Journal of Machine Learning Research 15 (2014) 1281-1316 Submitted 1/13; Revised 1/14; Published 4/14

Follow the Leader If You Can, Hedge If You Must

Steven de Rooij STEVEN.DE.ROOIJ@GMAIL.COM
VU University and University of Amsterdam
Science Park 904, P.O. Box 94323, 1090 GH Amsterdam, the Netherlands

Tim van Erven TIM@TIMVANERVEN.NL
Département de Mathématiques
Université Paris-Sud, 91405 Orsay Cedex, France

Peter D. Grünwald PDG@CWI.NL
Wouter M. Koolen WMKOOLEN@CWI.NL
Leiden University (Grünwald) and Centrum Wiskunde & Informatica (Grünwald and Koolen)
Science Park 123, P.O. Box 94079, 1090 GB Amsterdam, the Netherlands

Editor: Nicolò Cesa-Bianchi

Abstract

Follow-the-Leader (FTL) is an intuitive sequential prediction strategy that guarantees constant regret in the stochastic setting, but has poor performance for worst-case data. Other hedging strategies have better worst-case guarantees but may perform much worse than FTL if the data are not maximally adversarial. We introduce the FlipFlop algorithm, which is the first method that provably combines the best of both worlds. As a stepping stone for our analysis, we develop AdaHedge, which is a new way of dynamically tuning the learning rate in Hedge without using the doubling trick. AdaHedge refines a method by Cesa-Bianchi, Mansour, and Stoltz (2007), yielding improved worst-case guarantees. By interleaving AdaHedge and FTL, FlipFlop achieves regret within a constant factor of the FTL regret, without sacrificing AdaHedge's worst-case guarantees. AdaHedge and FlipFlop do not need to know the range of the losses in advance; moreover, unlike earlier methods, both have the intuitive property that the issued weights are invariant under rescaling and translation of the losses. The losses are also allowed to be negative, in which case they may be interpreted as gains.

Keywords: Hedge, learning rate, mixability, online learning, prediction with expert advice

1. Introduction

We consider sequential prediction in the general framework of Decision Theoretic Online Learning (DTOL) or the *Hedge setting* (Freund and Schapire, 1997), which is a variant of *prediction with expert advice* (Littlestone and Warmuth, 1994; Vovk, 1998; Cesa-Bianchi and Lugosi, 2006). Our goal is to develop a sequential prediction algorithm that performs well not only on adversarial data, which is the scenario most studies worry about, but also when the data are easy, as is often the case in practice. Specifically, with adversarial data, the worst-case regret (defined below) for any algorithm is $\Omega(\sqrt{T})$, where T is the number of predictions to be made. Algorithms such as Hedge, which have been designed to achieve this lower bound, typically continue to suffer regret of order $\sqrt{T}$, even for easy data, where the regret of the more intuitive but less robust Follow-the-Leader (FTL) algorithm (also defined below) is *bounded*. Here, we present the first algorithm which, up to constant factors, provably achieves both the regret lower bound in the worst case, *and* a regret not exceeding that of FTL. Below, we first describe the Hedge setting. Then we introduce FTL, discuss sophisticated versions of Hedge from the literature, and give an overview of the results and contents of this paper.

1.1 Overview

In the Hedge setting, prediction proceeds in rounds. At the start of each round $t = 1, 2, \ldots$, a learner has to decide on a weight vector $\boldsymbol{w}_t = (w_{t,1}, \ldots, w_{t,K}) \in \mathbb{R}^K$ over K "experts". Each weight $w_{t,k}$ is required to be nonnegative, and the sum of the weights should be 1. Nature then reveals a K-dimensional vector containing the losses of the experts $\boldsymbol{\ell}_t = (\ell_{t,1}, \ldots, \ell_{t,K}) \in \mathbb{R}^K$. Learner's loss is the dot product $h_t = \boldsymbol{w}_t \cdot \boldsymbol{\ell}_t$, which can be interpreted as the expected loss if Learner uses a mixed strategy and chooses expert k with probability $w_{t,k}$. We denote aggregates of per-trial quantities by their capital letter, and vectors are in bold face. Thus, $L_{t,k} = \ell_{1,k} + \ldots + \ell_{t,k}$ denotes the cumulative loss of expert k after t rounds, and $H_t = h_1 + \ldots + h_t$ is Learner's cumulative loss (the *Hedge loss*).

Learner's performance is evaluated in terms of her *regret*, which is the difference between her cumulative loss and the cumulative loss of the best expert:

$$\mathcal{R}_t = H_t - L_t^*, \qquad \text{where } L_t^* = \min_k L_{t,k}.$$

We will always analyse the regret after an arbitrary number of rounds T. We will omit the subscript T for aggregate quantities such as L_T^* or $\mathcal{R}_T$ wherever this does not cause confusion.

A simple and intuitive strategy for the Hedge setting is Follow-the-Leader (FTL), which puts all weight on the expert(s) with the smallest loss so far. More precisely, we will define the weights $\boldsymbol{w}_t$ for FTL to be uniform on the set of leaders $\{k \mid L_{t-1,k} = L_{t-1}^*\}$, which is often just a singleton. FTL works very well in many circumstances, for example in stochastic scenarios where the losses are independent and identically distributed (i.i.d.). In particular, the regret for Follow-the-Leader is bounded by the number of times the leader is overtaken by another expert (Lemma 10), which in the i.i.d. case almost surely happens only a finite number of times (by the uniform law of large numbers), provided the mean loss of the best expert is strictly smaller than the mean loss of the other experts. As demonstrated by the experiments in Section 5, many more sophisticated algorithms can perform significantly worse than FTL.

The problem with FTL is that it breaks down badly when the data are antagonistic. For example, if one out of two experts incurs losses $\frac{1}{2}, 0, 1, 0, \ldots$ while the other incurs opposite losses $0, 1, 0, 1, \ldots$, the regret for FTL at time T is about $T/2$ (this scenario is further discussed in Section 5.1). This has prompted the development of a multitude of alternative algorithms that provide better worst-case regret guarantees.

The seminal strategy for the learner is called *Hedge* (Freund and Schapire, 1997, 1999). Its performance crucially depends on a parameter η called the *learning rate*. Hedge can be interpreted as a generalisation of FTL, which is recovered in the limit for $\eta \to \infty$. In many analyses, the learning rate is changed from infinity to a lower value that optimizes some upper bound on the regret. Doing so requires precognition of the number of rounds of the game, or of some property of the data such as the eventual loss of the best expert L^*. Provided that the relevant statistic is monotonically nondecreasing in t (such as L_t^*), a simple way to address this issue is the so-called *doubling trick*: setting a budget on the statistic, and restarting the algorithm with a double budget when the budget is depleted (Cesa-Bianchi and Lugosi, 2006; Cesa-Bianchi et al., 1997; Hazan and Kale, 2008); η can then be optimised for each individual block in terms of the budget. Better bounds, but harder analyses, are typically obtained if the learning rate is adjusted each round based on previous observations, see e.g. (Cesa-Bianchi and Lugosi, 2006; Auer et al., 2002).

The Hedge strategy presented by Cesa-Bianchi, Mansour, and Stoltz (2007) is a sophisticated example of such adaptive tuning. The relevant algorithm, which we refer to as CBMS, is defined in (16) in Section 4.2 of their paper. To discuss its guarantees, we need the following notation. Let $\ell_t^- = \min_k \ell_{t,k}$ and $\ell_t^+ = \max_k \ell_{t,k}$ denote the smallest and largest loss in round t, and let $L_t^- = \ell_1^- + \ldots + \ell_t^-$ and $L_t^+ = \ell_1^+ + \ldots + \ell_t^+$ denote the cumulative minimum and maximum loss respectively. Further let $s_t = \ell_t^+ - \ell_t^-$ denote the loss range in trial t and let $S_t = \max\{s_1, \ldots, s_t\}$ denote the largest loss range after t trials. Then, *without prior knowledge* of any property of the data, including T, S and L^*, the CBMS strategy achieves regret bounded by[1]

$$\mathcal{R}^{\text{CBMS}} \leq 4\sqrt{\frac{(L^* - L^-)(L^- + ST - L^*)}{T} \ln K} + \text{lower order terms} \tag{1}$$

(Cesa-Bianchi et al., 2007, Corollary 3). Hence, in the worst case $L^* = L^- + ST/2$ and the bound is of order $S\sqrt{T}$, but when the loss of the best expert $L^* \in [L^-, L^- + ST]$ is close to either boundary the guarantees are much stronger.

The contributions of this work are twofold: first, in Section 2, we develop AdaHedge, which is a refinement of the CBMS strategy. A (very) preliminary version of this strategy was presented at NIPS (Van Erven et al., 2011). Like CMBS, AdaHedge is completely parameterless and tunes the learning rate in terms of a direct measure of past performance. We derive an improved worst-case bound of the following form. Again without any assumptions, we have

$$\mathcal{R}^{\text{ah}} \leq 2\sqrt{S\frac{(L^* - L^-)(L^+ - L^*)}{L^+ - L^-} \ln K} + \text{lower order terms} \tag{2}$$

(see Theorem 8). The parabola under the square root is always smaller than or equal to its CMBS counterpart (since it is nondecreasing in L^+ and $L^+ \leq L^- + ST$); it expresses that the regret is small if $L^* \in [L^-, L^+]$ is close to either boundary. It is maximized in L^* at the midpoint between L^- and L^+, and in this case we recover the worst-case bound of order $S\sqrt{T}$. Like (1), the regret bound (2) is "fundamental", which means that it is invariant under translation of the losses and proportional to their scale. Moreover, not only AdaHedge's regret *bound* is fundamental: the weights issued by the algorithm are themselves invariant

1. As pointed out by a referee, it is widely known that the leading constant of 4 can be improved to $2\sqrt{2} \approx 2.83$ using techniques by Györfi and Ottucsák (2007) that are essentially equivalent to our Lemma 2 below; Gerchinovitz (2011, Remark 2.2) reduced it to approximately 2.63. AdaHedge allows a slight further reduction to 2.

under translation and scaling (see Section 4). The CBMS algorithm and AdaHedge are insensitive to trials in which all experts suffer the same loss, a natural property we call "timelessness". An attractive feature of the new bound (2) is that it expresses this property. A more detailed discussion appears below Theorem 8.

Our second contribution is to develop a second algorithm, called FlipFlop, that retains the worst-case bound (2) (up to a constant factor), but has even better guarantees for easy data: its performance is never substantially worse than that of Follow-the-Leader. At first glance, this may seem trivial to accomplish: simply take both FTL and AdaHedge, and combine the two by using FTL or Hedge recursively. To see why such approaches do not work, suppose that FTL achieves regret $\mathcal{R}^{\text{ftl}}$, while AdaHedge achieves regret $\mathcal{R}^{\text{ah}}$. We would only be able to prove that the regret of the combined strategy compared to the best original expert satisfies $\mathcal{R}^{\text{c}} \leq \min\{\mathcal{R}^{\text{ftl}}, \mathcal{R}^{\text{ah}}\} + \mathcal{G}^{\text{c}}$, where $\mathcal{G}^{\text{c}}$ is the worst-case regret guarantee for the combination method, e.g. (1). In general, either $\mathcal{R}^{\text{ftl}}$ or $\mathcal{R}^{\text{ah}}$ may be close to zero, while at the same time the regret of the combination method, or at least its bound $\mathcal{G}^{\text{c}}$, is proportional to $\sqrt{T}$. That is, the overhead of the combination method will dominate the regret!

The FlipFlop approach we describe in Section 3 circumvents this by alternating between Following the Leader and using AdaHedge in a carefully specified way. For this strategy we can guarantee

$$\mathcal{R}^{\text{ff}} = O(\min\{\mathcal{R}^{\text{ftl}}, \mathcal{G}^{\text{ah}}\}),$$

where $\mathcal{G}^{\text{ah}}$ is the regret guarantee for AdaHedge; Theorem 15 provides a precise statement. Thus, FlipFlop is the first algorithm that provably combines the benefits of Follow-the-Leader with robust behaviour for antagonistic data.

A key concept in the design and analysis of our algorithms is what we call the *mixability gap*, introduced in Section 2.1. This quantity also appears in earlier works, and seems to be of fundamental importance in both the current Hedge setting as well as in stochastic settings. We elaborate on this in Section 6.2 where we provide the big picture underlying this research and we briefly indicate how it relates to practical work such as (Devaine et al., 2013).

1.2 Related Work

As mentioned, AdaHedge is a refinement of the strategy analysed by Cesa-Bianchi et al. (2007), which is itself more sophisticated than most earlier approaches, with two notable exceptions. First, Chaudhuri, Freund, and Hsu (2009) describe a strategy called NormalHedge that can efficiently compete with the best ϵ-quantile of experts; their bound is incomparable with the bounds for CBMS and for AdaHedge. Second, Hazan and Kale (2008) develop a strategy called Variation MW that has especially low regret when the losses of the best expert vary little between rounds. They show that the regret of Variation MW is of order $\sqrt{\text{VAR}_T^{\max} \ln K}$, where $\text{VAR}_T^{\max} = \max_{t \leq T} \sum_{s=1}^{t} \left(\ell_{s,k_t^*} - \frac{1}{t} L_{t,k_t^*}\right)^2$ with k_t^* the best expert after t rounds. This bound dominates our worst-case result (2) (up to a multiplicative constant). As demonstrated by the experiments in Section 5, their method does not achieve the benefits of FTL, however. In Section 5 we also discuss the performance of NormalHedge and Variation MW compared to AdaHedge and FlipFlop.

Other approaches to sequential prediction include Defensive Forecasting (Vovk et al., 2005), and Following the Perturbed Leader (Kalai and Vempala, 2003). These radically different approaches also allow competing with the best ϵ-quantile, as shown by Chernov and Vovk (2010) and Hutter and Poland (2005); the latter also consider nonuniform weights on the experts.

The "safe MDL" and "safe Bayesian" algorithms by Grünwald (2011, 2012) share the present work's focus on the mixability gap as a crucial part of the analysis, but are concerned with the stochastic setting where losses are not adversarial but i.i.d. FlipFlop, safe MDL and safe Bayes can all be interpreted as methods that attempt to choose a learning rate η that keeps the mixability gap small (or, equivalently, that keeps the Bayesian posterior or Hedge weights "concentrated").

1.3 Outline

In the next section we present and analyse AdaHedge and compare its worst-case regret bound to existing results, in particular the bound for CBMS. Then, in Section 3, we build on AdaHedge to develop the FlipFlop strategy. The analysis closely parallels that of AdaHedge, but with extra complications at each of the steps. In Section 4 we show that both algorithms have the property that their behaviour does not change under translation and scaling of the losses. We further illustrate the relationship between the learning rate and the regret, and compare AdaHedge and FlipFlop to existing methods, in experiments with artificial data in Section 5. Finally, Section 6 contains a discussion, with ambitious suggestions for future work.

