



How to Better Use Expert Advice

RANI YAROSHINSKY
RAN EL-YANIV

Department of Computer Science, Technion - Israel Institute of Technology

yaroshin@netvision.net.il
rani@cs.technion.ac.il

STEVEN S. SEIDEN

Department of Computer Science, Louisiana State University

Editor: Lisa Hellerstein

Dedicated to the Memory of Steve Seiden

Abstract. This work is concerned with online learning from expert advice. Extensive work on this problem generated numerous “expert advice algorithms” whose total loss is provably bounded above in terms of the loss incurred by the best expert in hindsight. Such algorithms were devised for various problem variants corresponding to various loss functions. For some loss functions, such as the square, Hellinger and entropy losses, *optimal* algorithms are known. However, for two of the most widely used loss functions, namely the 0/1 and absolute loss, there are still gaps between the known lower and upper bounds.

In this paper we present two new expert advice algorithms and prove for them the best known 0/1 and absolute loss bounds. Given an expert advice algorithm ALG , the goal is to form an upper bound on the *regret* $L_{ALG} - L^*$ of ALG , where L_{ALG} is the loss of ALG and L^* is the loss of the best expert in hindsight. Typically, regret bounds of a “canonical form” $C \cdot \sqrt{L^* \ln N}$ are sought where N is the number of experts and C is a constant. So far, the best known constant for the absolute loss function is $C = 2.83$, which is achieved by the recent IAWM algorithm of Auer et al. (2002). For the 0/1 loss function no bounds of this canonical form are known and the best known regret bound is $L_{ALG} - L^* \leq L^* + C_1 \ln N + C_2 \sqrt{L^* \ln N + \frac{\epsilon}{4} \ln^2 N}$, where $C_1 = e - 2$ and $C_2 = 2\sqrt{e}$. This bound is achieved by a “P-norm” algorithm of Gentile and Littlestone (1999). Our first algorithm is a randomized extension of the “guess and double” algorithm of Cesa-Bianchi et al. (1997). While the guess and double algorithm achieves a canonical regret bound with $C = 3.32$, the *expected* regret of our randomized algorithm is canonically bounded with $C = 2.49$ for the absolute loss function. The algorithm utilizes one random choice at the start of the game. Like the deterministic guess and double algorithm, a deficiency of our algorithm is that it occasionally restarts itself and therefore “forgets” what it learned. Our second algorithm does not forget and enjoys the best known asymptotic performance guarantees for both the absolute and 0/1 loss functions. Specifically, in the case of the absolute loss, our algorithm is canonically bounded with C approaching $\sqrt{2}$ and in the case of the 0/1 loss, with C approaching $3/\sqrt{2} \approx 2.12$. In the 0/1 loss case the algorithm is randomized and the bound is on the expected regret.

Keywords: online learning, online prediction, learning from expert advice

1. Introduction

In the *online prediction using expert advice* problem (DeSantis, Markowsky, & Wegman, 1988; Vovk, 1990; Littlestone & Warmuth, 1994; Blum, 1996), a learning algorithm is required, at the start of each *trial*, to predict an unknown outcome, which may be discrete or

real. The algorithm's input for generating this prediction is the advice of each of N experts where an expert advice is a (discrete or real) prediction of the outcome. By the end of the trial, after making a prediction, the algorithm incurs some loss which measures, via some loss function, the discrepancy between its prediction and the true outcome. Then a new trial starts, etc. The general goal is to devise an expert advice algorithm that predicts as best as possible by utilizing experts' opinions.

Without any (statistical) assumptions on expert predictions and in particular, within an adversarial context, it is impossible to ensure overall quality of an expert advice algorithm. However, it is possible to ensure performance quality relative to some offline "comparison class" of prediction algorithms that also utilize the expert advice. This methodology of worst case analysis of online algorithms has been used in various disciplines such as statistics (Chernoff, 1954; Milnor, 1954), where it is called *regret analysis*, computer science (Sleator & Tarjan, 1985; Borodin & El-Yaniv, 1998), where it is called *competitive analysis*, game theory (Blackwell, 1956) and information theory (Cover, 1991; Feder, Merhav, & Gutman, 1992). We use this approach here. The problem of online prediction using expert advice falls within the more general context of online classification and regression, whereby a learning algorithm needs to predict the label or value assigned to each item in a sequence of feature vectors that is sequentially revealed.

