T = O\left(\frac{\ln K}{\eta}\right) \qquad \text{for all } \eta < \eta_{t^*}^{\mathrm{ah}}, \tag{6}$$

which is not in terms of $\mathcal{R}_T^\eta$, but still improves on the standard bound (1) because $\delta_t^\eta \geq 0$ for all η.

The LLR algorithm takes two parameters, which determine the trade-off between constants in the bounds (2)–(6) above. Normally we would propose to set these parameters to moderate values, but if we do let them approach various limits, LLR becomes essentially the same as FlipFlop, AdaHedge or FTL (see Section 2).

We emphasise that we do not just have a bound on LLR that is unavailable for earlier methods; there also exist actual losses for which the optimal learning rate with hindsight $\hat{\eta}_T$ is fundamentally in between the robust learning rates chosen by AdaHedge and the aggressive choice $\eta = \infty$ of FTL. On such data, Hedge with fixed learning rate $\hat{\eta}_T$ performs significantly better than both these extremes; see Figure 1. In Appendix A we describe the data used to generate Figure 1 and explain why the regret obtained by LLR is significantly smaller than the regret of AdaHedge, FTL and all other tunings described above.

Computational Efficiency Although LLR employs a grid of η, it does not have to search over this grid. Instead, in each time step it only has to do computations for the single η that is active, and, as a consequence, it runs as fast as using exponential weights with a single fixed η, which is linear in K and T. LLR, as presented here, does store information about all the grid points, which requires $O(\ln(K)\ln(T))$ storage, but we describe a simple approximation that runs equally fast and only requires a constant amount of storage.

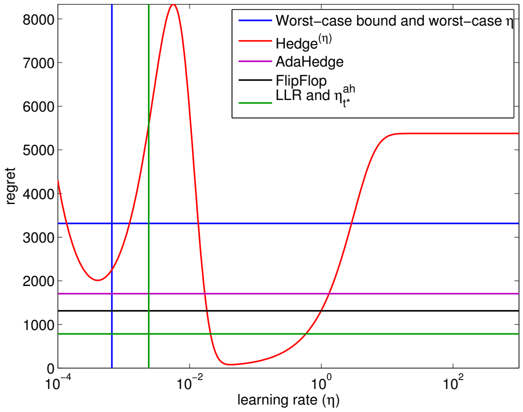

Figure 1: Example data (details in Appendix A) on which Hedge/exponential weights with intermediate learning rate (global minimum) performs much better than both the worst-case optimal learning rate (local minimum on the left) and large learning rates (plateau on the right). We also show the performance of the algorithms mentioned in the introduction.

Outline The paper is organized as follows. In Section 2 we define the LLR algorithm and in Section 3 we make precise how it satisfies (2), (3), (4), (5) and (6). Section 4 provides a discussion. Finally, the appendix contains a description of the data in Figure 1 and most of the proofs.

2 The Learning the Learning Rate Algorithm

In this section we describe the LLR algorithm, which is a particular strategy for choosing a time-varying learning rate in exponential weights. We start by formally describing the setting and then explain how LLR chooses its learning rates.

2.1 The Hedge Setting

At the start of each round $t = 1, 2, \dots$ the learner produces a probability distribution $\boldsymbol{w}_t = (w_t^1, \dots, w_t^K)$ on $K \geq 2$ experts. Then the experts incur losses $\boldsymbol{\ell}_t = (\ell_t^1, \dots, \ell_t^K) \in [0,1]^K$ and the learner's loss $h_t = \boldsymbol{w}_t \cdot \boldsymbol{\ell}_t = \sum_k w_t^k \ell_t^k$ is the expected loss under $\boldsymbol{w}_t$. After T rounds, the learner's cumulative loss is $H_T = \sum_{t=1}^T h_t$ and the cumulative losses for the experts are $L_T^k = \sum_{t=1}^T \ell_t^k$. The goal is to minimize the regret $\mathcal{R}_T = H_T - L_T^*$ with respect to the cumulative loss $L_T^* = \min_k L_T^k$ of the best expert. We consider strategies for the learner that play the exponential weights distribution

$$w_t^k = \frac{e^{-\eta_t L_{t-1}^k}}{\sum_{j=1}^K e^{-\eta_t L_{t-1}^j}}$$

for a choice of learning rate η_t that may depend on all losses before time t. To analyse such methods, it is common to approximate the learner's loss h_t by the *mix loss* $m_t = -\frac{1}{\eta_t} \ln \sum_k w_t^k e^{-\eta_t \ell_t^k}$, which appears under a variety of names in e.g. [7, 4, 11, 5]. The resulting approximation error or *mixability gap* $\delta_t = h_t - m_t$ is always non-negative and cannot exceed 1. This, and some other basic properties of the mix loss are listed in Lemma 1 of De Rooij *et al.* [5], which we reproduce as Lemma C.1 in the additional material.

As will be explained in the next section, LLR does not monitor the regrets of all learning rates directly. Instead, it tracks their cumulative mixability gaps, which provide a convenient lower bound on the regret that is monotonically increasing with the number of rounds T, in contrast to the regret itself. To show this, let $\mathcal{R}_T^\eta$ denote the regret of the exponential weights strategy with fixed learning rate $\eta_t = \eta$, and similarly let $M_T^\eta = \sum_{t=1}^T m_t^\eta$ and $\Delta_T^\eta = \sum_{t=1}^T \delta_t^\eta$ denote its cumulative mix loss and mixability gap.

Lemma 2.1. *For any fixed learning rate $\eta \in (0, \infty]$, the regret of exponential weights satisfies*

$$\mathcal{R}_T^\eta \geq \Delta_T^\eta. \tag{7}$$

Proof. Apply property 3 in Lemma C.1 to the regret decomposition $\mathcal{R}_T^\eta = M_T^\eta - L_T^* + \Delta_T^\eta$. □

We will use the following notational conventions. Lower-case letters indicate instantaneous quantities like m_t, δ_t and $\boldsymbol{w}_t$, whereas uppercase letters denote cumulative quantities like M_T, Δ_T and $\mathcal{R}_T$. In the absence of a superscript the learning rates present in any such quantity are those chosen by LLR. In contrast, the superscript $^\eta$ refers to using the same fixed learning rate η throughout.

2.2 LLR's Choice of Learning Rate

The LLR algorithm is a member of the exponential weights family of algorithms. Its defining property is its adaptive and non-monotonic selection of the learning rate η_t, which is specified in Algorithm 1 and explained next. The LLR algorithm works in regimes in which it speculatively tries out different strategies for η_t. Almost all of these strategies consist of choosing a fixed η from the following *grid*:

$$\eta^1 = \infty, \qquad \eta^i = \alpha^{2-i} \quad \text{for } i = 2, 3, \dots, \tag{8}$$

where the exponential base

$$\alpha = 1 + 1/\log_2 K \tag{9}$$

Algorithm 1 LLR($\pi^{\text{ah}}, \pi^{\infty}$). The grid $\eta^1, \eta^2, \ldots$ and weights $\pi^1, \pi^2, \ldots$ are defined in (8) and (12).

```
Initialise $b_0 := 0$; $\Delta_0^{\text{ah}} := 0$; $\Delta_0^i := 0$ for all $i \geq 1$.
for $t = 1, 2, \ldots$ do
    if all active indices and ah are $b_{t-1}$-full then
        Increase $b_t := \phi \Delta_{t-1}^{\text{ah}} / \pi^{\text{ah}}$ (with $\phi$ as defined in (14))
    else
        Keep $b_t := b_{t-1}$
    end if
    Let $i$ be the least non-$b_t$-full index.
    if $i$ is active then
        Play $\eta^i$.
        Update $\Delta_t^i := \Delta_{t-1}^i + \delta_t^i$. Keep $\Delta_t^j := \Delta_{t-1}^j$ for $j \neq i$ and $\Delta_t^{\text{ah}} := \Delta_{t-1}^{\text{ah}}$.
    else
        Play $\eta_t^{\text{ah}}$ as defined in (10).
        Update $\Delta_t^{\text{ah}} := \Delta_{t-1}^{\text{ah}} + \delta_t^{\text{ah}}$. Keep $\Delta_t^j := \Delta_{t-1}^j$ for all $j \geq 1$.
    end if
end for
```

is chosen to ensure that the grid is dense enough so that η^i for $i \geq 2$ is representative for all $\eta \in [\eta^{i+1}, \eta^i]$ (this is made precise in Lemma 3.3). We also include the special value $\eta^1 = \infty$, because it corresponds to FTL, which works well for IID data and data with a small number of leader changes, as discussed by De Rooij *et al.* [5].