2. AdaHedge

In this section, we present and analyse the AdaHedge strategy. To introduce our notation and proof strategy, we start with the simplest possible analysis of vanilla Hedge, and then move on to refine it for AdaHedge.

2.1 Basic Hedge Analysis for Constant Learning Rate

Following Freund and Schapire (1997), we define the *Hedge* or *exponential weights* strategy as the choice of weights

$$w_{t,k} = \frac{w_{1,k} e^{-\eta L_{t-1,k}}}{Z_t}, \tag{3}$$

where $\boldsymbol{w}_1 = (1/K, \dots, 1/K)$ is the uniform distribution, $Z_t = \boldsymbol{w}_1 \cdot e^{-\eta \boldsymbol{L}_{t-1}}$ is a normalizing constant, and $\eta \in (0, \infty)$ is a parameter of the algorithm called the *learning rate*. If $\eta = 1$ and one imagines $L_{t-1,k}$ to be the negative log-likelihood of a sequence of observations, then $w_{t,k}$ is the Bayesian posterior probability of expert k and Z_t is the marginal likelihood of the observations. Like in Bayesian inference, the weights are updated multiplicatively, i.e. $w_{t+1,k} \propto w_{t,k} e^{-\eta \ell_{t,k}}$.

The loss incurred by Hedge in round t is $h_t = \boldsymbol{w}_t \cdot \boldsymbol{\ell}_t$, the cumulative Hedge loss is $H_t = h_1 + \ldots + h_t$, and our goal is to obtain a good bound on H_T. To this end, it turns out to be technically convenient to approximate h_t by the *mix loss*

$$m_t = -\frac{1}{\eta} \ln(\boldsymbol{w}_t \cdot e^{-\eta \boldsymbol{\ell}_t}), \tag{4}$$

which accumulates to $M_t = m_1 + \ldots + m_t$. This approximation is a standard tool in the literature. For example, the mix loss m_t corresponds to the loss of Vovk's (1998; 2001) Aggregating Pseudo Algorithm, and tracking the evolution of $-m_t$ is a crucial ingredient in the proof of Theorem 2.2 of Cesa-Bianchi and Lugosi (2006).

The definitions may be extended to $\eta = \infty$ by letting η tend to ∞. We then find that $\boldsymbol{w}_t$ becomes a uniform distribution on the set of experts $\{k \mid L_{t-1,k} = L^*_{t-1}\}$ that have incurred smallest cumulative loss before time t. That is, Hedge with $\eta = \infty$ reduces to *Follow-the-Leader*, where in case of ties the weights are distributed uniformly. The limiting value for the mix loss is $m_t = L^*_t - L^*_{t-1}$.

In our approximation of the Hedge loss h_t by the mix loss m_t, we call the approximation error $\delta_t = h_t - m_t$ the *mixability gap*. Bounding this quantity is a standard part of the analysis of Hedge-type algorithms (see, for example, Lemma 4 of Cesa-Bianchi et al. 2007) and it also appears to be a fundamental notion in sequential prediction even when only so-called mixable losses are considered (Grünwald, 2011, 2012); see also Section 6.2. We let $\Delta_t = \delta_1 + \ldots + \delta_t$ denote the cumulative mixability gap, so that the regret for Hedge may be decomposed as

$$\mathcal{R} = H - L^* = M - L^* + \Delta. \tag{5}$$

Here $M - L^*$ may be thought of as the regret under the mix loss and Δ is the cumulative approximation error when approximating the Hedge loss by the mix loss. Throughout the paper, our proof strategy will be to analyse these two contributions to the regret, $M - L^*$ and Δ, separately.

The following lemma, which is proved in Appendix A, collects a few basic properties of the mix loss:

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

1. $\ell^-_t \leq m_t \leq h_t \leq \ell^+_t$, *so that* $0 \leq \delta_t \leq s_t$.
2. *Cumulative mix loss telescopes:* $M = \begin{cases} -\frac{1}{\eta} \ln\left(\boldsymbol{w}_1 \cdot e^{-\eta \boldsymbol{L}}\right) & \textit{for } \eta < \infty, \\ L^* & \textit{for } \eta = \infty. \end{cases}$
3. *Cumulative mix loss approximates the loss of the best expert:* $L^* \leq M \leq L^* + \dfrac{\ln K}{\eta}$.
4. *The cumulative mix loss* M *is nonincreasing in* η.

In order to obtain a bound for Hedge, one can use the following well-known bound on the mixability gap, which is obtained using Hoeffding's bound on the cumulant generating function (Cesa-Bianchi and Lugosi, 2006, Lemma A.1):

$$\delta_t \leq \frac{\eta}{8} s_t^2, \tag{6}$$

from which $\Delta \leq S^2 T\eta/8$, where (as in the introduction) $S_t = \max\{s_1, \ldots, s_t\}$ is the maximum loss range in the first t rounds. Together with the bound $M - L^* \leq \ln(K)/\eta$ from mix loss property #3 this leads to

$$\mathcal{R} = (M - L^*) + \Delta \leq \frac{\ln K}{\eta} + \frac{\eta S^2 T}{8}. \tag{7}$$

The bound is optimized for $\eta = \sqrt{8\ln(K)/(S^2T)}$, which equalizes the two terms. This leads to a bound on the regret of $S\sqrt{T\ln(K)/2}$, matching the lower bound on worst-case regret from the textbook by Cesa-Bianchi and Lugosi (2006, Section 3.7). We can use this tuned learning rate if the time horizon T is known in advance. To deal with the situation where T is unknown, either the doubling trick or a time-varying learning rate (see Lemma 2 below) can be used, at the cost of a worse constant factor in the leading term of the regret bound.

In the remainder of this section, we introduce a completely parameterless algorithm called AdaHedge. We then refine the steps of the analysis above to obtain a better regret bound.

2.2 AdaHedge Analysis

In the previous section, we split the regret for Hedge into two parts: $M - L^*$ and Δ, and we obtained a bound for both. The learning rate η was then tuned to equalise these two bounds. The main distinction between AdaHedge and other Hedge approaches is that AdaHedge does not consider an upper bound on Δ in order to obtain this balance: instead it aims to equalize Δ and $\ln(K)/\eta$. As the cumulative mixability gap Δ_t is nondecreasing in t (by mix loss property #1) and can be *observed* on-line, it is possible to adapt the learning rate directly based on Δ_t.

Perhaps the easiest way to achieve this is by using the doubling trick: each subsequent block uses half the learning rate of the previous block, and a new block is started as soon as the observed cumulative mixability gap Δ_t exceeds the bound on the mix loss $\ln(K)/\eta$, which ensures these two quantities are equal at the end of each block. This is the approach taken in an earlier version of AdaHedge (Van Erven et al., 2011). However, we can achieve the same goal much more elegantly, by decreasing the learning rate with time according to

$$\eta_t^{\text{ah}} = \frac{\ln K}{\Delta_{t-1}^{\text{ah}}} \tag{8}$$

(where $\Delta_0^{\text{ah}} = 0$, so that $\eta_1^{\text{ah}} = \infty$). Note that the AdaHedge learning rate does not involve the end time T or any other unobserved properties of the data; all subsequent analysis is therefore valid for all T simultaneously. The definitions (3) and (4) of the weights and the mix loss are modified to use this new learning rate:

$$w_{t,k}^{\text{ah}} = \frac{w_{1,k}^{\text{ah}} e^{-\eta_t^{\text{ah}} L_{t-1,k}}}{\boldsymbol{w}_1^{\text{ah}} \cdot e^{-\eta_t^{\text{ah}} \boldsymbol{L}_{t-1}}} \qquad \text{and} \qquad m_t^{\text{ah}} = -\frac{1}{\eta_t^{\text{ah}}} \ln(\boldsymbol{w}_t^{\text{ah}} \cdot e^{-\eta_t^{\text{ah}} \boldsymbol{\ell}_t}), \tag{9}$$

with $\boldsymbol{w}_1^{\text{ah}} = (1/K, \ldots, 1/K)$ uniform. Note that the multiplicative update rule for the weights no longer applies when the learning rate varies with t; the last three results of Lemma 1 are also no longer valid. Later we will also consider other algorithms to determine

Single round quantities for trial t:		
$\boldsymbol{\ell}_t$		Loss vector
ℓ_t^-	$= \min_k \ell_{t,k}$, $\ell_t^+ = \max_k \ell_{t,k}$	Min and max loss
s_t	$= \ell_t^+ - \ell_t^-$	Loss range
$\boldsymbol{w}_t^{\mathrm{alg}}$	$= e^{-\eta_t^{\mathrm{alg}} \cdot \boldsymbol{L}_{t-1}} / \sum_k e^{-\eta_t^{\mathrm{alg}} L_{t-1,k}}$	Weights played
h_t^{alg}	$= \boldsymbol{w}_t^{\mathrm{alg}} \cdot \boldsymbol{\ell}_t$	Hedge loss
m_t^{alg}	$= -\frac{1}{\eta_t^{\mathrm{alg}}} \ln \left(\boldsymbol{w}_t^{\mathrm{alg}} \cdot e^{-\eta_t^{\mathrm{alg}} \boldsymbol{\ell}_t} \right)$	Mix loss
δ_t^{alg}	$= h_t^{\mathrm{alg}} - m_t^{\mathrm{alg}}$	Mixability gap
v_t^{alg}	$= \mathrm{Var}_{k \sim \boldsymbol{w}_t^{\mathrm{alg}}}[\ell_{t,k}]$	Loss variance
Aggregate quantities after t rounds:		
(The final time T is omitted from the subscript where possible, e.g. $L^* = L_T^*$)		
$\boldsymbol{L}_t$, L_t^-, L_t^+, H_t^{alg}, M_t^{alg}, Δ_t^{alg}, V_t^{alg}		$\sum_{\tau=1}^{t}$ of $\boldsymbol{\ell}_\tau$, ℓ_τ^-, ℓ_τ^+, h_τ^{alg}, m_τ^{alg}, $\delta_\tau^{\mathrm{alg}}$, v_τ^{alg}
S_t	$= \max\{s_1, \ldots, s_t\}$	Maximum loss range
L_t^*	$= \min_k L_{t,k}$	Cumulative loss of the best expert
$\mathcal{R}_t^{\mathrm{alg}}$	$= H_t^{\mathrm{alg}} - L_t^*$	Regret
Algorithms (the "alg" in the superscript above):		
(η)		Hedge with fixed learning rate η
ah		AdaHedge, defined by (8)
ftl		Follow-the-Leader ($\eta^{\mathrm{ftl}} = \infty$)
ff		FlipFlop, defined by (16)

Table 1: Notation

variable learning rates; to avoid confusion the considered algorithm is always specified in the superscript in our notation. See Table 1 for reference. From now on, AdaHedge will be defined as the Hedge algorithm with learning rate defined by (8). For concreteness, a MATLAB implementation appears in Figure 1.

Our learning rate is similar to that of Cesa-Bianchi et al. (2007), but it is less pessimistic as it is based on the mixability gap Δ_t itself rather than its bound, and as such may exploit easy sequences of losses more aggressively. Moreover our tuning of the learning rate simplifies the analysis, leading to tighter results; the essential new technical ingredients appear as Lemmas 3, 5 and 7 below.

We analyse the regret for AdaHedge like we did for a fixed learning rate in the previous section: we again consider $M^{\mathrm{ah}} - L^*$ and Δ^{ah} separately. This time, both legs of the analysis become slightly more involved. Luckily, a good bound can still be obtained with only a small amount of work. First we show that the mix loss is bounded by the mix loss we would have incurred if we would have used the final learning rate η_T^{ah} all along:

Lemma 2 *Let* dec *be any strategy for choosing the learning rate such that $\eta_1 \geq \eta_2 \geq \ldots$ Then the cumulative mix loss for* dec *does not exceed the cumulative mix loss for the strategy that uses the last learning rate η_T from the start: $M^{\mathrm{dec}} \leq M^{(\eta_T)}$.*

```
% Returns the losses of AdaHedge.
% l(t,k) is the loss of expert k at time t
function h = adahedge(l)
    [T, K]  = size(l);
    h       = nan(T,1);
    L       = zeros(1,K);
    Delta   = 0;

    for t = 1:T
        eta = log(K)/Delta;
        [w, Mprev] = mix(eta, L);
        h(t) = w * l(t,:)';
        L = L + l(t,:);
        [~, M] = mix(eta, L);
        delta = max(0, h(t)-(M-Mprev));
        % max clips numeric Jensen violation
        Delta = Delta + delta;
    end
end
```

```
% Returns the posterior weights and mix loss
% for learning rate eta and cumulative loss
% vector L, avoiding numerical instability.
function [w, M] = mix(eta, L)
    mn = min(L);
    if (eta == Inf) % Limit behaviour: FTL
        w = L==mn;
    else
        w = exp(-eta .* (L-mn));
    end
    s = sum(w);
    w = w / s;
    M = mn - log(s/length(L))/eta;
end
```

Figure 1: Numerically robust MATLAB implementation of AdaHedge

This lemma was first proved in its current form by Kalnishkan and Vyugin (2005, Lemma 3), and an essentially equivalent bound was introduced by Györfi and Ottucsák (2007) in the proof of their Lemma 1. Related techniques for dealing with time-varying learning rates go back to Auer et al. (2002).

Proof Using mix loss property #4, we have

$$M_T^{\text{dec}} = \sum_{t=1}^{T} m_t^{\text{dec}} = \sum_{t=1}^{T} \left(M_t^{(\eta_t)} - M_{t-1}^{(\eta_t)} \right) \leq \sum_{t=1}^{T} \left(M_t^{(\eta_t)} - M_{t-1}^{(\eta_{t-1})} \right) = M_T^{(\eta_T)},$$

which was to be shown. ■

We can now show that the two contributions to the regret are still balanced.

Lemma 3 *The AdaHedge regret is $\mathcal{R}^{\text{ah}} = M^{\text{ah}} - L^* + \Delta^{\text{ah}} \leq 2\Delta^{\text{ah}}$.*

Proof As $\delta_t^{\text{ah}} \geq 0$ for all t (by mix loss property #1), the cumulative mixability gap Δ_t^{ah} is nondecreasing. Consequently, the AdaHedge learning rate η_t^{ah} as defined in (8) is nonincreasing in t. Thus Lemma 2 applies to M^{ah}; together with mix loss property #3 and (8) this yields

$$M^{\text{ah}} \leq M^{(\eta_T^{\text{ah}})} \leq L^* + \frac{\ln K}{\eta_T^{\text{ah}}} = L^* + \Delta_{T-1}^{\text{ah}} \leq L^* + \Delta_T^{\text{ah}}.$$

Substitution into the trivial decomposition $\mathcal{R}^{\text{ah}} = M^{\text{ah}} - L^* + \Delta^{\text{ah}}$ yields the result. ■

The remaining task is to establish a bound on Δ^{ah}. As before, we start with a bound on the mixability gap in a single round, but rather than (6), we use Bernstein's bound on the mixability gap in a single round to obtain a result that is expressed in terms of the variance of the losses, $v_t^{\text{ah}} = \text{Var}_{k \sim \boldsymbol{w}_t^{\text{ah}}}[\ell_{t,k}] = \sum_k w_{t,k}^{\text{ah}} (\ell_{t,k} - h_t^{\text{ah}})^2$.