Much of the work on learning to predict by combining expert advice focused on the simplest comparison class where we attempt to predict as well as the best expert in hindsight and measure the quality of the expert advice algorithm via *regret*, defined to be the difference between the total loss of the expert advice algorithm and the total loss of the best expert. This game has been extensively studied with respect to various loss functions. To a large extent this extensive focus on the expert advice problem can be attributed to the wide range of applications of expert advice algorithms. Some of these applications are related to agnostic learning (Cesa-Bianchi et al., 1997), boosting (Freund & Schapire, 1997), pruning of decision trees (Helmbold & Schapire, 1995), Metrical Task Systems (Blum & Burch, 1997), online paging (Blum, Burch, & Kalai, 1989), adaptive disk spin-down for mobile computing (Helmbold, Long, & Sherrod, 1996), and repeated game playing (Freund & Schapire, 1997). Moreover, there is evidence that expert advice algorithms have practical significance, and as noted by Blum and others, these algorithms have "exceptionally good performance in the face of irrelevant features, noise, or a target function changing with time" (Blum, 1997).

Despite the conceptual simplicity of the expert advice problem and much research attention devoted to it, a number of basic questions have remained open. In particular, for two of the most widely used loss functions, namely the 0/1 and absolute loss, there are still gaps between the known lower and upper bounds. In this paper we present two new expert advice algorithms and prove for them the best known 0/1 and absolute loss bounds.

Given an expert advice algorithm ALG, the goal is to form an upper bound on the *regret* $L_{\text{ALG}} - L^*$ of ALG, where L_{ALG} is the loss of ALG and L^* is the loss of the best expert in hindsight. Typically, regret bounds of a "canonical form" $C \cdot \sqrt{L^* \ln N}$ are sought where N is the number of experts and C is a constant. So far, the best known constant for the absolute loss function is $C = 2.83$, which is achieved by the recent IAWM algorithm of Auer, Cesa-Bianchi, and Gentile (2002). For the 0/1 loss function no bounds with this

canonical form are known and the best known regret bound is $L_{\text{ALG}} - L^* \leq L^* + C_1 \ln N + C_2 \sqrt{L^* \ln N + \frac{\epsilon}{4} \ln^2 N}$ where $C_1 = e - 2$ and $C_2 = 2\sqrt{e}$. This bound is achieved by a ‘‘P-norm’’ algorithm of Gentile and Littlestone (1999).

Our first algorithm is a simple randomized extension of the ‘‘guess and double’’ algorithm of Cesa-Bianchi et al. (1997). While the guess and double algorithm achieves a canonical regret bound with $C = 3.32$, the *expected* regret of our randomized algorithm is canonically bounded with $C = 2.49$ for the absolute loss function. The algorithm utilizes one random choice at the start of the game. Like the deterministic guess and double algorithm, a deficiency of our algorithm is that it occasionally restarts itself and therefore ‘‘forgets’’ what it learned. Our second algorithm is based on a new ‘‘weight shift’’ technique and does not forget. The algorithm enjoys the best known asymptotic performance guarantees for both the absolute and 0/1 loss functions. Specifically, in the case of the absolute loss, our algorithm is canonically bounded with C approaching $\sqrt{2}$ and in the case of the 0/1 loss, with C approaching $3/\sqrt{2} \approx 2.12$. In the 0/1 loss case the algorithm is randomized and the bound is on the expected regret. Both our algorithms are presented as new ‘‘master’’ prediction algorithms which operate known prediction routines that are simple extensions of the well-known routine of Vovk (1990).

2. Problem definition, related work and our results

Consider the following prediction game between an online algorithm ALG and an adversary. The game consists of repeated *trials*. At the start of each trial t , ALG receives the advice of each of N experts, $\xi_{1,t}, \dots, \xi_{N,t}$, where $\xi_{j,t} \in \mathcal{D}$ and \mathcal{D} is some domain. The algorithm is supposed to utilize any kind of expert predictions and assumes no knowledge on how these expert predictions are formed. Hence, for analysis we assume that these predictions are controlled by the adversary. Given the vector of expert predictions the algorithm generates its own prediction $\hat{y}_t \in \mathcal{D}$. The adversary then generates the true outcome $y_t \in \mathcal{D}$ and ALG incurs a loss, $L(\hat{y}_t, y_t)$, where $L : \mathcal{D} \times \mathcal{D} \rightarrow \mathbb{R}^+$ is some loss function. For each outcome sequence $\mathbf{y} = y_1, \dots, y_\ell$ consisting of ℓ outcomes, the *total loss* of ALG is

$$L_{\text{ALG}}(\mathbf{y}) = \sum_{t=1}^{\ell} L(\hat{y}_t, y_t). \quad (1)$$

The total loss of expert j is $L_j(\mathbf{y}) = \sum_{t=1}^{\ell} L(\xi_{j,t}, y_t)$. The *regret* of ALG is defined to be $L_{\text{ALG}}(\mathbf{y}) - \min_j L_j(\mathbf{y})$. The adversary attempts to maximize the regret by choosing the outcome sequence (including its length ℓ) and expert predictions.