For each *index* $i = 1, 2, \ldots$ in the grid, let $\mathcal{A}_t^i \subseteq \{1, \ldots, t\}$ denote the set of rounds up to trial t in which the LLR algorithm plays η^i. Then LLR keeps track of the performance of η^i by storing the sum of mixability gaps $\delta_t^i \equiv \delta_t^{\eta^i}$ for which η^i is responsible:

$$\Delta_t^i = \sum_{s \in \mathcal{A}_t^i} \delta_s^i.$$

In addition to the grid in (8), LLR considers one more strategy, which we will call the *AdaHedge* strategy, because it is very similar to the learning rate chosen by the AdaHedge algorithm [5]. In the AdaHedge strategy, LLR plays η_t equal to

$$\eta_t^{\text{ah}} = \frac{\ln K}{\Delta_{t-1}^{\text{ah}}}, \tag{10}$$

where $\Delta_t^{\text{ah}} = \sum_{s \in \mathcal{A}_t^{\text{ah}}} \delta_s^{\text{ah}}$ is the sum of mixability gaps $\delta_t^{\text{ah}} \equiv \delta_t^{\eta_t^{\text{ah}}}$ during the rounds $\mathcal{A}_t^{\text{ah}} \subseteq \{1, \ldots, t\}$ in which LLR plays the AdaHedge strategy. The only difference to the original AdaHedge is that the latter sums the mixability gaps over all $s \in \{1, \ldots, t\}$, not just those in $\mathcal{A}_t^{\text{ah}}$. Note that, in our variation, η_t^{ah} does not change during rounds outside $\mathcal{A}_t^{\text{ah}}$.

The AdaHedge learning rate η_t^{ah} is non-increasing with t, and (as we will show in Theorem 3.6 below) it is small enough to guarantee the worst-case bound (3), which is optimal for adversarial data. We therefore focus on $\eta > \eta_t^{\text{ah}}$ and call an index i in the grid *active* in round t if $\eta^i > \eta_t^{\text{ah}}$. Let $i_{\max} \equiv i_{\max}(t)$ be the number of grid indices that are active at time t, such that $\eta^{i_{\max}(t)} \approx \eta_t^{\text{ah}}$. Then LLR cyclically alternates grid learning rates and the AdaHedge learning rate, in a way that approximately maintains

$$\frac{\Delta_t^1}{\pi^1} \approx \frac{\Delta_t^2}{\pi^2} \approx \ldots \approx \frac{\Delta_t^{i_{\max}}}{\pi^{i_{\max}}} \approx \frac{\Delta_t^{\text{ah}}}{\pi^{\text{ah}}} \qquad \text{for all } t, \tag{11}$$

where $\pi^{\text{ah}} > 0$ and $\pi^1, \pi^2, \ldots > 0$ are fixed weights that control the relative importance of AdaHedge and the grid points (higher weight = more important). The LLR algorithm takes as parameters π^{ah} and π^{∞}, where π^{ah} only has to be positive, but π^{∞} is restricted to $(0, 1)$. We then choose

$$\pi^1 = \pi^{\infty}, \qquad \pi^i = (1 - \pi^{\infty})\rho(i-1) \qquad \text{for } i \geq 2, \tag{12}$$

where ρ is a prior probability distribution on $\{1, 2, \ldots\}$. It follows that $\sum_{i=1}^{\infty} \pi^i = 1$, so that π^i may be interpreted as a prior probability mass on grid index i. For ρ, we require a distribution with very heavy tails (meaning $\rho(i)$ not much smaller than $\frac{1}{i}$), and we fix the convenient choice

$$\rho(i) = \int_{\frac{i-1}{\ln K}}^{\frac{i}{\ln K}} \frac{1}{(x+e)\ln^2(x+e)} \, \mathrm{d}x = \frac{1}{\ln\left(\frac{i-1}{\ln K}+e\right)} - \frac{1}{\ln\left(\frac{i}{\ln K}+e\right)}. \tag{13}$$

We cannot guarantee that the invariant (11) holds exactly, and our algorithm incurs overhead for changing learning rates, so we do not want to change learning rates too often. LLR therefore uses an exponentially increasing budget b and tries grid indices and the AdaHedge strategy in sequence until they exhaust the budget. To make this precise, we say that an index i is b*-full* in round t if $\Delta_{t-1}^i/\pi^i > b$ and similarly that AdaHedge is b-full in round t if $\Delta_{t-1}^{\text{ah}}/\pi^{\text{ah}} > b$. Let b_t be the budget at time t, which LLR chooses as follows: first it initialises $b_0 = 0$ and then, for $t \geq 1$, it tests whether all active indices and AdaHedge are b_{t-1}-full. If this is the case, LLR approximately increases the budget by a factor $\phi > 1$ by setting $b_t = \phi\Delta_{t-1}^{\text{ah}}/\pi^{\text{ah}} > \phi b_{t-1}$, otherwise it just keeps the budget the same: $b_t = b_{t-1}$. In particular, we will fix budget multiplier

$$\phi = 1 + \sqrt{\pi^{\text{ah}}}, \tag{14}$$

which minimises the constants in our bounds. Now if, at time t, there exists an active index that is not b_t-full, then LLR plays the first such index. And if all active indices are b_t-full, LLR plays the AdaHedge strategy, which cannot be b_t-full in this case by definition of b_t. This guarantees that all ratios Δ_T^i/π_T^i are approximately within a factor ϕ of each other for all i that are active at time t^*, which we define to be the last time $t \leq T$ that LLR plays AdaHedge:

$$t^* = \max \mathcal{A}_T^{\text{ah}}. \tag{15}$$

Whenever LLR plays AdaHedge it is possible, however, that a new index i becomes active and it then takes a while for this index's cumulative mixability gap Δ_T^i to also grow up to the budget. Since AdaHedge is not played while the new index is catching up, the ratio guarantee always still holds for all indices that were active at time t^*.

2.3 Choosing the LLR Parameters

LLR has several existing strategies as sub-cases. For $\pi^{\text{ah}} \to \infty$ it essentially becomes AdaHedge. For $\pi^\infty \to 1$ it becomes FlipFlop. For $\pi^\infty \to 1$ and $\pi^{\text{ah}} \to 0$ it becomes FTL. Intermediate values for π^{ah} and π^∞ retain the benefits of these algorithms, but in addition allow LLR to compete with essentially all learning rates ranging from worst-case safe to extremely aggressive.

2.4 Run time and storage

LLR, as presented here, runs in constant time per round. This is because, in each round, it only needs to compute the weights and update the corresponding cumulative mixability gap for a single learning rate strategy. If the current strategy exceeds its budget (becomes b_t-full), LLR proceeds to the next[1]. The memory requirement is dominated by the storage of $\Delta_t^1, \ldots, \Delta_t^{i_{\max}(t)}$, which, following the discussion below (5), is at most

$$i_{\max}(T) = 2 + \frac{\ln \frac{1}{\eta^{i_{\max}(T)}}}{\ln \alpha} \leq 2 + \log_\alpha \frac{1}{\eta_T^{\text{ah}}} = O(\ln(K)\ln(T)).$$

However, a minor approximation reduces the memory requirement down to a constant: At any point in time the grid strategies considered by LLR split in three. Let us say that η^i is played at time t. Then all preceding η^j for $j \leq i$ are already at (or slightly past) the budget. And all succeeding η^j for $i < j \leq i_{\max}$ are still at (or slightly past) the previous budget. So we can approximate their cumulative mixability gaps by simply ignoring these slight overshoots. It then suffices to store only the cumulative mixability gap for the currently advancing η^i, and the current and previous budget.

[1] In the early stages it may happen that the next strategy is already over the budget and needs to be skipped, but this start-up effect quickly disappears when the budget exceeds 1, as the weighted increment $\delta_t^i/\pi^i \leq \eta^i/8\log^{1+\epsilon}(1/\eta)$ is bounded for all $0 \leq \eta \leq 1$.

3 Analysis of the LLR algorithm

In this section we analyse the regret of LLR. We first show that for each loss sequence the regret is bounded in terms of the cumulative mixability gaps Δ_T^i and Δ_T^{ah} incurred by the active learning rates (Lemma 3.1). As LLR keeps the cumulative mixability gaps approximately balanced according to (11), we can then further bound the regret in terms of each of the individual learning rates in the grid (Lemma 3.2). The next step is to deal with learning rates between grid points, by showing that their cumulative mixability gap Δ_T^η relates to Δ_T^i for the nearest higher grid point $\eta^i \geq \eta$ (Lemma 3.3). In Lemma 3.4 we put all these steps together. As the cumulative mixability gap Δ_T^η does not exceed the regret $\mathcal{R}_T^\eta$ for fixed learning rates (Lemma 2.1), we can then derive the bounds (2) through (6) from the introduction in Theorems 3.5 and 3.6.

We start by showing that the regret of LLR is bounded by the cumulative mixability gaps of the learning rates that it plays. The proof, which appears in Section C.4, is a generalisation of Lemma 12 in [5]. It crucially uses the fact that the lowest learning rate played by LLR is the AdaHedge rate η_t^{ah} which relates to Δ_t^{ah}.

Lemma 3.1. *On any sequence of losses, the regret of the LLR algorithm with parameters $\pi^{\text{ah}} > 0$ and $\pi^\infty \in (0, 1)$ is bounded by*

$$\mathcal{R}_T \leq \Big(\frac{\phi}{\phi - 1} + 2\Big)\Delta_T^{\text{ah}} + \sum_{i=1}^{i_{\max}} \Delta_T^i,$$

where $i_{\max}$ is the largest i such that η^i is active in round T and ϕ is defined in (14).

The LLR budgeting scheme keeps the cumulative mixability gaps from Lemma 3.1 approximately balanced according to (11). The next result, proved in Section C.5, makes this precise.