Lemma 4 (Bernstein's Bound) *Let $\eta_t = \eta_t^{\mathrm{alg}} \in (0, \infty)$ denote the finite learning rate chosen for round t by any algorithm "alg". The mixability gap δ_t^{alg} satisfies*

$$\delta_t^{\mathrm{alg}} \leq \frac{g(s_t \eta_t)}{s_t} v_t^{\mathrm{alg}}, \quad \text{where} \quad g(x) = \frac{e^x - x - 1}{x}. \tag{10}$$

Further, $v_t^{\mathrm{alg}} \leq (\ell_t^+ - h_t^{\mathrm{alg}})(h_t^{\mathrm{alg}} - \ell_t^-) \leq s_t^2/4$.

Proof This is Bernstein's bound (Cesa-Bianchi and Lugosi, 2006, Lemma A.5) on the cumulant generating function, applied to the random variable $(\ell_{t,k} - \ell_t^-)/s_t \in [0, 1]$ with k distributed according to $\boldsymbol{w}_t^{\mathrm{alg}}$. ■

Bernstein's bound is more sophisticated than Hoeffding's bound (6), because it expresses that the mixability gap δ_t is small not only when η_t is small, but also when all experts have approximately the same loss, or when the weights $\boldsymbol{w}_t$ are concentrated on a single expert.

The next step is to use Bernstein's inequality to obtain a bound on the cumulative mixability gap Δ^{ah}. In the analysis of Cesa-Bianchi et al. (2007) this is achieved by first applying Bernstein's bound for each individual round, and then using a telescoping argument to obtain a bound on the sum. With our learning rate (8) it is convenient to reverse these steps: we first telescope, which can now be done with equality, and subsequently apply Bernstein's inequality in a stricter way.

Lemma 5 *AdaHedge's cumulative mixability gap satisfies*

$$\left(\Delta^{\mathrm{ah}}\right)^2 \leq V^{\mathrm{ah}} \ln K + (\tfrac{2}{3} \ln K + 1) S \Delta^{\mathrm{ah}}.$$

Proof In this proof we will omit the superscript "ah". Using the definition of the learning rate (8) and $\delta_t \leq s_t$ (from mix loss property #1), we get

$$\begin{aligned} \Delta^2 &= \sum_{t=1}^{T} \left(\Delta_t^2 - \Delta_{t-1}^2\right) = \sum_t \left((\Delta_{t-1} + \delta_t)^2 - \Delta_{t-1}^2\right) = \sum_t \left(2\delta_t \Delta_{t-1} + \delta_t^2\right) \\ &= \sum_t \left(2\delta_t \frac{\ln K}{\eta_t} + \delta_t^2\right) \leq \sum_t \left(2\delta_t \frac{\ln K}{\eta_t} + s_t \delta_t\right) \leq 2 \ln K \sum_t \frac{\delta_t}{\eta_t} + S\Delta. \end{aligned} \tag{11}$$

The inequalities in this equation replace a δ_t term by S, which is of no concern: the resulting term $S\Delta$ adds at most $2S$ to the regret bound. We will now show

$$\frac{\delta_t}{\eta_t} \leq \tfrac{1}{2} v_t + \tfrac{1}{3} s_t \delta_t. \tag{12}$$

This supersedes the bound $\delta_t/\eta_t \leq (e-2)v_t$ for $\eta_t s_t \leq 1$ used by Cesa-Bianchi et al. (2007). Even though at first sight circular, the form (12) has two major advantages. First, inclusion of the overhead $\frac{1}{3} s_t \delta_t$ will only affect smaller order terms of the regret, but admits a reduction of the leading constant to the optimal factor $\frac{1}{2}$. This gain directly percolates to our regret bounds below. Second, (12) holds for unbounded η_t, which simplifies tuning considerably.

First note that (12) is clearly valid if $\eta_t = \infty$. Assuming that η_t is finite, we can obtain this result by rewriting Bernstein's bound (10) as follows:

$$\tfrac{1}{2}v_t \geq \delta_t \cdot \frac{s_t}{2g(s_t\eta_t)} = \frac{\delta_t}{\eta_t} - s_t f(s_t\eta_t)\delta_t, \quad \text{where} \quad f(x) = \frac{e^x - \frac{1}{2}x^2 - x - 1}{xe^x - x^2 - x}.$$

Remains to show that $f(x) \leq 1/3$ for all $x \geq 0$. After rearranging, we find this to be the case if

$$(3-x)e^x \leq \tfrac{1}{2}x^2 + 2x + 3.$$

Taylor expansion of the left-hand side around zero reveals that $(3-x)e^x = \frac{1}{2}x^2 + 2x + 3 - \frac{1}{6}x^3ue^u$ for some $0 \leq u \leq x$, from which the result follows. The proof is completed by plugging (12) into (11) and finally relaxing $s_t \leq S$. ■

Combination of these results yields the following natural regret bound, analogous to Theorem 5 of Cesa-Bianchi et al. (2007).

Theorem 6 *AdaHedge's regret is bounded by*

$$\mathcal{R}^{\text{ah}} \leq 2\sqrt{V^{\text{ah}} \ln K} + S(\tfrac{4}{3}\ln K + 2).$$

Proof Lemma 5 is of the form

$$(\Delta^{\text{ah}})^2 \leq a + b\Delta^{\text{ah}}, \tag{13}$$

with a and b nonnegative numbers. Solving for Δ^{ah} then gives

$$\Delta^{\text{ah}} \leq \tfrac{1}{2}b + \tfrac{1}{2}\sqrt{b^2+4a} \leq \tfrac{1}{2}b + \tfrac{1}{2}(\sqrt{b^2} + \sqrt{4a}) = \sqrt{a} + b,$$

which by Lemma 3 implies that

$$\mathcal{R}^{\text{ah}} \leq 2\sqrt{a} + 2b.$$

Plugging in the values $a = V^{\text{ah}} \ln K$ and $b = S(\frac{2}{3}\ln K + 1)$ from Lemma 5 completes the proof. ■

This first regret bound for AdaHedge is difficult to interpret, because the cumulative loss variance V^{ah} depends on the actions of the AdaHedge strategy itself (through the weights $\boldsymbol{w}_t^{\text{ah}}$). Below, we will derive a regret bound for AdaHedge that depends only on the data. However, AdaHedge has one important property that is captured by this first result that is no longer expressed by the worst-case bound we will derive below. Namely, if the data are easy in the sense that there is a clear best expert, say k^*, then the weights played by AdaHedge will concentrate on that expert. If $w_{t,k^*}^{\text{ah}} \to 1$ as t increases, then the loss variance must decrease: $v_t^{\text{ah}} \to 0$. Thus, Theorem 6 suggests that the AdaHedge regret may be bounded if the weights concentrate on the best expert sufficiently quickly. This indeed turns out to be the case: we can prove that the regret is bounded for the stochastic setting where the loss vectors $\boldsymbol{\ell}_t$ are independent, and $\mathrm{E}[L_{t,k^*} - L_{t,k}] = \Omega(t^\beta)$ for all $k \neq k^*$ and any $\beta > 1/2$. This is an important feature of AdaHedge when it is used as a stand-alone algorithm, and Van Erven et al. (2011) provide a proof for the previous version of the strategy. See Section 5.4 for an example of concentration of the AdaHedge weights. Here we will not pursue this further, because the Follow-the-Leader strategy also incurs bounded loss in that case; we rather focus attention on how to successfully compete with FTL in Section 3.

We now proceed to derive a bound that depends only on the data, using an approach similar to the one taken by Cesa-Bianchi et al. (2007). We first bound the cumulative loss variance as follows:

Lemma 7 *Assume $L^* \leq H$. The cumulative loss variance for AdaHedge satisfies*

$$V^{\mathrm{ah}} \leq S\frac{(L^+ - L^*)(L^* - L^-)}{L^+ - L^-} + 2S\Delta.$$

In the degenerate case $L^- = L^+$ the fraction reads $0/0$, but since we then have $V^{\mathrm{ah}} = 0$, from here on we define the ratio to be zero in that case, which is also its limiting value.

Proof We omit all "ah" superscripts. By Lemma 4 we have $v_t \leq (\ell_t^+ - h_t)(h_t - \ell_t^-)$. Now

$$\begin{aligned} V = \sum_{t=1}^T v_t \leq \sum_t (\ell_t^+ - h_t)(h_t - \ell_t^-) \qquad &\leq S\sum_t \frac{(\ell_t^+ - h_t)(h_t - \ell_t^-)}{s_t} \\ = ST\sum_t \frac{1}{T}\frac{(\ell_t^+ - h_t)(h_t - \ell_t^-)}{(\ell_t^+ - h_t) + (h_t - \ell_t^-)} &\leq S\frac{(L^+ - H)(H - L^-)}{L^+ - L^-}, \end{aligned} \tag{14}$$

where the last inequality is an instance of Jensen's inequality applied to the function B defined on the domain $x, y \geq 0$ by $B(x,y) = \frac{xy}{x+y}$ for $xy > 0$ and $B(x,y) = 0$ for $xy = 0$ to ensure continuity. To verify that B is jointly concave, we will show that the Hessian is negative semi-definite on the interior $xy > 0$. Concavity on the whole domain then follows from continuity. The Hessian, which turns out to be the rank one matrix

$$\nabla^2 B(x,y) = -\frac{2}{(x+y)^3}\begin{pmatrix} y \\ -x \end{pmatrix}\begin{pmatrix} y \\ -x \end{pmatrix}^\intercal,$$

is negative semi-definite since it is a negative scaling of a positive outer product.

Subsequently using $H \geq L^*$ (by assumption) and $H \leq L^* + 2\Delta$ (by Lemma 3) yields

$$\frac{(L^+ - H)(H - L^-)}{L^+ - L^-} \leq \frac{(L^+ - L^*)(L^* + 2\Delta - L^-)}{L^+ - L^-} \leq \frac{(L^+ - L^*)(L^* - L^-)}{L^+ - L^-} + 2\Delta$$

as desired. ■

This can be combined with Lemmas 5 and 3 to obtain our first main result:

Theorem 8 (AdaHedge Worst-Case Regret Bound) *AdaHedge's regret is bounded by*

$$\mathcal{R}^{\mathrm{ah}} \leq 2\sqrt{S\frac{(L^+ - L^*)(L^* - L^-)}{L^+ - L^-}\ln K} + S(\tfrac{16}{3}\ln K + 2). \tag{15}$$

Proof If $H^{\mathrm{ah}} < L^*$, then $\mathcal{R}^{\mathrm{ah}} < 0$ and the result is clearly valid. But if $H^{\mathrm{ah}} \geq L^*$, we can bound V^{ah} using Lemma 7 and plug the result into Lemma 5 to get an inequality of the form (13) with $a = S(L^+ - L^*)(L^* - L^-)/(L^+ - L^-)$ and $b = S(\frac{8}{3}\ln K + 1)$. Following the steps of the proof of Theorem 6 with these modified values for a and b we arrive at the desired result. ■

This bound has several useful properties:

1. It is always smaller than the CBMS bound (1), with a leading constant that has been reduced from the previously best-known value of 2.63 to 2. To see this, note that (15) increases to (1) if we replace L^+ by the upper bound $L^- + ST$. It can be substantially stronger than (1) if the range of the losses s_t is highly variable.

2. The bound is "fundamental", a concept discussed in detail by Cesa-Bianchi et al. (2007): it is invariant to translations of the losses and proportional to their scale. It is therefore valid for arbitrary loss ranges, regardless of sign. In fact, not just the bound, but AdaHedge itself is fundamental in this sense: see Section 4 for a discussion and proof.

3. The regret is small when the best expert either has a very low loss, or a very high loss. The latter is important if the algorithm is to be used for a scenario in which we are provided with a sequence of gain vectors $\boldsymbol{g}_t$ rather than losses: we can transform these gains into losses using $\boldsymbol{\ell}_t = -\boldsymbol{g}_t$, and then run AdaHedge. The bound then implies that we incur small regret if the best expert has very small cumulative gain relative to the minimum gain.

4. The bound is not dependent on the number of trials but only on the losses; it is a "timeless" bound as discussed below.

2.3 What are Timeless Bounds?

All bounds presented for AdaHedge (and FlipFlop) are "timeless". We call a regret bound *timeless* if it does not change under insertion of additional trials where all experts are assigned the same loss. Intuitively, the prediction task does not become more difficult if nature should insert same-loss trials. Since these trials do nothing to differentiate between the experts, they can safely be ignored by the learner without affecting her regret; in fact, many Hedge strategies, including Hedge with a fixed learning rate, FTL, AdaHedge and CBMS already have the property that their future behaviour does not change under such insertions: they are robust against such time dilation. If any strategy does not have this property by itself, it can easily be modified to ignore equal-loss trials.

It is easy to imagine practical scenarios where this robustness property would be important. For example, suppose you hire a number of experts who continually monitor the assets in your portfolio. Usually they do not recommend any changes, but occasionally, when they see a rare opportunity or receive subtle warning signs, they may urge you to trade, resulting in a potentially very large gain or loss. It seems only beneficial to poll the experts often, and there is no reason why the many resulting equal-loss trials should complicate the learning task.

The oldest bounds for Hedge scale with $\sqrt{T}$ or $\sqrt{L^*}$, and are thus not timeless. From the results above we can obtain fundamental and timeless variants with, for parameterless algorithms, the best known leading constants (the first item below follows Corollary 1 of Cesa-Bianchi et al. 2007):

Corollary 9 *The AdaHedge regret satisfies the following inequalities:*

$$\mathcal{R}^{\text{ah}} \le \sqrt{\textstyle\sum_{t=1}^{T} s_t^2 \ln K} + S(\tfrac{4}{3}\ln K + 2) \qquad \text{(analogue of traditional } T\text{-based bounds),}$$
$$\mathcal{R}^{\text{ah}} \le 2\sqrt{S(L^* - L^-)\ln K} + S(\tfrac{16}{3}\ln K + 2) \qquad \text{(analogue of traditional } L^*\text{-based bounds),}$$
$$\mathcal{R}^{\text{ah}} \le 2\sqrt{S(L^+ - L^*)\ln K} + S(\tfrac{16}{3}\ln K + 2) \qquad \text{(symmetric bound, useful for gains).}$$

Proof We could get a bound that depends only on the loss ranges s_t by substituting the worst case $L^* = (L^+ + L^-)/2$ into Theorem 8, but a sharper result is obtained by plugging the inequality $v_t \le s_t^2/4$ from Lemma 4 directly into Theorem 6. This yields the first item above. The other two inequalities follow easily from Theorem 8. ■

In the next section, we show how we can compete with FTL while at the same time maintaining all these worst-case guarantees up to a constant factor.