Different problem variants are distinguished by choices of the loss function $L(\cdot, \cdot)$ and the type of outcome (and prediction) domain \mathcal{D} . These are typically determined by the application at hand. Commonly used loss functions that have been considered in the literature are the 0/1-loss (which counts 1 for any error), the absolute loss, ($L(x, y) = |x - y|$), the square loss ($L(x, y) = (x - y)^2$), the entropy loss (also called ‘‘log-loss’’) and Hellinger loss (which are given in Table 1). The simplest choice for \mathcal{D} is $\mathcal{D} = \{0, 1\}$, which is useful for binary predictions. For (bounded) regression problems the interval $\mathcal{D} = [0, 1]$ (or any

Table 1. c_L coefficients for three “realizable” loss functions.

Loss function	$L(y, \hat{y})$	c_L coefficient
Square	$(y - \hat{y})^2$	$\frac{1}{2}$
Hellinger	$\frac{1}{2}((\sqrt{1-y} - \sqrt{1-\hat{y}})^2 + (\sqrt{y} - \sqrt{\hat{y}})^2)$	$\frac{1}{\sqrt{2}}$
Relative entropy (logarithmic)	$(y \ln \frac{y}{\hat{y}} + (1-y) \ln \frac{1-y}{1-\hat{y}})$	1

finite real interval) can be used. In some classification applications \mathcal{D} is required to be an arbitrary set of indices (or “alphabet”). Clearly, the choice of the loss function should be made in accordance with the choice of \mathcal{D} . In this paper we present algorithms that can deal with any real interval domain while measuring absolute loss, and with any finite alphabet domain together with the 0/1 loss.

The algorithm ALG can be deterministic or randomized. The performance of randomized algorithms is measured by their expected regret

$$E[L_{\text{ALG}}(\mathbf{y}) - \min_j L_j(\mathbf{y})] = E[L_{\text{ALG}}(\mathbf{y})] - \min_j L_j(\mathbf{y}),$$

where the expectation is taken with respect to the random choices made by ALG.¹

We note that any deterministic absolute loss algorithm ALG for $\mathcal{D} = [0, 1]$ can be trivially translated to a 0/1 loss *randomized* algorithm ALG' for $\mathcal{D} = \{0, 1\}$ by simply using any real prediction $\hat{y} \in [0, 1]$ as a \hat{y} -biased coin, thus generating a discrete randomized binary prediction. It is not hard to see that any (deterministic) regret bound for ALG can be directly interpreted as an expected regret bound for ALG'. However, currently there is no general method for translating a deterministic absolute regret bound to an expected 0/1 regret bound when \mathcal{D} is a finite alphabet with cardinality greater than two.

2.1. The Weighted Majority (WM) algorithm

The Weighted Majority (WM) algorithm (Vovk, 1990; Littlestone & Warmuth, 1994) forms the basis of many expert advice algorithms. WM maintains a weight vector for the set of experts, and predicts the outcome via a weighted majority vote between the experts. WM online learns this weight vector by “punishing” erroneous experts. The punishment is done by multiplying an expert weight by an *update factor* (or *learning rate*) $0 < \beta < 1$. Several variants of WM are presented in Littlestone and Warmuth (1994):

- **WM General** (WVG): updates expert weights only when its own binary prediction is incorrect. It can be shown that its total loss is bounded above by $B/(\ln \frac{2}{1+\beta})$, where $B = \ln N + L^* \ln \frac{1}{\beta}$ and $L^* = \min_j L_j(\mathbf{y})$. A pseudo code for WVG is given in figure 1.
- **WM Randomized** (WVR): randomly chooses a single expert’s prediction with probability proportional to its weight. A pseudo code for WVR is given in figure 2.
- **WM Continuous** (WVC): generates a real prediction. The pseudo code for WVC is similar to the one given for the VPREDICT routine presented later in figure 6.