Lemma 3.2. *Fix t^* as in* (15). *Then for each index i that was active at time t^* and arbitrary $j \neq i$:*

$$\Delta_T^j \leq \phi\left(\frac{\pi^j}{\pi^i}\Delta_T^i + \frac{\pi^j}{\pi^{\text{ah}}}\right) + \min\{1, \eta^j/8\}, \tag{16a}$$

$$\Delta_T^j \leq \phi\frac{\pi^j}{\pi^{\text{ah}}}\Delta_T^{\text{ah}} + \min\{1, \eta^j/8\}, \tag{16b}$$

$$\Delta_T^{\text{ah}} \leq \frac{\pi^{\text{ah}}}{\pi^i}\Delta_T^i + 1. \tag{16c}$$

LLR employs an exponentially spaced grid of learning rates that are evaluated using — and played proportionally to — their cumulative mixability gaps. In the next step (which is restated and proved as Lemma C.7 in the additional material) we show that the mixability gap of a learning rate between grid-points cannot be much smaller than that of its next higher grid neighbour. This establishes in particular that an exponential grid is sufficiently fine.

Lemma 3.3. *For $\gamma \geq 1$ and for any sequence of losses with values in $[0, 1]$:*

$$\delta_t^{\gamma\eta} \leq \gamma e^{(\gamma-1)(\ln K+\eta)}\delta_t^\eta.$$

The preceding results now allow us to bound the regret of LLR in terms of the cumulative mixability gap of any fixed learning rate (which does not exceed its regret by Lemma 2.1) and in terms of the cumulative mixability gap of AdaHedge (which we will use to establish worst-case optimality).

Lemma 3.4. *Suppose the losses take values in $[0, 1]$, let $\pi^{\text{ah}} > 0$ and $\pi^\infty \in (0, 1)$ be the parameters of the LLR algorithm, and abbreviate $B = \big(\frac{\phi}{\phi-1} + 2\big)\pi^{\text{ah}} + \phi$. Then the regret of the LLR algorithm is bounded by*

$$\mathcal{R}_T \leq B\alpha e^{(\alpha-1)(\ln K+1)}\frac{\Delta_T^\eta}{\pi^{i(\eta)}} + \left(\frac{\alpha}{8(\alpha-1)} + \frac{\phi}{\pi^{\text{ah}}} + \frac{\phi}{\phi-1} + 3\right)$$

for all $\eta \in [\eta_{t^*}^{\text{ah}}, 1]$, where $i(\eta) = 2 + \lfloor \log_\alpha(1/\eta) \rfloor$ is the index of the nearest grid point above η, and by

$$\mathcal{R}_T \leq B\frac{\Delta_T^\infty}{\pi^\infty} + \left(\frac{\alpha}{8(\alpha-1)} + \frac{\phi}{\pi^{\text{ah}}} + \frac{\phi}{\phi-1} + 3\right)$$

for $\eta = \infty$. In addition

$$\mathcal{R}_T \leq B\frac{\Delta_T^{\text{ah}}}{\pi^{\text{ah}}} + \frac{\alpha}{8(\alpha - 1)} + 1,$$

and for any $\eta < \eta_{t^*}^{\text{ah}}$

$$\Delta_T^{\text{ah}} \leq \frac{\ln K}{\eta} + 1.$$

The proof appears in additional material Section C.6.

We are now ready for our main result, which is proved in Section C.7. It shows that LLR competes with the regret of any learning rate above the worst-case safe rate and below 1 modulo a mild factor. In addition, LLR also performs well on all data favoured by Follow-the-Leader.

Theorem 3.5. *Suppose the losses take values in $[0, 1]$, let $\pi^{\text{ah}} > 0$ and $\pi^{\infty} \in (0, 1)$ be the parameters of the LLR algorithm, and introduce the constants $B = 1 + 2\sqrt{\pi^{\text{ah}}} + 3\pi^{\text{ah}}$ and $C_K = (\log_2 K + 1)/8 + B/\pi^{\text{ah}} + 1$. Then the regret of LLR is simultaneously bounded by*

$$\mathcal{R}_T \leq \frac{4Be^1}{1 - \pi^{\infty}}(\log_2 K + 1)\underbrace{\ln(7/\eta)\ln^2\big(2\log_2(5/\eta)\big)}_{=O\left(\ln^{1+\varepsilon}(1/\eta)\right) \text{ for any } \varepsilon > 0}\mathcal{R}_T^{\eta} + C_K \qquad \text{for all } \eta \in [\eta_{t^*}^{\text{ah}}, 1]$$

and by

$$\mathcal{R}_T \leq \frac{B}{\pi^{\infty}}\mathcal{R}_T^{\infty} + C_K \qquad \text{for } \eta = \infty.$$

In addition

$$\mathcal{R}_T \leq \frac{B}{\pi^{\text{ah}}}\frac{\ln K}{\eta} + C_K \qquad \text{for any } \eta < \eta_{t^*}^{\text{ah}}.$$

To interpret the theorem, we recall from the introduction that $\ln(1/\eta)$ is better than $O(\ln T)$ for all $\eta \geq \eta_{t^*}^{\text{ah}}$.

We finally show that LLR is robust to the worst-case. We do this by showing something much stronger, namely that LLR guarantees a so-called second-order bound (a concept introduced in [7]). The bound is phrased in terms of the cumulative variance $V_T = \sum_{t=1}^{T} v_t$, where $v_t = \mathbb{V}_{k \sim \boldsymbol{w}_t}\left[\ell_t^k\right]$ is the variance of ℓ_t^k for k distributed according to $\boldsymbol{w}_t$. See Section C.8 for the proof.

Theorem 3.6. *Suppose the losses take values in $[0, 1]$, let $\pi^{\text{ah}} > 0$ and $\pi^{\infty} \in (0, 1)$ be the parameters of the LLR algorithm, and introduce the constants $B = \left(\frac{\phi}{\phi - 1} + 2\right)\pi^{\text{ah}} + \phi$ and $C_K = (\log_2 K + 1)/8 + B/\pi^{\text{ah}} + 1$. Then the regret of LLR is bounded by*

$$\mathcal{R}_T \leq \frac{B}{\pi^{\text{ah}}}\sqrt{V_T \ln K} + \left(C_K + \frac{2B\ln K}{3\pi^{\text{ah}}}\right)$$

and consequently by

$$\mathcal{R}_T \leq \frac{B}{\pi^{\text{ah}}}\sqrt{\frac{L_T^*(T - L_T^*)}{T}\ln K} + 2\left(C_K + \frac{2B\ln K}{3\pi^{\text{ah}}} + \frac{B^2\ln K}{(\pi^{\text{ah}})^2}\right).$$

4 Discussion

We have shown that our new LLR algorithm is able to recover the same second-order bounds as previous methods, which guard against worst-case data by picking a small learning rate if necessary. What LLR adds is that, at the cost of a (poly)logarithmic overhead factor, it is also able to learn a range of higher learning rates η, which can potentially achieve much smaller regret (see Figure 1). This is accomplished by covering this range with a grid of sufficient granularity. The overhead factor depends on a prior on the grid, for which we have fixed a particular choice with a heavy tail. However, the algorithm would also work with any other prior, so if it were known a priori that certain values in the grid were of special importance, they could be given larger prior mass. Consequently, a more advanced analysis demonstrating that only a subset of learning rates could potentially be optimal (in the sense of minimizing the regret $\mathcal{R}_T^{\eta}$) would directly lead to factors of improvement in the algorithm. Thus we raise the open question: what is the smallest subset $\mathcal{E}$ of learning rates such that, for any data, the minimum of the regret over this subset $\min_{\eta \in \mathcal{E}} \mathcal{R}_T^{\eta}$ is approximately the same as the minimum $\min_{\eta} \mathcal{R}_T^{\eta}$ over all or a large range of learning rates?

References

[1] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

[2] V. Vovk. A game of prediction with expert advice. *Journal of Computer and System Sciences*, 56(2):153–173, 1998.

[3] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139, 1997.

[4] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

[5] S. de Rooij, T. van Erven, P. D. Grünwald, and W. M. Koolen. Follow the leader if you can, Hedge if you must. *Journal of Machine Learning Research*, 15:1281–1316, 2014.

[6] P. Auer, N. Cesa-Bianchi, and C. Gentile. Adaptive and self-confident on-line learning algorithms. *Journal of Computer and System Sciences*, 64:48–75, 2002.

[7] N. Cesa-Bianchi, Y. Mansour, and G. Stoltz. Improved second-order bounds for prediction with expert advice. *Machine Learning*, 66(2/3):321–352, 2007.

[8] T. van Erven, P. Grünwald, W. M. Koolen, and S. de Rooij. Adaptive hedge. In *Advances in Neural Information Processing Systems 24 (NIPS)*, 2011.

[9] M. Devaine, P. Gaillard, Y. Goude, and G. Stoltz. Forecasting electricity consumption by aggregating specialized experts; a review of the sequential aggregation of specialized experts, with an application to Slovakian and French country-wide one-day-ahead (half-)hourly predictions. *Machine Learning*, 90(2):231–260, 2013.

[10] P. Grünwald. The safe Bayesian: learning the learning rate via the mixability gap. In *Proceedings of the 23rd International Conference on Algorithmic Learning Theory (ALT)*. Springer, 2012.

[11] V. Vovk. Competitive on-line statistics. *International Statistical Review*, 69:213–248, 2001.

[12] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 1991.