3. FlipFlop

AdaHedge balances the cumulative mixability gap Δ^{ah} and the mix loss regret $M^{\text{ah}} - L^*$ by reducing η_t^{ah} as necessary. But, as we observed previously, if the data are not hopelessly adversarial we might not need to worry about the mixability gap: as Lemma 4 expresses, δ_t^{ah} is also small if the variance v_t^{ah} of the loss under the weights $w_{t,k}^{\text{ah}}$ is small, which is the case if the weight on the best expert $\max_k w_{t,k}^{\text{ah}}$ becomes close to one.

AdaHedge is able to exploit such a lucky scenario to an extent: as explained in the discussion that follows Theorem 6, if the weight of the best expert goes to one quickly, AdaHedge will have a small cumulative mixability gap, and therefore, by Lemma 3, a small regret. This happens, for example, in the stochastic setting with independent, identically distributed losses, when a single expert has the smallest expected loss. Similarly, in the experiment of Section 5.4, the AdaHedge weights concentrate sufficiently quickly for the regret to be bounded.

There is the potential for a nasty feedback loop, however. Suppose there are a small number of difficult early trials, during which the cumulative mixability gap increases relatively quickly. AdaHedge responds by reducing the learning rate (8), with the effect that the weights on the experts become more uniform. As a consequence, the mixability gap in future trials may be larger than what it would have been if the learning rate had stayed high, leading to further unnecessary reductions of the learning rate, and so on. The end result may be that AdaHedge behaves as if the data are difficult and incurs substantial regret, even in cases where the regret of Hedge with a fixed high learning rate, or of Follow-the-Leader, is bounded! Precisely this phenomenon occurs in the experiment in Section 5.2 below: AdaHedge's regret is close to the worst-case bound, whereas FTL hardly incurs any regret at all.

It appears, then, that we must *either* hope that the data are easy enough that we can make the weights concentrate quickly on a single expert, by not reducing the learning rate at all; *or* we fear the worst and reduce the learning rate as much as we need to be able to provide good guarantees. We cannot really interpolate between these two extremes: an intermediate learning rate may not yield small regret in favourable cases and may at the same time destroy any performance guarantees in the worst case.

It is unclear a priori whether we can get away with keeping the learning rate high, or that it is wiser to play it safe using AdaHedge. The most extreme case of keeping the learning rate high, is the limit as η tends to ∞, for which Hedge reduces to Follow-the-Leader. In this section we work out a strategy that combines the advantages of FTL and AdaHedge: it retains AdaHedge's worst-case guarantees up to a constant factor, but its regret is also bounded by a constant times the regret of FTL (Theorem 15). Perhaps surprisingly, this is not easy to achieve. To see why, imagine a scenario where the average loss of the best expert is substantial, whereas the *regret* of either Follow-the-Leader or AdaHedge, is small. Since our combination has to guarantee a similarly small regret, it has only a very limited margin for error. We cannot, for example, simply combine the two algorithms by recursively plugging them into Hedge with a fixed learning rate, or into AdaHedge: the performance guarantees we have for those methods of combination are too weak. Even if both FTL and AdaHedge yield small regret on the original problem, choosing the actions of FTL for some rounds and those of AdaHedge for the other rounds may fail if we do it naively, because the regret is not necessarily increasing, and we may end up picking each algorithm precisely in those rounds where the other one is better.

Luckily, alternating between the optimistic FTL strategy and the worst-case-proof AdaHedge does turn out to be possible if we do it in a careful way. In this section we explain the appropriate strategy, called FlipFlop (superscript: "ff"), and show that it combines the desirable properties of both FTL and AdaHedge.

3.1 Exploiting Easy Data by Following the Leader

We first investigate the potential benefits of FTL over AdaHedge. Lemma 10 below identifies the circumstances under which FTL will perform well, which is when the number of leader changes is small. It also shows that the regret for FTL is equal to its cumulative mixability gap when FTL is interpreted as a Hedge strategy with infinite learning rate.

Lemma 10 *Let c_t be an indicator for a leader change at time t: define $c_t = 1$ if there exists an expert k such that $L_{t-1,k} = L^*_{t-1}$ while $L_{t,k} \neq L^*_t$, and $c_t = 0$ otherwise. Let $C_t = c_1 + \ldots + c_t$ be the cumulative number of leader changes. Then the FTL regret satisfies*

$$\mathcal{R}^{\text{ftl}} = \Delta^{(\infty)} \leq S\,C.$$

Proof We have $M^{(\infty)} = L^*$ by mix loss property #3, and consequently $\mathcal{R}^{\text{ftl}} = \Delta^{(\infty)} + M^{(\infty)} - L^* = \Delta^{(\infty)}$.

To bound $\Delta^{(\infty)}$, notice that, for any t such that $c_t = 0$, all leaders remained leaders and incurred identical loss. It follows that $m_t^{(\infty)} = L^*_t - L^*_{t-1} = h_t^{(\infty)}$ and hence $\delta_t^{(\infty)} = 0$. By bounding $\delta_t^{(\infty)} \leq S$ for all other t we obtain

$$\Delta^{(\infty)} = \sum_{t=1}^{T} \delta_t^{(\infty)} = \sum_{t\colon c_t=1} \delta_t^{(\infty)} \leq \sum_{t\colon c_t=1} S = S\,C,$$

as required. ■

We see that the regret for FTL is bounded by the number of leader changes. This quantity is both fundamental and timeless. It is a natural measure of the difficulty of the problem, because it remains small whenever a single expert makes the best predictions on average, even in the scenario described above, in which AdaHedge gets caught in a feedback loop. One example where FTL outperforms AdaHedge is when the losses for two experts are $(1,0)$ on the first round, and keep alternating according to $(1,0)$, $(0,1)$, $(1,0)$, $\ldots$ for the remainder of the rounds. Then the FTL regret is only $1/2$, whereas AdaHedge's performance is close to the worst-case bound (because its weights $\boldsymbol{w}_t^{\mathrm{ah}}$ converge to $(1/2, 1/2)$, for which the bound (6) on the mixability gap is tight). This scenario is illustrated further in the experiments, Section 5.2.

3.2 FlipFlop

FlipFlop is a Hedge strategy in the sense that it uses exponential weights defined by (9), but the learning rate η_t^{ff} now alternates between infinity, such that the algorithm behaves like FTL, and the AdaHedge value, which decreases as a function of the mixability gap accumulated over the rounds where AdaHedge is used. In Definition 11 below, we will specify the "flip" regime $\overline{R}_t$, which is the subset of times $\{1, \ldots, t\}$ where we follow the leader by using an infinite learning rate, and the "flop" regime $\underline{R}_t = \{1, \ldots, t\} \setminus \overline{R}_t$, which is the set of times where the learning rate is determined by AdaHedge (mnemonic: the position of the bar refers to the value of the learning rate). We accumulate the mixability gap, the mix loss and the variance for these two regimes separately:

$$\overline{\Delta}_t = \sum_{\tau \in \overline{R}_t} \delta_\tau^{\mathrm{ff}}; \qquad \overline{M}_t = \sum_{\tau \in \overline{R}_t} m_\tau^{\mathrm{ff}}; \qquad \text{(flip)}$$

$$\underline{\Delta}_t = \sum_{\tau \in \underline{R}_t} \delta_\tau^{\mathrm{ff}}; \qquad \underline{M}_t = \sum_{\tau \in \underline{R}_t} m_\tau^{\mathrm{ff}}; \qquad \underline{V}_t = \sum_{\tau \in \underline{R}_t} v_\tau^{\mathrm{ff}}. \qquad \text{(flop)}$$

We also change the learning rate from its definition for AdaHedge in (8) to the following, which differentiates between the two regimes of the strategy:

$$\eta_t^{\mathrm{ff}} = \begin{cases} \eta_t^{\mathrm{flip}} & \text{if } t \in \overline{R}_t, \\ \eta_t^{\mathrm{flop}} & \text{if } t \in \underline{R}_t, \end{cases} \quad \text{where} \quad \eta_t^{\mathrm{flip}} = \eta_t^{\mathrm{ftl}} = \infty \quad \text{and} \quad \eta_t^{\mathrm{flop}} = \frac{\ln K}{\underline{\Delta}_{t-1}}. \tag{16}$$

Like for AdaHedge, $\eta_t^{\mathrm{flop}} = \infty$ as long as $\underline{\Delta}_{t-1} = 0$, which now happens for all t such that $\underline{R}_{t-1} = \emptyset$. Note that while the learning rates are defined separately for the two regimes, the exponential weights (9) of the experts are still always determined using the cumulative losses $L_{t,k}$ over *all* rounds. We also point out that, for rounds $t \in \underline{R}$, the learning rate $\eta_t^{\mathrm{ff}} = \eta_t^{\mathrm{flop}}$ is not equal to η_t^{ah}, because it uses $\underline{\Delta}_{t-1}$ instead of $\Delta_{t-1}^{\mathrm{ah}}$. For this reason, the

```
% Returns the losses of FlipFlop
% l(t,k) is the loss of expert k at time t; phi > 1 and alpha > 0 are parameters
function h = flipflop(l, alpha, phi)
    [T, K]  = size(l);
    h       = nan(T,1);
    L       = zeros(1,K);
    Delta  = [0 0];
    scale  = [phi/alpha alpha];
    regime = 1; % 1=FTL, 2=AH

    for t = 1:T
        if regime==1, eta = Inf; else eta = log(K)/Delta(2); end
        [w, Mprev] = mix(eta, L);
        h(t) = w * l(t,:)';
        L = L + l(t,:);
        [~, M] = mix(eta, L);
        delta = max(0, h(t)-(M-Mprev));
        Delta(regime) = Delta(regime) + delta;
        if Delta(regime) > scale(regime) * Delta(3-regime)
          regime = 3-regime;
        end
    end
end
```

Figure 2: FlipFlop, with new ingredients in boldface

FlipFlop regret may be either better or worse than the AdaHedge regret; our results below only preserve the regret *bound* up to a constant factor. In contrast, we do compete with the *actual* regret of FTL.

It remains to define the “flip” regime $\overline{R}_t$ and the “flop” regime $\underline{R}_t$, which we will do by specifying the times at which to switch from one to the other. FlipFlop starts optimistically, with an epoch of the “flip” regime, which means it follows the leader, until $\overline{\Delta}_t$ becomes too large compared to $\underline{\Delta}_t$. At that point it switches to an epoch of the “flop” regime, and keeps using η_t^{flop} until $\underline{\Delta}_t$ becomes too large compared to $\overline{\Delta}_t$. Then the process repeats with the next epochs of the “flip” and “flop” regimes. The regimes are determined as follows:

Definition 11 (FlipFlop’s Regimes) *Let $\varphi > 1$ and $\alpha > 0$ be parameters of the algorithm (tuned below in Corollary 16). Then*

- *FlipFlop starts in the “flip” regime.*
- *If t is the earliest time since the start of a “flip” epoch where $\overline{\Delta}_t > (\varphi/\alpha)\underline{\Delta}_t$, then the transition to the subsequent “flop” epoch occurs between rounds t and $t+1$. (Recall that during “flip” epochs $\overline{\Delta}_t$ increases in t whereas $\underline{\Delta}_t$ is constant.)*
- *Vice versa, if t is the earliest time since the start of a “flop” epoch where $\underline{\Delta}_t > \alpha\overline{\Delta}_t$, then the transition to the subsequent “flip” epoch occurs between rounds t and $t+1$.*

This completes the definition of the FlipFlop strategy. See Figure 2 for a MATLAB implementation.

The analysis proceeds much like the analysis for AdaHedge. We first show that, analogously to Lemma 3, the FlipFlop regret can be bounded in terms of the cumulative mixability gap; in fact, we can use the *smallest* cumulative mixability gap that we encountered in either of the two regimes, at the cost of slightly increased constant factors. This is the fundamental building block in our FlipFlop analysis. We then proceed to develop analogues of Lemmas 5 and 7, whose proofs do not have to be changed much to apply to FlipFlop. Finally, all these results are combined to bound the regret of FlipFlop in Theorem 15, which, after Theorem 8, is the second main result of this paper.

Lemma 12 (FlipFlop version of Lemma 3) *The following two bounds hold simultaneously for the regret of the FlipFlop strategy with parameters $\varphi > 1$ and $\alpha > 0$:*

$$\mathcal{R}^{\text{ff}} \leq \left(\frac{\varphi\alpha}{\varphi - 1} + 2\alpha + 1\right)\overline{\Delta} + S\left(\frac{\varphi}{\varphi - 1} + 2\right); \tag{17}$$

$$\mathcal{R}^{\text{ff}} \leq \left(\frac{\varphi}{\varphi - 1} + \frac{\varphi}{\alpha} + 2\right)\underline{\Delta} + S. \tag{18}$$

Proof The regret can be decomposed as

$$\mathcal{R}^{\text{ff}} = H^{\text{ff}} - L^* = \overline{\Delta} + \underline{\Delta} + \overline{M} + \underline{M} - L^*. \tag{19}$$

Our first step will be to bound the mix loss $\overline{M} + \underline{M}$ in terms of the mix loss M^{flop} of the auxiliary strategy that uses η_t^{flop} for all t. As η_t^{flop} is nonincreasing, we can then apply Lemma 2 and mix loss property #3 to further bound

$$M^{\text{flop}} \leq M^{(\eta_T^{\text{flop}})} \leq L^* + \frac{\ln K}{\eta^{\text{flop}}} = L^* + \underline{\Delta}_{T-1} \leq L^* + \underline{\Delta}. \tag{20}$$

Let $0 = u_1 < u_2 < \ldots < u_b < T$ denote the times just before the epochs of the "flip" regime begin, i.e. round $u_i + 1$ is the first round in the i-th "flip" epoch. Similarly let $0 < v_1 < \ldots < v_b \leq T$ denote the times just before the epochs of the "flop" regime begin, where we artificially define $v_b = T$ if the algorithm is in the "flip" regime after T rounds. These definitions ensure that we always have $u_b < v_b \leq T$. For the mix loss in the "flop" regime we have

$$\underline{M} = (M_{u_2}^{\text{flop}} - M_{v_1}^{\text{flop}}) + (M_{u_3}^{\text{flop}} - M_{v_2}^{\text{flop}}) + \ldots + (M_{u_b}^{\text{flop}} - M_{v_{b-1}}^{\text{flop}}) + (M^{\text{flop}} - M_{v_b}^{\text{flop}}). \tag{21}$$