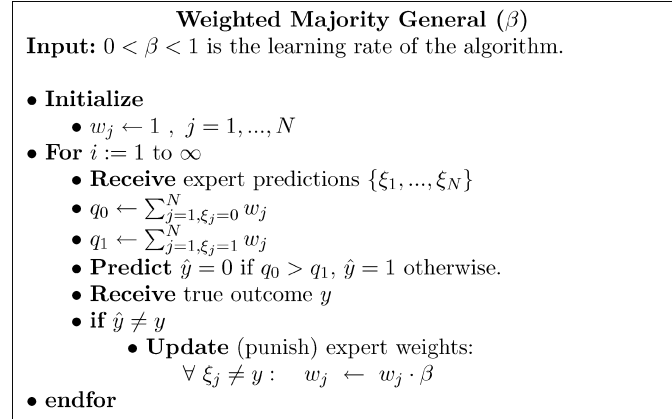


Figure 1. The Weighted Majority General (WMG) algorithm.

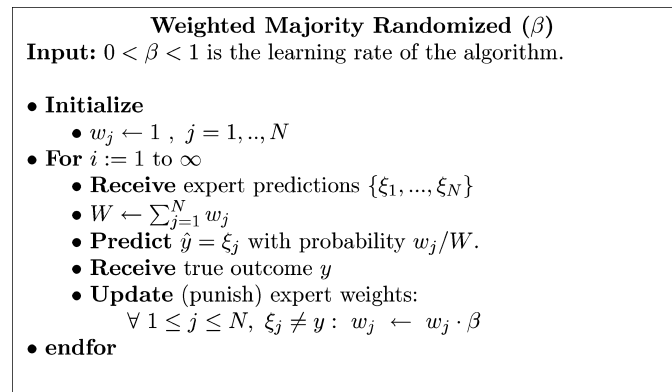


Figure 2. The Weighted Majority Randomized (WMR) algorithm of Littlestone and Warmuth.

Both WMC and WMR attain a regret bound of $B/(1 - \beta)$ (where in the case of WMR L^* is an expected value). A better (absolute loss) regret bound of $B/(2 \ln \frac{2}{1+\beta})$ for WM can be achieved by considering particular constrained functions for computing the weight updates and the prediction (Vovk, 1990).

2.2. Vovk's (c, η) -realizable loss functions and the "generic algorithm"

Here we briefly mention some of the ideas originally presented in Vovk's "aggregating algorithms" paper (Vovk, 1990) and later discussed, simplified and generalized in Haussler, Kivinen, and Warmuth (1998). The results in these papers show that performance guarantees of the form $L_A - L^* \leq c_1 \cdot \sqrt{L^* \cdot \ln N} + c_2 \ln N$ (where c_1 and c_2 are constants) are achievable for certain loss functions. In the sequel we term bounds of this form "canonical".

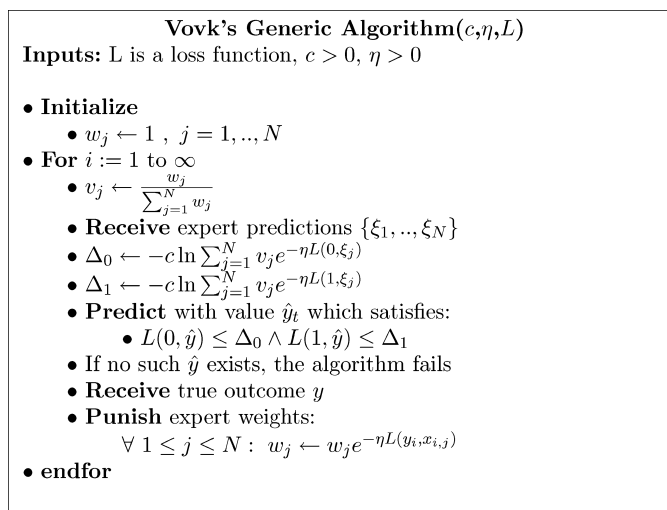


Figure 3. Vovk's "generic" algorithm.

The discussion here follows the analysis presented in Haussler, Kivinen, and Warmuth (1998).