Learning the Learning Rate for Prediction with Expert Advice: Additional Material

A Simulation Study

Figure 1 shows that an intermediate learning rate $\hat{\eta}_T$ can outperform both the robust small learning rates chosen by methods like AdaHedge and the aggressive large learning $\eta = \infty$ chosen by FTL. In this section, we first discuss the features of Figure 1 in more detail. Then we describe how we generated the underlying data and we explain why the regret obtained by LLR is significantly smaller than the regret of AdaHedge, FTL and the other methods described in the introduction.

A.1 Interpretation

The red line in Figure 1 shows the regret $\mathcal{R}_T^\eta$ of the exponential weights algorithm with a fixed learning rate $\eta_t = \eta$ as a function of η. Its minimum at $\hat{\eta}_T \approx 1/70$ is the optimal learning rate in hindsight, with corresponding regret $\mathcal{R}_T^{\hat{\eta}_T} \approx 100$. Two blue lines mark the conservative tuning $\eta_T^{\text{wc}} = \sqrt{8(\ln K)/T}$ described in the introduction, and the corresponding worst-case regret bound $\mathcal{R}_T^{\text{wc}} \leq \sqrt{T \ln(K)/2}$. As can be seen, both the bound for η_T^{wc} and its actual regret are substantially larger (about 2200) than the global minimum at $\hat{\eta}_T$. The regret of AdaHedge is indicated by the purple horizontal line, and this line may also be taken as indicative of the performance of the second-order tuning of Cesa-Bianchi *et al.* [7], which is very similar. Although smaller than the regret for the worst-case tuning η_T^{wc}, the regret for these second-order methods is still much larger than $\mathcal{R}_T^{\hat{\eta}_T}$. The reason is that these methods use learning rates that are too small. On the other hand, large learning rates (in particular $\eta = \infty$ as used by FTL) also perform much worse than the best possible learning rate, so it is important to find the right intermediate value. This is the objective of LLR (the green line; we used parameters $\pi^{\text{ah}} = \pi^\infty = 1/5$), which achieves the smallest regret of all adaptive algorithms described in the introduction. Thus, this data pattern illustrates that intermediate learning rates can be optimal on some data, and motivates the search for an adaptive algorithm like LLR that can learn them. The remaining gap between LLR and the optimal learning rate $\hat{\eta}_T$ is the price we pay for learning the learning rate.

A.2 Data Generating Process

We now explain the data generating process that was used to generate the data in Figure 1. There are $K = 3$ experts, which each receive $T = 2 \cdot 10^7$ losses. Our focus really is on experts 1 and 2, because the third expert always gets the maximal loss 1; we explain why we include it further below. On a high level, our method to generate the losses for experts 1 and 2 is as follows: there exist some data for which small η is much better than large η, and there also exist data for which large η is much better than small η. We simply alternate these two types of data, which ensures that some intermediate η will be the best. In practice, especially when the number of experts is large, there might be other, more complicated interactions between experts that cause intermediate η to be best, but our current approach seems to be the simplest illustration of this phenomenon.

More precisely, T losses for the experts are constructed according to Algorithm 2, which depends on parameters $\alpha > \beta$ and γ, for which we select the values $\alpha = 1/6$, $\beta = 1/14$ and $\gamma = 1/6$. The pattern of losses for experts 1 and 2 is constructed in four phases, which are repeated T^α times. Out of these, the crucial parts are Phase 1 and Phase 3, during which the difference in cumulative loss between experts 1 and 2 stays approximately constant, except that it goes up and down by 1 every two rounds, which we call *wiggling*. Phase 1 takes place at a particular cumulative loss difference designed to punish large learning rates and Phase 3 at another designed to punish small learning rates. Phases 2 and 4 simply take care of the transition from Phase 1 to Phase 3 and *vice versa*. For simplicity, we have ignored rounding issues in our rendering of Algorithm 2, which need to be taken care of to make sure that all phases have integer lengths and that at each end of Phase 4 we have $L_t^1 - L_t^2$ *exactly* equal to $1/2$.

Phase 1: Punish Large Learning Rates In Phase 1, which lasts $T^{1/2-\beta}$ rounds every time it is repeated, the difference in cumulative loss between experts 1 and 2 is approximately 0 and every

Algorithm 2 The Data Generating Process

```
Parameters: $T$, $0 < \alpha < 1/2$, $0 < \beta < \alpha$, $0 < \gamma < 1/2$
for $t = 1, 2, \ldots, T$ do                                  ▷ Expert 3 is always bad
    $\ell_t^3 \coloneqq 1$
end for
$\ell_1^1 \coloneqq 1/2$ ; $\ell_1^2 \coloneqq 0$ ; $t \coloneqq 2$                                  ▷ Tie-breaker
for $j \coloneqq 1, 2, \ldots, T^\alpha$ do
    for $i \coloneqq 1, 2, \ldots, T^{1/2-\beta}$ do                ▷ Phase 1: Wiggles, $|L_t^1 - L_t^2| = 1/2$
        $\ell_t^1 \coloneqq 0; \ell_t^2 \coloneqq 1$ ; $\ell_{t+1}^1 \coloneqq 1$ ; $\ell_t^2 \coloneqq 0$ ; $t \coloneqq t + 2$
    end for
    for $i = 1, 2, \ldots, T^{1/2-\gamma}$ do                ▷ Phase 2: Expert 2 gets better than 1
        $\ell_t^1 \coloneqq 1$ ; $\ell_t^2 \coloneqq 0$ ; $t \coloneqq t + 1$
    end for
    for $i = 1, 2, \ldots, T^{1-\alpha} - T^{1/2-\beta} - 2T^{1/2-\gamma}$ do    ▷ Phase 3: Wiggles, $L_t^1 - L_t^2 \approx T^{1/2-\gamma}$
        $\ell_t^1 \coloneqq 1$ ; $\ell_t^2 \coloneqq 0$ ; $\ell_{t+1}^1 \coloneqq 0$ ; $\ell_t^2 \coloneqq 1$ ; $t \coloneqq t + 2$
    end for
    for $i = 1, 2, \ldots, T^{1/2-\gamma}$ do                ▷ Phase 4: Expert 2 gets worse again
        $\ell_t^1 \coloneqq 0$ ; $\ell_t^2 \coloneqq 1$ ; $t \coloneqq t + 1$
    end for                                                    ▷ Now $L_t^1 - L_t^2 = 1/2$ again
end for
```

wiggle causes the leader (the expert with smallest cumulative loss) to change two times. As is well-known, each such leader change leads to an additional regret of $1/2$ for FTL. Since the total number of rounds spent in Phase 1 is $T^\alpha T^{1/2-\beta} = T^{1/2+\epsilon}$ for $\epsilon = \alpha - \beta > 0$, this ensures that FTL will incur regret of order strictly larger than $\sqrt{T}$ and hence will not be competitive with the standard worst-case learning rate η_T^{wc}, whose regret grows no faster than $O(\sqrt{T})$. Thus, in Phase 1, FTL and similar large learning rates are ruled out.

Phase 3: Punish Small Learning Rates In Phase 3, the difference in cumulative loss $L_t^1 - L_t^2$ between experts 1 and 2 is approximately $T^{1/2-\gamma}$. This distinguishes between small learning rates $\eta \ll T^{-1/2+\gamma}$ for which the exponential weights are not converged on a single expert during Phase 3 and larger learning rates $\eta \gg T^{-1/2+\gamma}$ for which the exponential weights are converged. Convergence of the weights is important, because it may be derived from Lemma C.2 below that the mixability gap can be approximated as

$$\delta_t^\eta \propto \eta v_t^\eta \qquad \text{for } \eta \leq 1,$$

where v_t^η is the variance of the exponential weights distribution at time t, which is approximately 0 if and only if the weights are converged on a single expert. And the mixability gap to a large extent controls the regret, which (1) and Lemma 2.1 show to be sandwiched by

$$\sum_{t=1}^T \delta_t^\eta \leq \mathcal{R}_T^\eta \leq \frac{\ln K}{\eta} + \sum_{t=1}^T \delta_t^\eta.$$

Since $\alpha < 1/2$, the fraction of rounds spent in Phase 3 goes to 1 as T tends to infinity, such that

$$\mathcal{R}_T^\eta \geq T\big(1 + o(1)\big)\eta \qquad \text{for } \eta \ll T^{-1/2+\gamma}.$$

This explains the spike in the regret for η between $\eta_T^{\text{wc}} \approx T^{-1/2} \approx 10^{-3.7}$ and $T^{-1/2+\gamma} \approx 10^{-2.4}$, and we see in Figure 1 that the spike continues for somewhat larger η as well.

If η becomes large enough, however, then Phase 3 stops hurting because v_t^η will be very small. This can quantified using Lemma 6 of Van Erven *et al.* [8], which bounds the mixability gap by

$$\delta_t^\eta \lesssim \eta(1 - \max_k w_t^k) \leq 1 - w_t^2 \qquad \text{for } \eta \leq 1.$$

Assuming that the weight of expert 3 is negligible, we have $1 - w_t^2 \approx \exp\big(-\eta T^{1/2-\gamma}\big)$, which is exponentially small in T for $\eta \gg T^{-1/2+\gamma}$. For such η the sum of mixability gaps over all repetitions of Phase 3 is therefore bounded by a constant.