Let us temporarily write $\eta_t = \eta_t^{\text{flop}}$ to avoid double superscripts. For the "flip" regime, the properties in Lemma 1, together with the observation that η_t^{flop} does not change during the "flip" regime, give

$$\begin{aligned}
\overline{M} &= \sum_{i=1}^{b}\left(M_{v_i}^{(\infty)} - M_{u_i}^{(\infty)}\right) = \sum_{i=1}^{b}\left(M_{v_i}^{(\infty)} - L_{u_i}^*\right) \leq \sum_{i=1}^{b}\left(M_{v_i}^{(\eta_{v_i})} - L_{u_i}^*\right)\\
&\leq \sum_{i=1}^{b}\left(M_{v_i}^{(\eta_{v_i})} - M_{u_i}^{(\eta_{v_i})} + \frac{\ln K}{\eta_{v_i}}\right) = \sum_{i=1}^{b}\left(M_{v_i}^{\text{flop}} - M_{u_i}^{\text{flop}} + \frac{\ln K}{\eta_{u_i+1}}\right)\\
&= \left(M_{v_1}^{\text{flop}} - M_{u_1}^{\text{flop}}\right) + \left(M_{v_2}^{\text{flop}} - M_{u_2}^{\text{flop}}\right) + \ldots + \left(M_{v_b}^{\text{flop}} - M_{u_b}^{\text{flop}}\right) + \sum_{i=1}^{b}\underline{\Delta}_{u_i}.
\end{aligned} \tag{22}$$

From the definition of the regime changes (Definition 11), we know the value of $\underline{\Delta}_{u_i}$ very accurately at the time u_i of a change from a "flop" to a "flip" regime:

$$\underline{\Delta}_{u_i} > \alpha\overline{\Delta}_{u_i} = \alpha\overline{\Delta}_{v_{i-1}} > \varphi\underline{\Delta}_{v_{i-1}} = \varphi\underline{\Delta}_{u_{i-1}}.$$

By unrolling from low to high i, we see that

$$\sum_{i=1}^{b} \underline{\Delta}_{u_i} \leq \sum_{i=1}^{b} \varphi^{1-i} \underline{\Delta}_{u_b} \leq \sum_{i=1}^{\infty} \varphi^{1-i} \underline{\Delta}_{u_b} = \frac{\varphi}{\varphi - 1} \underline{\Delta}_{u_b}.$$

Adding up (21) and (22), we therefore find that the total mix loss is bounded by

$$\overline{M} + \underline{M} \leq M^{\text{flop}} + \sum_{i=1}^{b} \underline{\Delta}_{u_i} \leq M^{\text{flop}} + \frac{\varphi}{\varphi - 1} \underline{\Delta}_{u_b} \leq L^* + \left(\frac{\varphi}{\varphi - 1} + 1 \right) \underline{\Delta},$$

where the last inequality uses (20). Combination with (19) yields

$$\mathcal{R}^{\text{ff}} \leq \left(\frac{\varphi}{\varphi - 1} + 2 \right) \underline{\Delta} + \overline{\Delta}. \tag{23}$$

Our next goal is to relate $\underline{\Delta}$ and $\overline{\Delta}$: by construction of the regimes, they are always within a constant factor of each other. First, suppose that after T trials we are in the bth epoch of the "flip" regime, that is, we will behave like FTL in round $T + 1$. In this state, we know from Definition 11 that $\underline{\Delta}$ is stuck at the value $\underline{\Delta}_{u_b}$ that prompted the start of the current epoch. As the regime change happened after u_b, we have $\underline{\Delta}_{u_b} - S \leq \alpha \overline{\Delta}_{u_b}$, so that $\underline{\Delta} - S \leq \alpha \overline{\Delta}$. At the same time, we know that $\overline{\Delta}$ is not large enough to trigger the next regime change. From this we can deduce the following bounds:

$$\frac{1}{\alpha}(\underline{\Delta} - S) \leq \overline{\Delta} \leq \frac{\varphi}{\alpha} \underline{\Delta}.$$

On the other hand, if after T rounds we are in the bth epoch of the "flop" regime, then a similar reasoning yields

$$\frac{\alpha}{\varphi}(\overline{\Delta} - S) \leq \underline{\Delta} \leq \alpha \overline{\Delta}.$$

In both cases, it follows that

$$\underline{\Delta} < \alpha \overline{\Delta} + S;$$
$$\overline{\Delta} < \frac{\varphi}{\alpha} \underline{\Delta} + S.$$

The two bounds of the lemma are obtained by plugging first one, then the other of these bounds into (23). ■

The "flop" cumulative mixability gap $\underline{\Delta}$ is related, as before, to the variance of the losses.

Lemma 13 (FlipFlop version of Lemma 5) *The cumulative mixability gap for the "flop" regime is bounded by the cumulative variance of the losses for the "flop" regime:*

$$\underline{\Delta}^2 \leq \underline{V} \ln K + (\tfrac{2}{3} \ln K + 1) S \underline{\Delta}. \tag{24}$$

Proof The proof is analogous to the proof of Lemma 5, with $\underline{\Delta}$ instead of Δ^{ah}, $\underline{V}$ instead of V^{ah}, and using $\eta_t = \eta_t^{\text{flop}} = \ln(K)/\underline{\Delta}_{t-1}$ instead of $\eta_t = \eta_t^{\text{ah}} = \ln(K)/\Delta_{t-1}^{\text{ah}}$. Furthermore, we only need to sum over the rounds $\underline{R}$ in the "flop" regime, because $\underline{\Delta}$ does not change during the "flip" regime. ■

As it is straight-forward to prove an analogue of Theorem 6 for FlipFlop by solving the quadratic inequality in (24), we proceed directly towards establishing an analogue of Theorem 8. The following lemma provides the equivalent of Lemma 7 for FlipFlop. It can probably be strengthened to improve the lower order terms; we provide the version that is easiest to prove.

Lemma 14 (FlipFlop version of Lemma 7) *Suppose $H^{\text{ff}} \geq L^*$. The cumulative loss variance for FlipFlop with parameters $\varphi > 1$ and $\alpha > 0$ satisfies*

$$\underline{V} \leq S\frac{(L^+ - L^*)(L^* - L^-)}{L^+ - L^-} + \left(\frac{\varphi}{\varphi - 1} + \frac{\varphi}{\alpha} + 2\right) S\underline{\Delta} + S^2.$$

Proof The sum of variances satisfies

$$\underline{V} = \sum_{t \in \underline{R}} v_t^{\text{ff}} \leq \sum_{t=1}^{T} v_t^{\text{ff}} \leq S\frac{(L^+ - H^{\text{ff}})(H^{\text{ff}} - L^-)}{L^+ - L^-},$$

where the first inequality simply includes the variances for FTL rounds (which are often all zero), and the second follows from the same reasoning as employed in (14). Subsequently using $L^* \leq H^{\text{ff}}$ (by assumption) and, from Lemma 12, $H^{\text{ff}} \leq L^* + \gamma$, where γ denotes the right-hand side of the bound (18), we find

$$\underline{V} \leq S\frac{(L^+ - L^*)(L^* + \gamma - L^-)}{L^+ - L^-} \leq S\frac{(L^+ - L^*)(L^* - L^-)}{L^+ - L^-} + S\gamma,$$

which was to be shown. ■

Combining Lemmas 12, 13 and 14, we obtain our second main result:

Theorem 15 (FlipFlop Regret Bound) *The regret for FlipFlop with doubling parameters $\varphi > 1$ and $\alpha > 0$ simultaneously satisfies the two bounds*

$$\mathcal{R}^{\text{ff}} \leq \left(\frac{\varphi\alpha}{\varphi - 1} + 2\alpha + 1\right)\mathcal{R}^{\text{ftl}} + S\left(\frac{\varphi}{\varphi - 1} + 2\right),$$

$$\mathcal{R}^{\text{ff}} \leq c_1\sqrt{S\frac{(L^+ - L^*)(L^* - L^-)}{L^+ - L^-}\ln K} + c_1 S\left((c_1 + \tfrac{2}{3})\ln K + \sqrt{\ln K} + 1\right) + S,$$

where $c_1 = \dfrac{\varphi}{\varphi - 1} + \dfrac{\varphi}{\alpha} + 2$.

This shows that, up to a multiplicative factor in the regret, FlipFlop is always as good as the best of Follow-the-Leader and AdaHedge's bound from Theorem 8. Of course, if

AdaHedge significantly outperforms its bound, it is not guaranteed that FlipFlop will outperform the bound in the same way.

In the experiments in Section 5 we demonstrate that the multiplicative factor is not just an artifact of the analysis, but can actually be observed on simulated data.

Proof From Lemma 10, we know that $\overline{\Delta} \leq \Delta^{(\infty)} = \mathcal{R}^{\text{ftl}}$. Substitution in (17) of Lemma 12 yields the first inequality.

For the second inequality, note that $L^* > H^{\text{ff}}$ means the regret is negative, in which case the result is clearly valid. We may therefore assume w.l.o.g. that $L^* \leq H^{\text{ff}}$ and apply Lemma 14. Combination with Lemma 13 yields

$$\underline{\Delta}^2 \leq \underline{V} \ln K + (\tfrac{2}{3} \ln K + 1) S \underline{\Delta} \leq S \frac{(L^+ - L^*)(L^* - L^-)}{L^+ - L^-} \ln K + S^2 \ln K + c_2 S \underline{\Delta},$$

where $c_2 = (c_1 + \frac{2}{3}) \ln K + 1$. We now solve this quadratic inequality as in (13) and relax it using $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ for nonnegative numbers a, b to obtain

$$\begin{aligned} \underline{\Delta} &\leq \sqrt{S \frac{(L^+ - L^*)(L^* - L^-)}{L^+ - L^-} \ln K + S^2 \ln K} + c_2 S \\ &\leq \sqrt{S \frac{(L^+ - L^*)(L^* - L^-)}{L^+ - L^-} \ln K} + S \left(\sqrt{\ln K} + c_2 \right). \end{aligned}$$

In combination with Lemma 12, this yields the second bound of the theorem. ■

Finally, we propose to select the parameter values that minimize the constant factor in front of the leading terms of these regret bounds.

Corollary 16 *The parameter values* $\varphi^* = 2.37$ *and* $\alpha^* = 1.243$ *approximately minimize the worst of the two leading factors in the bounds of Theorem 15. The regret for FlipFlop with these parameters is simultaneously bounded by*

$$\begin{aligned} &\mathcal{R}^{\text{ff}} \leq 5.64 \mathcal{R}^{\text{ftl}} + 3.73 S, \\ &\mathcal{R}^{\text{ff}} \leq 5.64 \sqrt{S \frac{(L^+ - L^*)(L^* - L^-)}{L^+ - L^-} \ln K} + S \left(35.53 \ln K + 5.64 \sqrt{\ln K} + 6.64 \right). \end{aligned}$$

Proof The leading factors $f(\varphi, \alpha) = \frac{\varphi \alpha}{\varphi - 1} + 2\alpha + 1$ and $g(\varphi, \alpha) = \frac{\varphi}{\varphi - 1} + \frac{\varphi}{\alpha} + 2$ are respectively increasing and decreasing in α. They are equalized for $\alpha(\varphi) = (2\varphi - 1 + \sqrt{12\varphi^3 - 16\varphi^2 + 4\varphi + 1})/(6\varphi - 4)$. The analytic solution for the minimum of $f(\varphi, \alpha(\varphi))$ in φ is too long to reproduce here, but it is approximately equal to $\varphi^* = 2.37$, at which point $\alpha(\varphi^*) \approx 1.243$. ■

4. Invariance to Rescaling and Translation

A common simplifying assumption made in the literature is that the losses $\ell_{t,k}$ are translated and normalised to take values in the interval $[0, 1]$. However, doing so requires *a priori* knowledge of the range of the losses. One would therefore prefer algorithms that do not require the losses to be normalised. As discussed by Cesa-Bianchi et al. (2007), the regret bounds for such algorithms should not change when losses are translated (because this does not change the regret) and should scale by σ when the losses are scaled by a factor $\sigma > 0$ (because the regret scales by σ). They call such regret bounds *fundamental* and show that most of the methods they introduce satisfy such fundamental bounds.

Here we go even further: it is not just our bounds that are fundamental, but also our algorithms, which do not change their output weights if the losses are scaled or translated.

Theorem 17 *Both AdaHedge and FlipFlop are invariant to translation and rescaling of the losses. Starting with losses $\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_T$, obtain rescaled, translated losses $\boldsymbol{\ell}'_1, \ldots, \boldsymbol{\ell}'_T$ by picking any $\sigma > 0$ and arbitrary reals $\tau_1, \ldots, \tau_T$, and setting $\ell'_{t,k} = \sigma\ell_{t,k} + \tau_t$ for $t = 1, \ldots, T$ and $k = 1, \ldots, K$. Both AdaHedge and FlipFlop issue the exact same sequence of weights $\boldsymbol{w}'_t = \boldsymbol{w}_t$ on $\boldsymbol{\ell}'_t$ as they do on $\boldsymbol{\ell}_t$.*

Proof We annotate any quantity with a prime to denote that it is defined with respect to the losses $\boldsymbol{\ell}'_t$. We omit the algorithm name from the superscript. First consider AdaHedge. We will prove the following relations by induction on t:

$$\Delta'_{t-1} = \sigma\Delta_{t-1}; \qquad \eta'_t = \frac{\eta_t}{\sigma}; \qquad \boldsymbol{w}'_t = \boldsymbol{w}_t. \tag{25}$$

For $t = 1$, these are valid since $\Delta'_0 = \sigma\Delta_0 = 0$, $\eta'_1 = \eta_1/\sigma = \infty$, and $\boldsymbol{w}'_1 = \boldsymbol{w}_1$ are uniform. Now assume towards induction that (25) is valid for some $t \in \{1, \ldots, T\}$. We can then compute the following values from their definition: $h'_t = \boldsymbol{w}'_t \cdot \boldsymbol{\ell}'_t = \sigma h_t + \tau_t$; $m'_t = -(1/\eta'_t)\ln(\boldsymbol{w}'_t \cdot e^{-\eta'_t\boldsymbol{\ell}'_t}) = \sigma m_t + \tau_t$; $\delta'_t = h'_t - m'_t = \sigma(h_t - m_t) = \sigma\delta_t$. Thus, the mixability gaps are also related by the scale factor σ. From there we can re-establish the induction hypothesis for the next round: we have $\Delta'_t = \Delta'_{t-1} + \delta'_t = \sigma\Delta_{t-1} + \sigma\delta_t = \sigma\Delta_t$, and $\eta'_{t+1} = \ln(K)/\Delta'_t = \eta_{t+1}/\sigma$. For the weights we get $\boldsymbol{w}'_{t+1} \propto e^{-\eta'_{t+1}\boldsymbol{L}'_t} = e^{-(\eta_{t+1}/\sigma)(\sigma\boldsymbol{L}_t)} \propto \boldsymbol{w}_{t+1}$, which means the two must be equal since both sum to one. Thus the relations of (25) are also valid for time $t + 1$, proving the result for AdaHedge.