This leaves room for a learning rate $\hat{\eta}_T$ that is significantly larger than $T^{-1/2+\gamma}$ (such that it is not hurt by Phase 3), but at the same time is not so large that it suffers from Phase 1. And indeed our experiments confirm that such an intermediate learning rate minimizes the regret. Choosing $\gamma = 0$, we already find an $\hat{\eta}_T$ that beats $\eta_T^{\text{wc}} \propto T^{-1/2}$ and FTL, but AdaHedge (which chooses a learning rate substantially higher than $T^{-1/2}$ when $\gamma = 0$) and hence FlipFlop are then still competitive with all η. By choosing γ slightly above 0, we find that there exists an $\hat{\eta}_T$ that also significantly beats AdaHedge and FlipFlop. As mentioned above, Figure 1 was obtained for $T = 2 \cdot 10^7$, $\alpha = 1/6$, $\beta = 1/14$ and $\gamma = 1/6$.

The Role of Expert 3 At the final time T, the cumulative losses of experts 1 and 2 differ by $1/2$, a constant. Therefore, if we were to leave out expert 3, which always gets maximal loss, it would actually be optimal to use learning rate 0, i.e. not learn anything at all: Hedge would then predict by a uniform mixture of expert 1 and 2 at all t, which would give a regret of at most $1/2$. Including expert 3 ensures that this trivial, non-learning version of Hedge does not perform well, for it would put mass $1/3$ at the bad expert 3. Indeed, if we repeat the experiment without this bad expert we end up with a figure that, unlike Figure 1, has no local minimum at $\eta_0 \approx 7 \cdot 10^{-3}$; the red curve is then increasing on $(0, \eta_0)$, while to the right of η_0 it still behaves just like in Figure 1.

B For Binary Losses, LLR Is Also Competitive with All $\eta \in [1, \infty]$.

The following result substantially generalizes the first implication in Theorem 18 of [5], who show that, for $K = 2$ experts and losses in $\{0, 1\}$, unbounded (as $T \to \infty$) regret for FTL implies unbounded regret for Hedge with constant learning rate $\eta_t = \eta$. Note that the case with losses in $\{0, 1\}$ corresponds to prediction with expert advice in which the experts always predict with a 0 or 1, the loss is the 0/1-loss, and the learner is allowed to judge randomized predictions by their expected loss.

Theorem B.1. *Fix any $0 < \eta < \infty$ and $K \in \mathbb{N}$. Consider a loss sequence $\boldsymbol{\ell}_1, \boldsymbol{\ell}_2, \ldots$ with each $\boldsymbol{\ell}_t \in \{0, 1\}^K$. Then there is a constant $C > 0$, depending on η, such that for all $T > 0$, $\mathcal{R}_T^\eta \geq C \cdot \mathcal{R}_T^\infty$. In particular, for $\eta \geq 1$, the inequality holds for $C \geq 1/(2eK)$.*

The theorem shows that, if one is only interested in regret bounds up to constant factors, and the losses of the experts are guaranteed to be in $\{0, 1\}$, then nothing is lost by only considering $\eta = \infty$ (FTL) and all $\eta < \eta_0$ where η_0 is some fixed constant; the precise constants, hidden in the result, depends on this choice of η_0. On the other hand, Example 2 of [5] shows that there are cases with losses in $\{0, 1\}$ in which the regret of FTL is bounded, whereas for $\eta = 1$, $\mathcal{R}^\eta$ increases linearly (!) in T. Hence, including $\eta = \infty$ is essential. This shows that in the special case of 0/1-valued losses, the LLR algorithm is really competitive with all interesting values of η as long as one is only interested in regret optimality up to constant factors.

Proof. Let $\hat{\mathcal{K}}_{t-1}$ be the set of leaders at time $t - 1$, i.e. the set of $k \in \{1, \ldots, K\}$ that achieve minimum cumulative loss at time $t - 1$. If there is no leader change at time t, i.e. if $\hat{\mathcal{K}}_{t-1} = \hat{\mathcal{K}}_t$, then $\ell_{t,k} = \ell_{t,k'}$ for all $k, k' \in \hat{\mathcal{K}}_{t-1}$ and FTL incurs no regret, i.e. $\mathcal{R}_t^\infty = \mathcal{R}_{t-1}^\infty$. The other possibility is that there is a leader change, at time t, i.e. there is a $k \in \hat{\mathcal{K}}_{t-1}$ with $k \notin \hat{\mathcal{K}}_t$. Then there must be an expert $k' \in \hat{\mathcal{K}}_t$ so that $L_{t,k'} < L_{t,k}$ (because k does not lead any more at time t) whereas $L_{t-1,k} \leq L_{t-1,k'}$ (because k leads at time $t - 1$). Since $L_{t-1,k}$ and $L_{t-1,k'}$ are integers and $\ell_{t,k}$ and $\ell_{t,k'}$ are both in $\{0, 1\}$, this implies that we must have $\ell_{t,k} = 1$, $\ell_{t,k'} = 0$, $L_{t-1,k} = L_{t-1,k'}$. It follows that $\hat{\mathcal{K}}_t \cap \hat{\mathcal{K}}_{t-1}$ is nonempty, and each k_0 in the intersection has $\ell_{t,k_0} = 0$, and each $k_1 \in \hat{\mathcal{K}}_{t-1} \setminus \hat{\mathcal{K}}_t$ has $\ell_{t+1,k_1} = 1$. Setting $K' \geq 1$ equal to the number of experts in $\hat{\mathcal{K}}_{t-1} \setminus \mathcal{K}_t$, it follows that

$$\mathcal{R}_t^\infty = \mathcal{R}_{t-1}^\infty + \frac{K'}{|\hat{\mathcal{K}}_{t-1}|} \leq \mathcal{R}_{t-1}^\infty + 1.$$

Thus, $\mathcal{R}_T^\infty \leq \#(\text{lc})$, where $\#(\text{lc})$ denotes the number of leader changes up till time T.

Below we prove that for every t with a leader change, $\delta_t^\eta \geq C$, where C is a constant, which is at least $1/(2eK)$ if $\eta \geq 1$. Since by Lemma C.1 below, at all other t', $\delta_{t'}^\eta \geq 0$, it follows that

$\Delta_T^\eta \geq C \cdot \#(\mathrm{lc}) \geq C\mathcal{R}_T^\infty$, where the final inequality was shown at the end of the previous paragraph. Since $\mathcal{R}_T^\eta \geq \Delta_T^\eta$ by Lemma 2.1, the result then follows.

Thus, it only remains to prove that $\delta_t^\eta \geq C$ with C as above if there is a leader change at time t. To see this, note that by Lemma C.2, we have for each t, $\delta_t \geq c_\eta v_t$ where $c_\eta = (e^{-\eta} + \eta - 1)/\eta$ is a constant depending on η which, by standard calculus, can be seen to be larger than 0 and increasing for all $\eta > 0$. Thus, for $\eta \geq 1$, $c_\eta \geq c_1 = e^{-1}$ and it is sufficient if we can show that, if there is a leader change at time t, then $v_t \geq 1/(2K)$. But we know that at time $t-1$, there must be at least two leaders (denoted k and k' above). Since these have maximal weights and weights sum to 1, both of these must have weight at least $1/K$. Using that $\ell_{t,k} = 1$ and $\ell_{t,k'} = 0$, we have

$$v_t = \boldsymbol{w}_t \cdot (\boldsymbol{\ell}_t - h_t)^2 \geq \frac{1}{K}(1-h_t)^2 + \frac{1}{K}h_t^2 \geq \frac{1}{2K},$$

where we used that $\min_{a\in[0,1]}(1-a)^2 + a^2 = 1/2$. This finishes the proof. □

C Proofs

This section is dedicated to the proofs referenced in the main exposition.

C.1 Lemma C.1: Basic Properties of the Mix Loss

The following lemma is proved in [5].

Lemma C.1 (Mix Loss with Constant Learning Rate). *For any learning rate $\eta \in (0, \infty]$*

1. $0 \leq m_t^\eta \leq h_t^\eta \leq 1$, *so that* $0 \leq \delta_t^\eta \leq 1$.
2. *Cumulative mix loss telescopes:* $M_T^\eta = \begin{cases} -\frac{1}{\eta}\ln\left(\sum_k w_1^k e^{-\eta L_T^k}\right) & \text{for } \eta < \infty, \\ L_T^* & \text{for } \eta = \infty. \end{cases}$
3. *Cumulative mix loss approximates the loss of the best expert:* $L_T^* \leq M_T^\eta \leq L_T^* + \dfrac{\ln K}{\eta}$.
4. *The cumulative mix loss M_T^η is non-increasing in η.*

C.2 Bernstein Sandwich

Here we show that the mixability gap δ_t is well approximated by the variance $\boldsymbol{v}_t = \boldsymbol{w}_t \cdot (\boldsymbol{\ell}_t - h_t)^2$ for small learning rates η.

Lemma C.2 (Bernstein Sandwich). *For $\boldsymbol{\ell}_t \in [0,1]^K$ and $\eta > 0$*

$$\frac{(e^{-\eta} + \eta - 1)}{\eta} v_t \leq \delta_t \leq \frac{(e^{\eta} - \eta - 1)}{\eta} v_t.$$

Proof. As $(e^x - x - 1)/x^2$ is increasing in x, all $x \in [-1, 1]$ satisfy

$$e^{-\eta} + \eta - 1 \leq \frac{e^{\eta x} - \eta x - 1}{x^2} \leq e^{\eta} - \eta - 1.$$

Combination with Lemma C.4 results in

$$(e^{-\eta} + \eta - 1)\min_\lambda \frac{1}{\eta}\sum_k w_k(\lambda - \ell_t^k)^2 \leq \delta_t \leq (e^{\eta} - \eta - 1)\min_\lambda \frac{1}{\eta}\sum_k w_k(\lambda - \ell_t^k)^2$$

The lemma follows by plugging in the optimizer $\lambda = \boldsymbol{w} \cdot \boldsymbol{\ell}_t$. □

C.3 Proof of Lemma 3.3, restated as Lemma C.7

We build up to the proof using a series of lemmas.