For FlipFlop, if we assume regime changes occur at the same times for $\boldsymbol{\ell}'$ and $\boldsymbol{\ell}$, then similar reasoning reveals $\overline{\Delta}'_t = \sigma\overline{\Delta}_t$; $\underline{\Delta}'_t = \sigma\underline{\Delta}_t$, $\eta'^{\text{flip}}_t = \eta^{\text{flip}}_t/\sigma = \infty$, $\eta'^{\text{flop}}_t = \eta^{\text{flop}}_t/\sigma$, and $\boldsymbol{w}'_t = \boldsymbol{w}_t$. Remains to check that the regime changes do indeed occur at the same times. Note that in Definition 11, the "flop" regime is started when $\overline{\Delta}'_t > (\varphi/\alpha)\underline{\Delta}'_t$, which is equivalent to testing $\overline{\Delta}_t > (\varphi/\alpha)\underline{\Delta}_t$ since both sides of the inequality are scaled by σ. Similarly, the "flip" regime starts when $\underline{\Delta}'_t > \alpha\overline{\Delta}'_t$, which is equivalent to the test $\underline{\Delta}_t > \alpha\overline{\Delta}_t$. ∎

5. Experiments

We performed four experiments on artificial data, designed to clarify how the learning rate determines performance in a variety of Hedge algorithms. These experiments are designed to illustrate as clearly as possible the intricacies involved in the central question of this paper: whether to use a high learning rate (by following the leader) or to play it safe by using a smaller learning rate instead. Rather than mimic real-world data, on which high learning rates often seem to work well (Devaine et al., 2013), we vary the main factor that appears to drive the best choice of learning rate: the difference in cumulative loss between the experts.

We have kept the experiments as simple as possible: the data are deterministic, and involve two experts. In each case, the data consist of one initial hand-crafted loss vector $\boldsymbol{\ell}_1$, followed by a sequence of loss vectors $\boldsymbol{\ell}_2, \ldots, \boldsymbol{\ell}_T$, which are either $(0, 1)$ or $(1, 0)$. For each experiment $\xi \in \{1, 2, 3, 4\}$, we want the cumulative loss difference $L_{t,1} - L_{t,2}$ between the experts to follow a target $f_\xi(t)$, which will be a continuous, nondecreasing function of t. As the losses are binary, we cannot make $L_{t,1} - L_{t,2}$ exactly equal to the target $f_\xi(t)$, but after the initial loss $\boldsymbol{\ell}_1$, we choose every subsequent loss vector such that it brings $L_{t,1} - L_{t,2}$ as close as possible to $f_\xi(t)$. All functions f_ξ change slowly enough that $|L_{t,1} - L_{t,2} - f_\xi(t)| \leq 1$ for all t.

For each experiment, we let the number of trials be $T = 1000$, and we first plot the regret $\mathcal{R}^{(\eta)}$ of the Hedge algorithm as a function of the fixed learning rate η. We subsequently plot the regret $\mathcal{R}_t^{\text{alg}}$ as a function of $t = 1, \ldots, T$, for each of the following algorithms "alg":

1. Follow-the-Leader (Hedge with learning rate ∞)
2. Hedge with fixed learning rate $\eta = 1$
3. Hedge with the learning rate that optimizes the worst-case bound (7), which equals $\eta = \sqrt{8 \ln(K)/(S^2 T)} \approx 0.0745$; we will call this algorithm "safe Hedge" for brevity.
4. AdaHedge
5. FlipFlop, with parameters $\varphi^* = 2.37$ and $\alpha^* = 1.243$ as in Corollary 16
6. Variation MW by Hazan and Kale (2008), using the fixed learning rate that optimises the bound provided in their Theorem 4
7. NormalHedge, described by Chaudhuri et al. (2009)

Note that the safe Hedge strategy (the third item above) can only be used in practice if the horizon T is known in advance. Variation MW (the sixth item) additionally requires precognition of the empirical variance of the sequence of losses of the best expert up until T (that is, $\text{VAR}_T^{\max}$ as defined in Section 1.2), which is not available in practice, but which we are supplying anyway.

We include algorithms 6 and 7 because, as explained in Section 1.2, they are the state of the art in Hedge-style algorithms. Like AdaHedge, Variation MW is a refinement of the CBMS strategy described by Cesa-Bianchi et al. (2007). They modify the definition of the weights in the Hedge algorithm to include second-order terms; the resulting bound is never more than a constant factor worse than the bounds (1) for CBMS and (15) for AdaHedge, but for some easy data it can be substantially better. For this reason it is a natural performance target for AdaHedge.

The bounds for CBMS and AdaHedge are incomparible with the bound for NormalHedge, being better for some, worse for other data. The reason we include it in the experiments is because, compared to the other methods, its performance in practice turns out to be excellent. We do not know whether there are data sequences on which FlipFlop significantly outperforms NormalHedge, nor whether there is a theoretical reason for this good performance, as the NormalHedge bound (Chaudhuri et al., 2009) is not tight for our experiments.

To reduce clutter, we omit results for CBMS; its behaviour is very similar to that of AdaHedge. Below we provide an exact description of each experiment, and discuss the results.

5.1 Experiment 1. Worst Case for FTL

The experiment is defined by $\boldsymbol{\ell}_1 = (\frac{1}{2}\ 0)$, and $f_1(t) = 0$. This yields the following losses:

$$\begin{pmatrix}1/2\\0\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}1\\0\end{pmatrix}, \ldots$$

These data are the worst case for FTL: each round, the leader incurs loss one, while each of the two individual experts only receives a loss once every two rounds. Thus, the FTL regret increases by one every two rounds and ends up around 500. For any learning rate η, the weights used by the Hedge algorithm are repeated every two rounds, so the regret $H_t - L_t^*$ increases by the same amount every two rounds: the regret increases linearly in t for every fixed η that does not vary with t. However, the constant of proportionality can be reduced greatly by reducing the value of η, as the top graph in Figure 3 shows: for $T = 1000$, the regret becomes negligible for any η less than about 0.01. Thus, in this experiment, a learning algorithm must reduce the learning rate to shield itself from incurring an excessive overhead.

The bottom graph in Figure 3 shows the expected breakdown of the FTL algorithm; Hedge with fixed learning rate $\eta = 1$ also performs quite badly. When η is reduced to the value that optimises the worst-case bound, the regret becomes competitive with that of the other algorithms. Note that Variation MW has the best performance; this is because its learning rate is tuned in relation to the bound proved in the paper, which has a relatively large constant in front of the leading term. As a consequence the algorithm always uses a relatively small learning rate, which turns out to be helpful in this case but harmful in later experiments.

FlipFlop behaves as theory suggests it should: its regret increases alternately like the regret of AdaHedge and the regret of FTL. The latter performs horribly, so during those intervals the regret increases quickly, on the other hand the FTL intervals are relatively short-lived so in the end they do not harm the regret by more than a constant factor.

The NormalHedge algorithm still has acceptable performance, although its regret is relatively large in this experiment; we have no explanation for this but in fairness we do observe good performance of NormalHedge in the other three experiments as well as in numerous further unreported simulations.

5.2 Experiment 2. Best Case for FTL

The second experiment is defined by $\boldsymbol{\ell}_1 = (1, 0)$ and $f_2(t) = 3/2$. This leads to the sequence of losses

$$\begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}, \ldots$$

in which the loss vectors are alternating for $t \geq 2$. These data look very similar to the first experiment, but as the top graph in Figure 4 illustrates, because of the small changes at the start of the sequence, it is now viable to reduce the regret by using a very *high* learning rate. In particular, since there are no leader changes after the first round, FTL incurs a regret of only 1/2.

As in the first experiment, the regret increases linearly in t for every fixed η (provided it is less than ∞); but now the constant of linearity is large only for learning rates close to 1. Once FlipFlop enters the FTL regime for the second time, it stays there indefinitely, which results in bounded regret. After this small change in the setup compared to the previous experiment, NormalHedge also suddenly adapts very well to the data. The behaviour of the other algorithms is very similar to the first experiment: their regret grows without bound.

5.3 Experiment 3. Weights do not Concentrate in AdaHedge

The third experiment uses $\boldsymbol{\ell}_1 = (1, 0)$, and $f_3(t) = t^{0.4}$. The first few loss vectors are the same as in the previous experiment, but every now and then there are two loss vectors $(1, 0)$ in a row, so that the first expert gradually falls behind the second in terms of performance. By $t = T = 1000$, the first expert has accumulated 508 loss, while the second expert has only 492.

For any fixed learning rate η, the weights used by Hedge now concentrate on the second expert. We know from Lemma 4 that the mixability gap in any round t is bounded by a constant times the variance of the loss under the weights played by the algorithm; as these weights concentrate on the second expert, this variance must go to zero. One can show that this happens quickly enough for the cumulative mixability gap to be *bounded* for any fixed η that does not vary with t or depend on T. From (5) we have

$$\mathcal{R}^{(\eta)} = M - L^* + \Delta^{(\eta)} \leq \frac{\ln K}{\eta} + \text{bounded} = \text{bounded}.$$

So in this scenario, as long as the learning rate is kept fixed, we will eventually learn the identity of the best expert. However, if the learning rate is very small, this will happen so slowly that the weights still have not converged by $t = 1000$. Even worse, the top graph in Figure 5 shows that for intermediate values of the learning rate, not only do the weights fail to converge on the second expert sufficiently quickly, but they are sensitive enough to the alternation of the loss vectors to increase the overhead incurred each round.

For this experiment, it really pays to use a large learning rate rather than a safe small one. Thus FTL, Hedge with $\eta = 1$, FlipFlop and NormalHedge perform excellently, while safe Hedge, AdaHedge and Variation MW incur a substantial overhead. Extrapolating the trend in the graph, it appears that the overhead of these algorithms is *not* bounded. This is possible because the three algorithms with poor performance use a learning rate that decreases as a function of t. As a concequence the used learning rate may remain too small for the weights to concentrate. For the case of AdaHedge, this is an example of the “nasty feedback loop” described in Section 3.

5.4 Experiment 4. Weights do Concentrate in AdaHedge

The fourth and last experiment uses $\boldsymbol{\ell}_1 = (1, 0)$, and $f_4(t) = t^{0.6}$. The losses are comparable to those of the third experiment, but the performance gap between the two experts is somewhat larger. By $t = T = 1000$, the two experts have loss 532 and 468, respectively. It is now so easy to determine which of the experts is better that the top graph in Figure 6 is nonincreasing: the larger the learning rate, the better.

The algorithms that managed to keep their regret bounded in the previous experiment obviously still perform very well, but it is clearly visible that AdaHedge now achieves the same. As discussed below Theorem 6, this happens because the weight concentrates on the second expert quickly enough that AdaHedge's regret is bounded in this setting. The crucial difference with the previous experiment is that now we have $f_\xi(t) = t^\beta$ with $\beta > 1/2$. Thus, while the previous experiment shows that AdaHedge can be tricked into reducing the learning rate while it would be better not to do so, the present experiment shows that on the other hand, sometimes AdaHedge does adapt really nicely to easy data, in contrast to algorithms that are tuned in terms of a worst-case bound.

6. Discussion and Conclusion

The main contributions of this work are twofold. First, we develop a new hedging algorithm called AdaHedge. The analysis simplifies existing results and we obtain improved bounds (Theorems 6 and 8). Moreover, AdaHedge is "fundamental" in the sense that its weights are invariant under translation and scaling of the losses (Section 4) and its bounds are "timeless" in the sense that they do not degenerate when rounds are inserted in which all experts incur the same loss. Second, we explain in detail why it is difficult to tune the learning rate such that good performance is obtained both for easy and for hard data, and we address the issue by developing the FlipFlop algorithm. FlipFlop never performs much worse than the Follow-the-Leader strategy, which works very well on easy data (Lemma 10), but it also retains a worst-case bound similar to the bound for AdaHedge (Theorem 15). As such, this work may be seen as solving a special case of a more general question: can we compete with Hedge for any fixed learning rate? We will now briefly discuss this question and then place our work in a broader context, which provides an ambitious agenda for future work.

6.1 General Question: Competing with Hedge for any Fixed Learning Rate

Up to multiplicative constants, FlipFlop is at least as good as FTL and as (the bound for) AdaHedge. These two algorithms represent two extremes of choosing the learning rate η_t in Hedge: FTL takes $\eta_t = \infty$ to exploit easy data, whereas AdaHedge decreases η_t with t to protect against the worst case. It is now natural to ask whether we can design a "Universal Hedge" algorithm that can compete with Hedge with any fixed learning rate $\eta \in (0, \infty]$. That is, for all T, the regret up to time T of Universal Hedge should be within a constant factor C of the regret incurred by Hedge run with the fixed $\hat{\eta}$ that minimizes the Hedge loss $H^{(\hat{\eta})}$. This appears to be a difficult question, and maybe such an algorithm does not even exist. Yet, even partial results (such as an algorithm that competes with $\eta \in [\sqrt{\ln(K)/(S^2T)}, \infty]$ or with a factor C that increases slowly, say, logarithmically, in T) would already be of significant interest.