Lemma C.3. *Let $w_t^{\eta,k} = \frac{e^{-\eta L_{t-1}^k}}{\sum_j e^{-\eta L_{t-1}^j}}$ be the exponential weights distribution on K experts for learning rate $\eta > 0$ and let $\gamma \geq 1$. Then*

$$w_t^{\gamma\eta,k} \leq K^{\gamma-1} w_t^{\eta,k} \qquad \textit{for all } k. \tag{17}$$

Proof. By the log-sum inequality (see [12])

$$\begin{aligned}
\ln \frac{w_t^{\gamma\eta,k}}{w_t^{\eta,k}} &= \ln \frac{\sum_j e^{-\eta(L_{t-1}^j - L_{t-1}^k)}}{\sum_j e^{-\gamma\eta(L_{t-1}^j - L_{t-1}^k)}} \\
&\leq \sum_j \frac{e^{-\eta(L_{t-1}^j - L_{t-1}^k)}}{\sum_j e^{-\eta(L_{t-1}^j - L_{t-1}^k)}} \ln \frac{e^{-\eta(L_{t-1}^j - L_{t-1}^k)}}{e^{-\gamma\eta(L_{t-1}^j - L_{t-1}^k)}} \\
&= (\gamma - 1) \sum_j \frac{e^{-\eta(L_{t-1}^j - L_{t-1}^k)}}{\sum_j e^{-\eta(L_{t-1}^j - L_{t-1}^k)}} \eta(L_{t-1}^j - L_{t-1}^k) \\
&\leq (\gamma - 1) \Big(- \sum_j \frac{e^{-\eta(L_{t-1}^j - L_{t-1}^k)}}{\sum_j e^{-\eta(L_{t-1}^j - L_{t-1}^k)}} \ln \Big(\frac{e^{-\eta(L_{t-1}^j - L_{t-1}^k)}}{\sum_j e^{-\eta(L_{t-1}^j - L_{t-1}^k)}} \Big) \Big).
\end{aligned}$$

The second inequality follows by $\sum_j e^{-\eta(L_{t-1}^j - L_{t-1}^k)} \geq e^{-\eta(L_{t-1}^k - L_{t-1}^k)} = 1$. Upper bounding that Shannon entropy by $\ln K$ results in (17). □

Lemma C.4. *Fix any learning rate η and probability vector $\boldsymbol{w}$. Let $\delta_t^\eta(\boldsymbol{w}) = \boldsymbol{w} \cdot \boldsymbol{\ell}_t - m_t^\eta(\boldsymbol{w})$ be the mixability gap of $\boldsymbol{w}$, where $m_t^\eta(\boldsymbol{w}) = \frac{-1}{\eta} \ln \sum_k w_k e^{-\eta \ell_t^k}$ is the mix loss of $\boldsymbol{w}$. Then*

$$\delta_t^\eta(\boldsymbol{w}) = \min_\lambda \frac{1}{\eta} \sum_k w_k \psi\big(\eta(\lambda - \ell_t^k)\big)$$

where $\psi(x) = e^x - x - 1$ and the minimum is achieved by $\lambda = m_t^\eta(\boldsymbol{w})$.

Proof. Let $\triangle$ be the probability simplex on K outcomes. We will use that, up to scaling, the mix loss is the convex conjugate of the Kullback-Leibler divergence $D(\boldsymbol{v} \| \boldsymbol{w}) = \sum_k v_k \ln \frac{v_k}{w_k}$:

$$-\eta m_t^\eta(\boldsymbol{w}) = \sup_{\boldsymbol{v} \in \triangle} \boldsymbol{v} \cdot (-\eta \boldsymbol{\ell}_t) - D(\boldsymbol{v} \| \boldsymbol{w}).$$

As the Kullback-Leibler may be extended off the simplex to $D(\boldsymbol{v} \| \boldsymbol{w}) = \sum_k (v_k \ln \frac{v_k}{w_k} - v_k + w_k)$ for any vectors $\boldsymbol{v}$ and $\boldsymbol{w}$ with non-negative components, we may introduce a Lagrange multiplier λ to enforce the restriction to the simplex and reason as follows:

$$\begin{aligned}
m_t^\eta(\boldsymbol{w}) &= \inf_{\boldsymbol{v} \in \triangle} \frac{1}{\eta} D(\boldsymbol{v} \| \boldsymbol{w}) + \boldsymbol{v} \cdot \boldsymbol{\ell}_t \\
&= \sup_\lambda \inf_{\boldsymbol{v} \in \mathbb{R}_+^K} \frac{1}{\eta} D(\boldsymbol{v} \| \boldsymbol{w}) + \boldsymbol{v} \cdot \boldsymbol{\ell}_t - \lambda(\boldsymbol{1} \cdot \boldsymbol{v} - 1) \\
&= \sup_\lambda \frac{1}{\eta} \sum_k w_k \left(1 - e^{\eta(\lambda - \ell_t^k)}\right) + \lambda \\
&= \boldsymbol{w} \cdot \boldsymbol{\ell}_t - \inf_\lambda \frac{1}{\eta} \sum_k w_k \psi\left(\eta(\lambda - \ell_t^k)\right),
\end{aligned}$$

from which the result follows. □

Lemma C.5 (Continuous Log-Sum Inequality)**.** *Let $f, g\colon [a,b] \to \mathbb{R}$ be positive functions such that $\int_a^b g(x)\mathrm{d}x < \infty$. Then*

$$\ln \frac{\int_a^b f(x)\mathrm{d}x}{\int_a^b g(x)\mathrm{d}x} \le \int_a^b \frac{f(x)}{\int_a^b f(y)\mathrm{d}y} \ln \frac{f(x)}{g(x)} \mathrm{d}x.$$

Proof. Let $h(x) = f(x)/g(x)$. Then, by Jensen's inequality and convexity of $z \ln z$,

$$\begin{aligned}
\int_a^b \frac{f(x)}{\int_a^b g(y)\mathrm{d}y} \ln \frac{f(x)}{g(x)} \mathrm{d}x &= \int_a^b \frac{g(x)}{\int_a^b g(y)\mathrm{d}y} \Big(h(x) \ln h(x) \Big) \mathrm{d}x \\
&\ge \Big(\int_a^b \frac{g(x)}{\int_a^b g(y)\mathrm{d}y} h(x)\mathrm{d}x \Big) \ln \Big(\int_a^b \frac{g(x)}{\int_a^b g(y)\mathrm{d}y} h(x)\mathrm{d}x \Big) \\
&= \frac{\int_a^b f(x)\mathrm{d}x}{\int_a^b g(y)\mathrm{d}y} \ln \frac{\int_a^b f(x)\mathrm{d}x}{\int_a^b g(y)\mathrm{d}y}.
\end{aligned}$$

Dividing both sides by $\frac{\int_a^b f(x)\mathrm{d}x}{\int_a^b g(y)\mathrm{d}y}$, the result follows. □

Lemma C.6. *Let $\psi(x) = e^x - x - 1$. Then for $\gamma \ge 1$ and $x \le B$ for $B \ge 0$*

$$\frac{\psi(\gamma x)}{\psi(x)} \le \gamma^2 e^{(\gamma-1)B}.$$

Proof. We use that

$$\psi(x) = x^2 \int_0^1 (1-u) e^{xu} \, \mathrm{d}u.$$

By the log-sum inequality (c.f. Lemma C.5)

$$\begin{aligned}
\ln \frac{\psi(\gamma x)}{\psi(x)} &\le \int_0^1 \frac{(\gamma x)^2 (1-u) e^{\gamma x u}}{\psi(\gamma x)} \ln \frac{(\gamma x)^2 (1-u) e^{\gamma x u}}{x^2 (1-u) e^{xu}} \, \mathrm{d}u \\
&= \ln \gamma^2 + (\gamma - 1) x \int_0^1 \frac{(\gamma x)^2 (1-u) e^{\gamma x u}}{\psi(\gamma x)} u \, \mathrm{d}u \\
&\le \ln \gamma^2 + (\gamma - 1) B \int_0^1 \frac{(\gamma x)^2 (1-u) e^{\gamma x u}}{\psi(\gamma x)} u \, \mathrm{d}u \\
&\le \ln \gamma^2 + (\gamma - 1) B,
\end{aligned}$$

where the last inequality uses $u \le 1$. □

Lemma C.7. *Fix $\eta > 0$ and $\gamma \ge 1$. Let $\boldsymbol{w}^\eta$ be the exponential weight distribution with learning rate η (as defined in Lemma C.3) and let $\delta^\eta(\boldsymbol{w})$ be the mixability gap of $\boldsymbol{w}$ as defined in Lemma C.4. Then for any $\boldsymbol{\ell}_t \in [0,1]^K$*

$$\delta_t^{\gamma\eta}(\boldsymbol{w}^{\gamma\eta}) \le \gamma e^{(\gamma-1)\eta} K^{1-\gamma} \delta_t^\eta(\boldsymbol{w}^\eta).$$