In this regard, it is interesting to note that, in practice, the learning rates chosen by sophisticated versions of Hedge do not always perform very well; higher learning rates often do better. This is noted by Devaine et al. (2013), who resolve the issue by adapting the learning rate sequentially in an ad-hoc fashion, which works well in their application, but

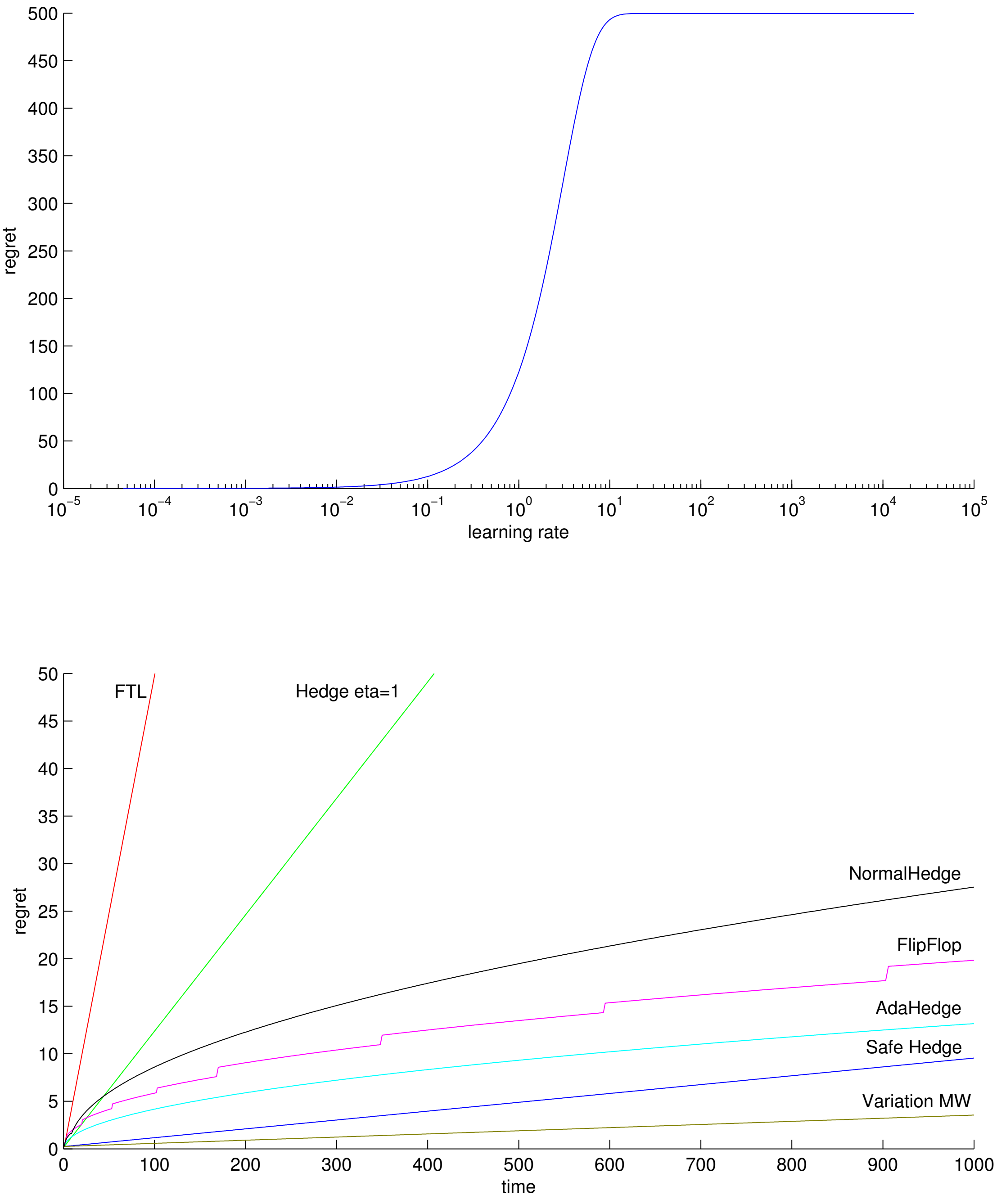


Figure 3: Hedge regret for Experiment 1 (FTL worst-case)

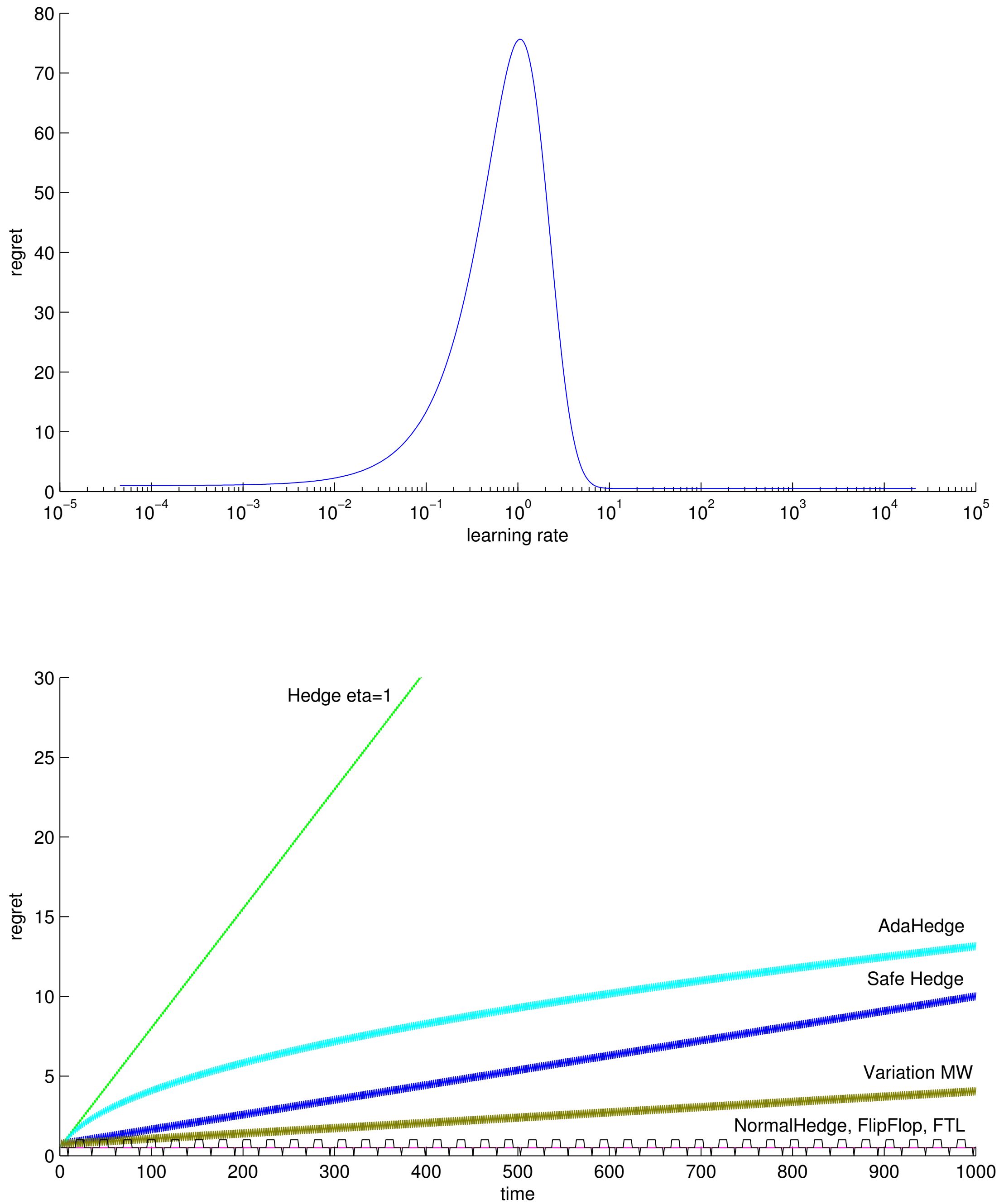
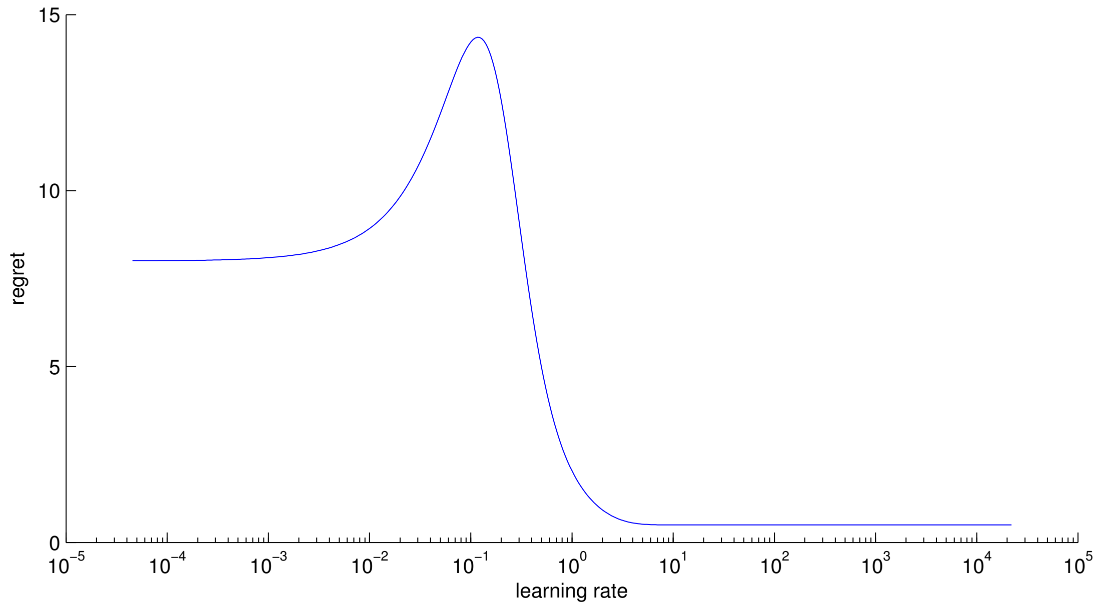


Figure 4: Hedge regret for Experiment 2 (FTL best-case)



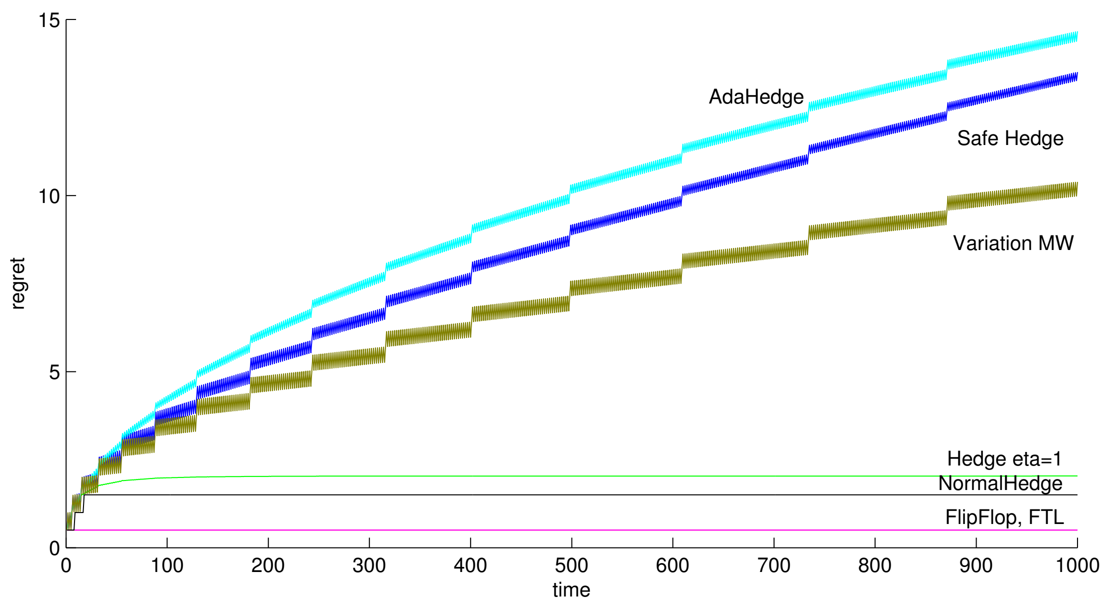


Figure 5: Hedge regret for Experiment 3 (weights do not concentrate in AdaHedge)

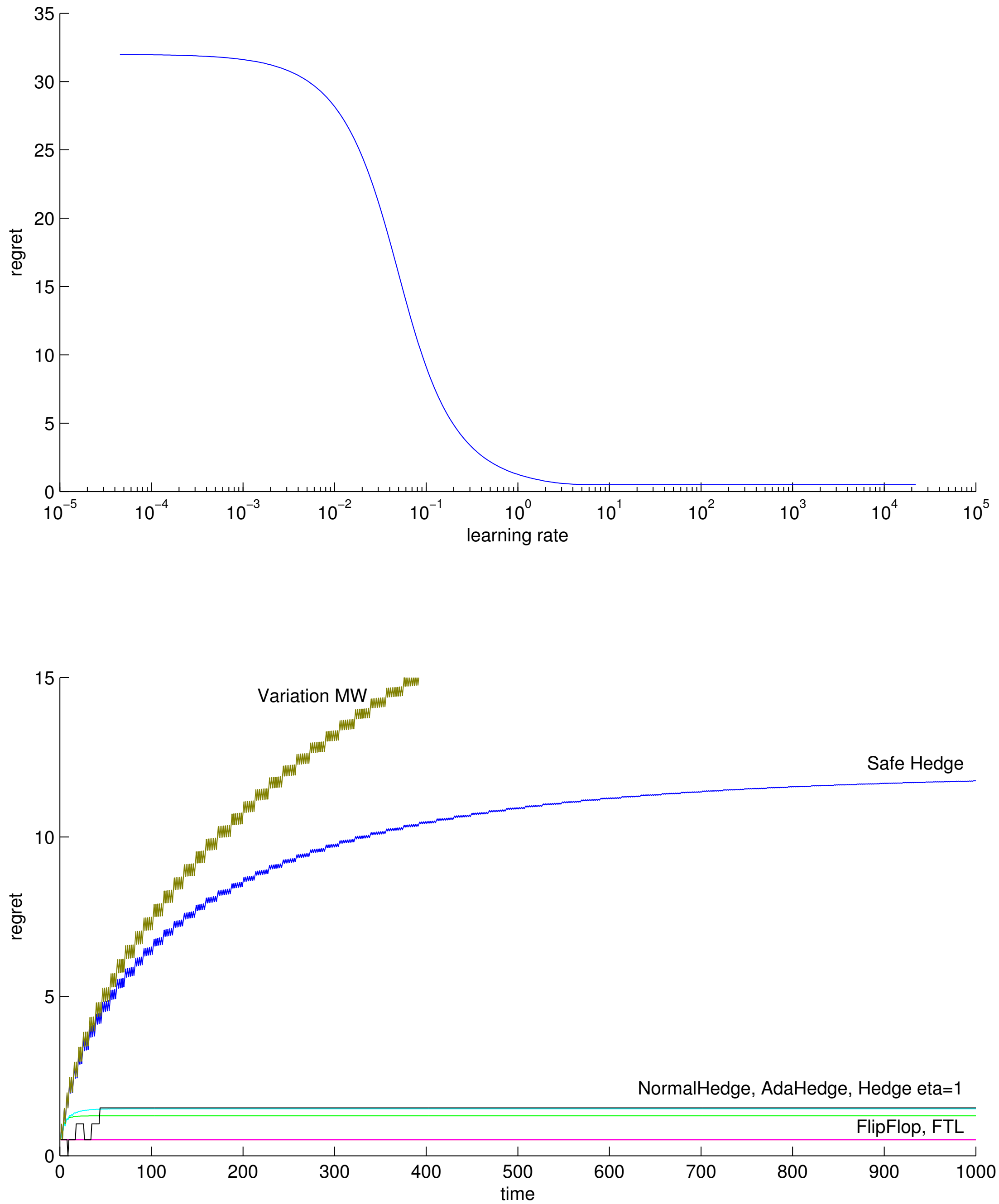


Figure 6: Hedge regret for Experiment 4 (weights do concentrate in AdaHedge)

for which they can provide no guarantees. A Universal Hedge algorithm would adapt to the learning rate that is optimal with hindsight. FlipFlop is a first step in this direction. Indeed, it already has some of the properties of such an ideal algorithm: under some conditions we can show that if Hedge achieves bounded regret using *any* learning rate, then FTL, and therefore FlipFlop, also achieves bounded regret:

Theorem 18 *Fix any $\eta > 0$. For $K = 2$ experts with losses in $\{0, 1\}$ we have*

$$\mathcal{R}^{(\eta)} \text{ is bounded } \Rightarrow \mathcal{R}^{\mathrm{ftl}} \text{ is bounded } \Rightarrow \mathcal{R}^{\mathrm{ff}} \text{ is bounded.}$$

The proof is in Appendix B. While the second implication remains valid for more experts and other losses, we currently do not know if the first implication continues to hold as well.