Proof. Substituting the sub-optimal $\lambda = m_t^\eta(\boldsymbol{w}^\eta)$ into the expression for $\delta_t^{\gamma\eta}(\boldsymbol{v})$ given by Lemma C.4, using Lemma C.3 and $\psi \ge 0$, followed by Lemma C.6 we find

$$\begin{aligned}
\delta_t^{\gamma\eta}(\boldsymbol{w}^{\gamma\eta}) &\le \frac{1}{\gamma\eta} \sum_k w_t^{\gamma\eta,k} \psi\big(\gamma\eta(m_t^\eta(\boldsymbol{w}^\eta) - \ell_t^k)\big) \\
&\le K^{\gamma-1} \frac{1}{\gamma\eta} \sum_k w_t^{\eta,k} \psi\big(\gamma\eta(m_t^\eta(\boldsymbol{w}^\eta) - \ell_t^k)\big) \\
&\le K^{\gamma-1} \frac{1}{\gamma\eta} \sum_k w_t^{\eta,k} \psi\big(\eta(m_t^\eta(\boldsymbol{w}^\eta) - \ell_t^k)\big) \gamma^2 e^{(\gamma-1)\eta} \\
&= K^{\gamma-1} e^{(\gamma-1)\eta} \gamma \delta_t^\eta(\boldsymbol{w}^\eta).
\end{aligned}$$

□

C.4 Proof of Lemma 3.1

Suppose that after round T LLR has increased its budget d times. For $j = 1, \dots, d$, let v_j be the last round before the j-th increase of the budget, and also define $v_0 = 0$ and $v_{d+1} = T$ for convenience. For $j = 1, \dots, d+1$, let $M_{[j]} = \sum_{t=v_{j-1}+1}^{v_j} m_t$ be the cumulative mix loss during the j-th value of the budget. By construction, the learning rate η_t chosen by LLR is non-increasing from round $v_{j-1}+1$ to round v_j. Consequently, as the cumulative mix loss M_t^η for the first t rounds is non-increasing in η (see Lemma C.1),

$$\begin{aligned} M_{[j]} &= m_{v_{j-1}+1} + \sum_{t=v_{j-1}+2}^{v_j} M_t^{\eta_t} - M_{t-1}^{\eta_t} \leq m_{v_{j-1}+1} + \sum_{t=v_{j-1}+2}^{v_j} M_t^{\eta_t} - M_{t-1}^{\eta_{t-1}} \\ &= m_{v_{j-1}+1} + M_{v_j}^{\eta_{v_j}} - M_{v_{j-1}+1}^{\eta_{v_{j-1}+1}} = M_{v_j}^{\eta_{v_j}} - M_{v_{j-1}}^{\eta_{v_{j-1}+1}} \leq M_{v_j}^{\eta^{\text{ah}}_{v_j}} - M_{v_{j-1}}^{\eta_{v_{j-1}+1}}. \quad (18) \end{aligned}$$

For $j = 1$, $M_{v_{j-1}}^{\eta_{v_{j-1}+1}} = M_0^{\eta_{v_{j-1}+1}} = 0 = M_{v_{j-1}}^{\eta^{\text{ah}}_{v_{j-1}}}$; for $j = 2, \dots, d+1$, property 3 of Lemma C.1 implies

$$M_{v_{j-1}}^{\eta_{v_{j-1}+1}} \geq L^*_{v_{j-1}} \geq M_{v_{j-1}}^{\eta^{\text{ah}}_{v_{j-1}}} - \frac{\ln K}{\eta^{\text{ah}}_{v_{j-1}}} = M_{v_{j-1}}^{\eta^{\text{ah}}_{v_{j-1}}} - \Delta^{\text{ah}}_{v_{j-1}-1} \geq M_{v_{j-1}}^{\eta^{\text{ah}}_{v_{j-1}}} - \Delta^{\text{ah}}_{v_{j-1}}.$$

Combining this with (18), we get

$$\begin{aligned} M_T &= \sum_{j=1}^{d+1} M_{[j]} \leq \sum_{j=1}^{d+1} \Big(M_{v_j}^{\eta^{\text{ah}}_{v_j}} - M_{v_{j-1}}^{\eta_{v_{j-1}+1}} \Big) \leq \sum_{j=1}^{d+1} \Big(M_{v_j}^{\eta^{\text{ah}}_{v_j}} - M_{v_{j-1}}^{\eta^{\text{ah}}_{v_{j-1}}} \Big) + \sum_{j=2}^{d+1} \Delta^{\text{ah}}_{v_{j-1}} \\ &= M_T^{\eta^{\text{ah}}_T} + \sum_{j=1}^{d} \Delta^{\text{ah}}_{v_j} \overset{(\dagger)}{\leq} L^*_T + \Delta^{\text{ah}}_{T-1} + \sum_{j=1}^{d} \Delta^{\text{ah}}_{v_j} \leq L^*_T + \Delta^{\text{ah}}_T + \sum_{j=1}^{d} \Delta^{\text{ah}}_{v_j}. \end{aligned}$$

where inequality $(\dagger)$ follows from property 3 of Lemma C.1 and the definition (10) of η^{ah}_T. Because the budget has been exceeded d times, we know that

$$\Delta^{\text{ah}}_{v_d} \geq \phi \Delta^{\text{ah}}_{v_{d-1}} \geq \phi^2 \Delta^{\text{ah}}_{v_{d-2}} \geq \ldots \geq \phi^{d-1} \Delta^{\text{ah}}_{v_1},$$

so that

$$\sum_{j=1}^{d} \Delta^{\text{ah}}_{v_j} \leq \sum_{j=1}^{d} \phi^{j-d} \Delta^{\text{ah}}_{v_d} = \Delta^{\text{ah}}_{v_d} \sum_{j=0}^{d-1} \phi^{-j} \leq \Delta^{\text{ah}}_{v_d} \sum_{j=0}^{\infty} \phi^{-j} = \Delta^{\text{ah}}_{v_d} \frac{\phi}{\phi - 1} \leq \Delta^{\text{ah}}_T \frac{\phi}{\phi - 1}.$$

We can now decompose the regret of LLR as

$$\begin{aligned} \mathcal{R}_T = M_T - L^*_T + \Delta_T \leq \Delta^{\text{ah}}_T + \sum_{j=1}^{d} \Delta^{\text{ah}}_{v_j} + \Delta_T &\leq \Big(\frac{\phi}{\phi - 1} + 1 \Big) \Delta^{\text{ah}}_T + \Delta_T \\ &= \Big(\frac{\phi}{\phi - 1} + 2 \Big) \Delta^{\text{ah}}_T + \sum_{i=1}^{i_{\max}} \Delta^i_T, \end{aligned}$$

as required.

C.5 Proof of Lemma 3.2

The value of the budget after T rounds is b_T. Assume first that $b_T > 0$. Let u be the round just before the budget was last increased, i.e. u is the last round such that $b_u < b_T$. At time t^*, we know $b_{t^*} \geq b_u$ because AdaHedge was played at least once while the budget was b_u to cause its increase. Since i was active at time t^* but AdaHedge was played, i must have been full, i.e. $\Delta^i_{t^*}/\pi^i > b_{t^*}$. Hence

$$b_u \leq b_{t^*} < \Delta^i_{t^*}/\pi^i \leq \Delta^i_T/\pi^i. \quad (19)$$

By definition of the LLR budgeting, $\Delta_t^j/\pi^j \leq b_t + \delta_t^j/\pi^j$ and similarly $\Delta_t^{\text{ah}}/\pi^{\text{ah}} \leq b_t + \delta_t^{\text{ah}}/\pi^{\text{ah}}$ at any time t. By Lemma C.1 $\delta_t^\eta \leq 1$, and by Hoeffding's bound on the cumulant generating function [4, Lemma A.1] $\delta_t^\eta \leq \eta/8$ regardless of the choice of η. Hence

$$\begin{aligned}\frac{\Delta_T^j}{\pi^j} &\leq b_T + \frac{\min\{1,\eta^j/8\}}{\pi^j} = \phi\Delta_u^{\text{ah}}/\pi^{\text{ah}} + \frac{\min\{1,\eta^j/8\}}{\pi^j}\\ &\leq \phi\left(b_u + \frac{1}{\pi^{\text{ah}}}\right) + \frac{\min\{1,\eta^j/8\}}{\pi^j} \leq \phi\left(\frac{\Delta_T^i}{\pi^i} + \frac{1}{\pi^{\text{ah}}}\right) + \frac{\min\{1,\eta^j/8\}}{\pi^j},\end{aligned}$$

where the last inequality follows by (19). Similarly, $b_T = \phi\Delta_u^{\text{ah}}/\pi^{\text{ah}} \leq \phi\Delta_T^{\text{ah}}/\pi^{\text{ah}}$ implies

$$\frac{\Delta_T^j}{\pi^j} \leq b_T + \frac{\min\{1,\eta^j/8\}}{\pi^j} \leq \phi\frac{\Delta_T^{\text{ah}}}{\pi^{\text{ah}}} + \frac{\min\{1,\eta^j/8\}}{\pi^j}. \tag{20}$$

For the last bound we use that AdaHedge is played by LLR only after all active i are full (i.e. have exhausted the current budget).

$$\frac{\Delta_T^{\text{ah}}}{\pi^{\text{ah}}} = \frac{\Delta_{t^*}^{\text{ah}}}{\pi^{\text{ah}}} \leq b_{t^*} + \frac{\delta_{t^*}^{\text{ah}}}{\pi^{\text{ah}}} < \frac{\Delta_{t^*}^i}{\pi^i} + \frac{\delta_{t^*}^{\text{ah}}}{\pi^{\text{ah}}} \leq \frac{\Delta_T^i}{\pi^i} + \frac{1}{\pi^{\text{ah}}}.$$

If $b_T = 0$ then $\eta_{t^*}^{\text{ah}} = \infty$ and hence no i is active at time t^*, so we only need to prove (16b). Since $b_T = 0 \leq \phi\Delta_T^{\text{ah}}/\pi^{\text{ah}}$ this again follows by (20).