6.2 The Big Picture

Broadly speaking, a "learning rate" is any single scalar parameter controlling the relative weight of the data and a prior regularization term in a learning task. Such learning rates pop up in batch settings as diverse as L_1/L_2-regularized regression such as Lasso and Ridge, standard Bayesian nonparametric and PAC-Bayesian inference (Zhang, 2006; Audibert, 2004; Catoni, 2007), and—as in this paper—in sequential prediction. All the applications just mentioned can formally be seen as variants of Bayesian inference: Bayesian MAP in Lasso and Ridge, randomized drawing from the posterior ("Gibbs sampling") in the PAC-Bayesian setting and in the Hedge setting. Moreover, in each of these applications, selecting the appropriate learning rate is nontrivial: simply adding the learning rate as another parameter and putting a Bayesian prior on it can lead to very bad results (Grünwald and Langford, 2007). An ideal method for adapting the learning rate would work in all such applications. In addition to the FlipFlop algorithm described here, we currently have methods that are guaranteed to work for several PAC-Bayesian style stochastic settings (Grünwald, 2011, 2012). It is encouraging that all these methods are based on the same, apparently fundamental, quantity, the *mixability gap* as defined before Lemma 1: they all employ different techniques to ensure a learning rate under which the posterior is concentrated and hence the mixability gap is small. This gives some hope that the approach can be taken even further. To give but one example, the "Safe Bayesian" method of Grünwald (2012) uses essentially the same technique as Devaine et al. (2013), with an additional online-to-batch conversion step. Grünwald (2012) proves that this approach adapts to the optimal learning rate in an i.i.d. stochastic setting with arbitrary (countably or uncountably infinite) sets of "experts" (predictors); in contrast, AdaHedge and FlipFlop in the form presented in this paper are suitable for a worst-case setting with a finite set of experts. This raises, of course, the question of whether either the Safe Bayesian method can be extended to the worst-case setting (which would imply formal guarantees for the method of Devaine et al. 2013), or the FlipFlop algorithm can be extended to the setting with infinitely many experts.

Thus, we have two major, interrelated questions for future work: first, as explained in Section 6.1, we would like to be able to compete with all η in some set that contains a whole range rather than just two values. Second, we would like to compete with the best η in a setting with a countably infinite or even uncountable number of experts equipped with an arbitrary prior distribution.

A third question for future work is whether our methods can be extended beyond the standard worst-case Hedge setting and the stochastic i.i.d. setting. A particularly intriguing (and, as initial research suggests, nontrivial) question is whether AdaHedge and FlipFlop can be adapted to settings with limited feedback such as the adversarial *bandit setting* (Cesa-Bianchi and Lugosi, 2006). We would also like to extend our approach to the Hedge-based strategies for combinatorial decision domains like *Component Hedge* by Koolen et al. (2010), and for matrix-valued predictions like those by Tsuda et al. (2005).

Acknowledgments

We would like to thank Wojciech Kotłowski, Gilles Stoltz and two anonymous referees for critical feedback. This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778 and by NWO Rubicon grants 680-50-1010 and 680-50-1112.

Appendix A. Proof of Lemma 1

The result for $\eta = \infty$ follows from $\eta < \infty$ as a limiting case, so we may assume without loss of generality that $\eta < \infty$. Then $m_t \leq h_t$ is obtained by using Jensen's inequality to move the logarithm inside the expectation, and $m_t \geq \ell_t^-$ and $h_t \leq \ell_t^+$ follow by bounding all losses by their minimal and maximal values, respectively. The next two items are analogues of similar basic results in Bayesian probability. Item 2 generalizes the chain rule of probability $\Pr(x_1, \ldots, x_T) = \prod_{t=1}^T \Pr(x_t \mid x_1, \ldots, x_{t-1})$:

$$M = -\frac{1}{\eta} \ln \prod_{t=1}^{T} \frac{\boldsymbol{w}_1 \cdot e^{-\eta \boldsymbol{L}_t}}{\boldsymbol{w}_1 \cdot e^{-\eta \boldsymbol{L}_{t-1}}} = -\frac{1}{\eta} \ln(\boldsymbol{w}_1 \cdot e^{-\eta \boldsymbol{L}}).$$

For the third item, use item 2 to write

$$M = -\frac{1}{\eta} \ln \left(\sum_k w_{1,k} e^{-\eta L_{T,k}} \right).$$

The lower bound is obtained by bounding all $L_{T,k}$ from below by L^*; for the upper bound we drop all terms in the sum except for the term corresponding to the best expert and use $w_{1,k} = 1/K$.

For the last item, let $0 < \eta < \gamma$ be any two learning rates. Then Jensen's inequality gives

$$-\frac{1}{\eta} \ln \boldsymbol{w}_1 \cdot e^{-\eta \boldsymbol{L}} = -\frac{1}{\eta} \ln \boldsymbol{w}_1 \cdot \left(e^{-\gamma \boldsymbol{L}}\right)^{\eta/\gamma} \geq -\frac{1}{\eta} \ln \left(\boldsymbol{w}_1 \cdot e^{-\gamma \boldsymbol{L}}\right)^{\eta/\gamma} = -\frac{1}{\gamma} \ln \boldsymbol{w}_1 \cdot e^{-\gamma \boldsymbol{L}}.$$

This completes the proof. ■

Appendix B. Proof of Theorem 18

The second implication follows from Theorem 15, so we only need to prove the first implication. To this end, consider any infinite sequence of losses on which FTL has unbounded regret. We will argue that Hedge with fixed η must have unbounded regret as well.

Our argument is based on finding an infinite subsequence of the losses on which (a) the regret for Hedge with fixed η is at most as large as on the original sequence of losses; and (b) the regret for Hedge is infinite.

To construct this subsequence, first remove all trials t such that $\ell_{t,1} = \ell_{t,2}$ (that is, both experts suffer the same loss), as these trials do not change the regret of either FTL or Hedge, nor their behaviour on any of the other rounds.

Next, we will selectively remove certain local extrema. We call a pair of two consecutive trials $(t, t+1)$ a *local extremum* if the losses in these trials are opposite: either $\boldsymbol{\ell}_t = (0,1)$ and $\boldsymbol{\ell}_{t+1} = (1,0)$ or vice versa. Removing any local extremum will only decrease the regret for Hedge, as may be seen as follows.

We observe that removing a local extremum will not change the cumulative losses of the experts or the behaviour of Hedge on other rounds, so it suffices to consider only the regret incurred on rounds t and $t+1$ themselves. By symmetry it is further sufficient to consider the case that $\boldsymbol{\ell}_t = (0,1)$ and $\boldsymbol{\ell}_{t+1} = (1,0)$. Then, over trials t and $t+1$, the individual experts both suffer loss 1, and for Hedge the loss is $h_t + h_{t+1} = \boldsymbol{w}_t \cdot \boldsymbol{\ell}_t + \boldsymbol{w}_{t+1} \cdot \boldsymbol{\ell}_{t+1} = w_{t,2} + w_{t+1,1}$. Now, since the loss received by expert 1 in round t was less than that of expert 2, some weight shifts to the first expert: we must have $w_{t+1,1} > w_{t,1}$. Substitution gives $h_t + h_{t+1} > w_{t,1} + w_{t,2} = 1$. Thus, Hedge suffers more loss in these two rounds than whichever expert turns out to be best in hindsight, and it follows that removing trials t and $t+1$ will only decrease its regret (by an amount that depends only on η).

We proceed to select the local extrema to remove. To this end, let $d_t = L_{t,2} - L_{t,1}$ denote the difference in cumulative loss between the experts after t trials, and observe that removal of a local extremum at $(t, t+1)$ will simply remove the elements d_t and d_{t+1} from the sequence $d_1, d_2, \ldots$ while leaving the other elements of the sequence unchanged. We will remove local extrema in a way that leads to an infinite subsequence of losses such that

$$d_1, d_2, d_3, d_4, d_5, \ldots = \pm 1,\ 0, \pm 1,\ 0, \pm 1, \ldots \tag{26}$$

In this subsequence, every two consecutive trials still constitute a local extremum, on which Hedge incurs a certain fixed positive regret. Consequently, the Hedge regret $\mathcal{R}_t$ grows linearly in t and is therefore unbounded.

If the losses already satisfy (26), we are done. If not, then observe that there can only be a leader change at time $t+1$ in the sense of Lemma 10 when $d_t = 0$. Since the FTL regret is bounded by the number of leader changes (Lemma 10), and since FTL was assumed to have infinite regret, there must therefore be an infinite number of trials t such that $d_t = 0$. We will remove local extrema in a way that preserves this property. In addition, we must have $|d_{t+1} - d_t| = 1$ for all t, because $d_{t+1} = d_t$ would imply that $\ell_{t+1,1} = \ell_{t+1,2}$ and we have already removed such trials. This second property is automatically preserved regardless of which trials we remove.

If the losses do not yet satisfy (26), there must be a first trial u with $|d_u| \geq 2$. Since there are infinitely many t with $d_t = 0$, there must then also be a first trial $w > u$ with $d_w = 0$. Now choose any $v \in [u, w)$ so that $|d_v| = \max_{t \in [u,w]} |d_t|$ maximizes the discrepancy between the cumulative losses of the experts. Since v attains the maximum and $|d_{t+1} - d_t| = 1$ for all t as mentioned above, we have $|d_{v+1}| = |d_v| - 1$, so that $(v, v+1)$ must be a local extremum, and this is the local extremum we remove. Since $|d_v| \geq |d_u| \geq 2$, we also have $|d_{v+1}| \geq 1$, so that this does not remove any of the trials in which $d_t = 0$. Repetition of this process will eventually lead to $v = u$, so that trial u is removed. Given any T, the process may therefore be repeated until $|d_t| \leq 1$ for all $t \leq T$. As $|d_{t+1} - d_t| = 1$ for all t, we then match (26) for the first T trials. Hence by letting T go to infinity we obtain the desired result. ■

References

Jean-Yves Audibert. *PAC-Bayesian statistical learning theory.* PhD thesis, Université Paris VI, 2004.

Peter Auer, Nicolò Cesa-Bianchi, and Claudio Gentile. Adaptive and self-confident on-line learning algorithms. *Journal of Computer and System Sciences*, 64:48–75, 2002.

Olivier Catoni. *PAC-Bayesian Supervised Classification.* Lecture Notes-Monograph Series. IMS, 2007.

Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games.* Cambridge University Press, 2006.

Nicolò Cesa-Bianchi, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice. *Journal of the ACM*, 44(3):427–485, 1997.

Nicolò Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. Improved second-order bounds for prediction with expert advice. *Machine Learning*, 66(2/3):321–352, 2007.

Kamalika Chaudhuri, Yoav Freund, and Daniel Hsu. A parameter-free hedging algorithm. In *Advances in Neural Information Processing Systems 22 (NIPS 2009)*, pages 297–305, 2009.

Alexey V. Chernov and Vladimir Vovk. Prediction with advice of unknown number of experts. In Peter Grünwald and Peter Spirtes, editors, *UAI*, pages 117–125. AUAI Press, 2010.

Marie Devaine, Pierre Gaillard, Yannig Goude, and Gilles 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, February 2013.

Yoav Freund and Robert 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.

Yoav Freund and Robert E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29:79–103, 1999.

Sébastien Gerchinovitz. *Prédiction de suites individuelles et cadre statistique classique: étude de quelques liens autour de la régression parcimonieuse et des techniques d'agrégation.* PhD thesis, Université Paris-Sud, 2011.

Peter Grünwald. Safe learning: bridging the gap between Bayes, MDL and statistical learning theory via empirical convexity. In *Proceedings of the 24th International Conference on Learning Theory (COLT 2011)*, pages 551–573, 2011.

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

Peter Grünwald and John Langford. Suboptimal behavior of Bayes and MDL in classification under misspecification. *Machine Learning*, 66(2-3):119–149, 2007. DOI 10.1007/s10994-007-0716-7.

László Györfi and György Ottucsák. Sequential prediction of unbounded stationary time series. *IEEE Transactions on Information Theory*, 53(5):1866–1872, 2007.

Elad Hazan and Satyen Kale. Extracting certainty from uncertainty: Regret bounded by variation in costs. In *Proceedings of the 21st Annual Conference on Learning Theory (COLT)*, pages 57–67, 2008.

Marcus Hutter and Jan Poland. Adaptive online prediction by following the perturbed leader. *Journal of Machine Learning Research*, 6:639–660, 2005.

Adam Kalai and Santosh Vempala. Efficient algorithms for online decision. In *Proceedings of the 16st Annual Conference on Learning Theory (COLT)*, pages 506–521, 2003.

Yuri Kalnishkan and Michael V. Vyugin. The weak aggregating algorithm and weak mixability. In *Proceedings of the 18th Annual Conference on Learning Theory (COLT)*, pages 188–203, 2005.

Wouter M. Koolen, Manfred K. Warmuth, and Jyrki Kivinen. Hedging structured concepts. In A.T. Kalai and M. Mohri, editors, *Proceedings of the 23rd Annual Conference on Learning Theory (COLT 2010)*, pages 93–105, 2010.

Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

Koji Tsuda, Gunnar Rätsch, and Manfred K. Warmuth. Matrix exponentiated gradient updates for on-line learning and Bregman projection. *Journal of Machine Learning Research*, 6:995–1018, 2005.

Tim van Erven, Peter Grünwald, Wouter M. Koolen, and Steven de Rooij. Adaptive hedge. In *Advances in Neural Information Processing Systems 24 (NIPS 2011)*, pages 1656–1664, 2011.

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

Vladimir Vovk. Competitive on-line statistics. *International Statistical Review*, 69(2):213–248, 2001.

Vladimir Vovk, Akimichi Takemura, and Glenn Shafer. Defensive forecasting. In *Proceedings of AISTATS 2005*, 2005. Archive version available at `http://www.vovk.net/df`.

Tong Zhang. Information theoretical upper and lower bounds for statistical estimation. *IEEE Transactions on Information Theory*, 52(4):1307–1321, 2006.