C.6 Proof of Lemma 3.4

Let i be an arbitrary index that is active at time t^*. Then in Lemma 3.1 we bound Δ_T^{ah} and Δ_T^j for $j \neq i$ in terms of Δ_T^i using Lemma 3.2, which gives

$$\begin{aligned}\mathcal{R}_T &\leq \Big(\frac{\phi}{\phi-1}+2\Big)\Big(\frac{\pi^{\text{ah}}}{\pi^i}\Delta_T^i + 1\Big) + \sum_{j\leq i_{\max};j\neq i}\left(\phi\left(\frac{\pi^j}{\pi^i}\Delta_T^i + \frac{\pi^j}{\pi^{\text{ah}}}\right) + \min\{1,\eta^j/8\}\right) + \Delta_T^i\\ &= \left(\Big(\frac{\phi}{\phi-1}+2\Big)\frac{\pi^{\text{ah}}}{\pi^i} + \phi\frac{\sum_{j\leq i_{\max};j\neq i}\pi^j}{\pi^i} + 1\right)\Delta_T^i\\ &\quad + \left(\frac{\phi}{\phi-1} + 2 + \phi\frac{\sum_{j\leq i_{\max};j\neq i}\pi^j}{\pi^{\text{ah}}} + \sum_{j\leq i_{\max};j\neq i}\min\{1,\eta^j/8\}\right).\end{aligned} \tag{21}$$

By definition Δ_T^i accumulates δ_t^i only in rounds t where LLR plays η^i. Since $\delta_t^i \geq 0$ (see Lemma C.1) we can extend the sum to all trials:

$$\Delta_T^i \leq \Delta_T^{\eta^i}. \tag{22}$$

For the sums over π^j, we have

$$\phi\frac{\sum_{j\leq i_{\max};j\neq i}\pi^j}{\pi^i} + 1 \leq \phi\frac{\sum_{j\leq i_{\max}}\pi^j}{\pi^i} \leq \frac{\phi}{\pi^i} \quad\text{and}\quad \phi\frac{\sum_{j\leq i_{\max};j\neq i}\pi^j}{\pi^{\text{ah}}} \leq \frac{\phi}{\pi^{\text{ah}}}. \tag{23}$$

We proceed to bound the sum $\sum_j \min\{1,\eta^j/8\}$, which is at most a constant by the definition of the grid (8). For $j = 1$ the minimum is 1 since $\eta^1 = \infty$, and for $j \geq 2$ we bound the minimum by $\eta^j/8$, which leads to

$$\sum_j \min\{1,\eta^j/8\} \leq 1 + \frac{1}{8}\sum_{j=2}^\infty \alpha^{2-j} = 1 + \frac{\alpha}{8(\alpha-1)}. \tag{24}$$

Plugging (22), (23) and (24) into (21) for $i = 1$ and using $\pi^1 = \pi^\infty$ $\big($see (12)$\big)$, we obtain the second inequality of the lemma.

Now let $\eta \in [\eta_{t^*}^{\text{ah}}, 1]$ be arbitrary. Then $i \equiv i(\eta)$ is active at time t^* and $\eta \leq \eta^i \leq \alpha\eta$, so that by Lemma 3.3 we have

$$\Delta_T^{\eta^i} \leq \alpha e^{(\alpha-1)(\ln K+\eta)}\Delta_T^\eta \leq \alpha e^{(\alpha-1)(\ln K+1)}\Delta_T^\eta. \tag{25}$$

Plugging (22), (23), (24) and (25) into (21) for $i = i(\eta)$ establishes the first inequality of the lemma.

Lemma 3.1 combined with (16b) results in

$$\mathcal{R}_T \leq \Big(\frac{\phi}{\phi - 1} + 2\Big)\Delta_T^{\text{ah}} + \phi\frac{\sum_{j=1}^{i_{\max}} \pi^j}{\pi^{\text{ah}}}\Delta_T^{\text{ah}} + \sum_{j \leq i_{\max}} \min\{1, \eta^j/8\},$$

and the third inequality of the theorem follows by $\sum_{i=1}^{i_{\max}} \pi^i \leq 1$ and (24).

Finally, suppose that $\eta < \eta_{t^*}^{\text{ah}}$. Then, since t^* is the last round in which AdaHedge was used and $\delta_{t^*}^{\text{ah}} \leq 1$ by Lemma C.1,

$$\Delta_T^{\text{ah}} = \Delta_{t^*-1}^{\text{ah}} + \delta_{t^*}^{\text{ah}} \leq \Delta_{t^*-1}^{\text{ah}} + 1 = \frac{\ln K}{\eta_{t^*}^{\text{ah}}} + 1 \leq \frac{\ln K}{\eta} + 1,$$

which proves the last inequality of the lemma.

C.7 Proof of Theorem 3.5

We continue from Lemma 3.4, and start by bounding $1/\pi^{i(\eta)}$ from above. To this end, we first observe that

$$i(\eta) \leq 2 + \log_\alpha(1/\eta) = 2 + \frac{\ln(1/\eta)}{\ln(1 + 1/\log_2 K)} \leq 2 + \big(\log_2 K + 1\big)\ln(1/\eta),$$

where the second inequality follows from $\ln(1+x) \geq x/(1+x)$. Next we lower bound the heavy-tailed prior ρ. We bound its defining integral (13) by the width times the lowest integrand to find

$$\rho(i) \geq \frac{1}{\ln K\left(\frac{i}{\ln K} + e\right)\ln^2\left(\frac{i}{\ln K} + e\right)}.$$

Hence by the definition of the grid-point weights (12)

$$\begin{aligned}\frac{1 - \pi^\infty}{\pi^{i(\eta)}} &\leq (i(\eta) - 1 + e\ln K)\ln^2\left(\frac{i(\eta) - 1}{\ln K} + e\right)\\ &\leq \big((\log_2 K + 1)\ln(1/\eta) + e\ln K + 1\big)\ln^2\left(\frac{1 + (\log_2 K + 1)\ln(1/\eta)}{\ln K} + e\right).\end{aligned}$$

The first factor is at most

$$(\log_2 K + 1)\ln(1/\eta) + e\ln K + 1 \leq (\log_2 K + 1)\ln(7/\eta),$$

and we use that $K \geq 2$, so that

$$\begin{aligned}\frac{1 + (\log_2 K + 1)\ln(1/\eta)}{\ln K} + e &\leq \left(\frac{1}{\ln 2} + \frac{1}{\ln K}\right)\ln(1/\eta) + \frac{1}{\ln K} + e\\ &\leq \left(\frac{1}{\ln 2} + \frac{1}{\ln 2}\right)\ln(1/\eta) + \frac{1}{\ln 2} + e \leq 2\log_2(5/\eta).\end{aligned}$$

Thus

$$\frac{1}{\pi^{i(\eta)}} \leq \frac{\log_2 K + 1}{1 - \pi^\infty}\ln(7/\eta)\ln^2\left(2\log_2(5/\eta)\right). \tag{26}$$

Next, we use the definition of α (9) and $K \geq 2$ to bound

$$\alpha e^{(\alpha - 1)(\ln K + 1)} = (1 + 1/\log_2 K)e^{\ln 2 + 1/\log_2 K} \leq 4e. \tag{27}$$

The first inequality of the theorem now follows by applying Lemma 2.1, plugging (26), (27) and (7) into Lemma 3.4, and evaluating $\frac{\alpha}{\alpha - 1} = \log_2 K + 1$. The second inequality follows directly by plugging in (7). And, finally, the third inequality follows by combining the last two inequalities of Lemma 3.4.

C.8 Proof of Theorem 3.6

Let $V_T^{\text{ah}} = \sum_{s \in \mathcal{A}_T^{\text{ah}}} v_t$ be the sum of variances v_t over all times $t \leq T$ that LLR plays AdaHedge. By the same argument as in the proofs of Lemma 5 and Theorem 6 in [5]

$$\Delta_T^{\text{ah}} \leq \sqrt{V_T^{\text{ah}} \ln K} + \left(\tfrac{2}{3} \ln K + 1\right). \tag{28}$$

Plugging this bound into the third inequality of Lemma 3.4, bounding $V_T^{\text{ah}} \leq V_T$ and evaluating $\frac{\alpha}{\alpha-1} = \log_2 K + 1$ and $\phi = 1 + \sqrt{\pi^{\text{ah}}}$, we obtain the first inequality of the theorem. The second inequality follows from the first by the same argument as in the proof of Corollary 3 of [7].