Let $L(y, \hat{y})$ be a loss function, and define $L_0(\hat{y}) = L(0, \hat{y})$ and $L_1(\hat{y}) = L(1, \hat{y})$. Given a loss function which is twice differentiable, the following functions are defined (Haussler, Kivinen, & Warmuth, 1998, p. 1909):

$$S(z) = L'_0(z)L''_1(z) - L'_1(z)L''_0(z);$$

$$R(z) = \frac{L'_0(z)L'_1(z)^2 - L'_1(z)L'_0(z)^2}{S(z)}.$$

Set $c_L = \sup_{0 < z < 1} R(z)$. Figure 3 presents Vovk's "generic" algorithm. Exactly as previously mentioned WM algorithms, the generic algorithm maintains weights w_j , $j = 1, \dots, N$, that reflect expert performance so far. Unlike other algorithms discussed in this paper, the "generic" algorithm receives as input a loss function, which automatically determines the prediction function. This property makes the algorithm more generic. Considering the $e^{-\eta}$ term appearing in the "punishment" step of the generic algorithm as a learning rate β would give us the same form of WM-type algorithms as presented earlier. If the algorithm does not fail (in the prediction step), given a loss function L and the parameters c and η , we say that the loss function L is " (c, η) -realizable". Given a loss function L with finite c_L , the generic algorithm with parameters $c = c_L$, $\eta = \frac{1}{c_L}$ achieves a tight regret bound of $L_A(\mathbf{y}) - L^* \leq c_L \ln N$. Table 1 shows the c_L coefficient for three common loss functions. For some loss functions, like the absolute-loss, c_L is not defined, and therefore the following weaker form of a regret bound is sought

$$L_A(\mathbf{y}) - L^* \leq +a \cdot \sqrt{L^* \cdot \ln N} + b \ln N.$$

For instance, this regret bound of the generic algorithm applied with the absolute loss function is optimized by using learning rate $\eta = \sqrt{\frac{c(\ln N)}{L^*}}$, for any positive c , which results in a regret bound of $L_T - L^* \leq 2 \cdot \sqrt{L^* \cdot c \cdot \ln N} + c \ln N$.² Thus, there is a trade-off between choices of a and b , where smaller b values imply larger a values and vice versa. The “canonical” bound is obtained by taking $b = 1$, and allows for the comparison of different algorithms by their a -coefficient.³

2.3. Algorithm P^* (DWM)

In all the above absolute loss variants of WM (Sections 2.1 and 2.2) the choice of the learning rate β , which can significantly affect the algorithms’ behavior and performance guarantees, is left open.⁴ For the case where the game is of *known duration* (i.e. ℓ is known to A at the start of the game), Cesa-Bianchi et al. (1997) show an optimal minimax strategy for the online player whose regret is bounded above by $(1 + o(1))\sqrt{(\ell \cdot \ln N)/2}$. They also show a strategy for the adversary, which extracts a lower bound of $(1 - o(1))\sqrt{(\ell \ln N)/2}$ on the regret of any expert advice algorithm. For the case of *unknown* duration they propose to operate WM algorithms in phases using a standard “guess and double” technique. In this scheme, the “master” algorithm, called P^* , operates a WMC-type routine (whose pseudo code is similar to the one appearing in figure 6) with an “optimal” learning rate that is computed based on a guess about the total loss of the best expert. In this paper we sometimes call the resulting algorithm “Doubling Weighted Majority (DWM)”. Once the loss of the best expert exceeds this guess, the master algorithm multiplies the guess by a fixed factor⁵ and starts a new phase by initializing all expert weights. The asymptotic growth rate of the regret, as L^* grows, is shown to be bounded above by $L^* + (3.32 + o(1))\sqrt{L^* \ln N}$.⁶

2.4. Algorithm $IAWM$

The major deficiency of the doubling algorithm P^* is the initialization it performs at the start of each phase where it “forgets” all the information it learned about the experts. Recently, Auer, Cesa-Bianchi, and Gentile (2002) proposed a new “self-confident” algorithm, called $IAWM$ (stands for Incrementally Adaptive Weighted Majority), which incrementally updates the learning rate and does not forget expert weights. The (absolute loss) regret of $IAWM$ is bounded above by $(2.83 + o(1))\sqrt{L^* \ln N}$.⁷ The algorithm’s pseudo code appears in figure 4. So far, this bound has been the best known for this setting.

2.5. P -NORM algorithms

Gentile and Littlestone (1999) considered “ P -NORM algorithms” in the context of expert advice.⁸ They show that for the 0/1-loss, the mistake bound of P -NORM algorithms is bounded above by $2 \cdot L^* + (e - 2) \cdot \ln N + 2\sqrt{e} \cdot \sqrt{L^* \ln N} + \frac{e}{4} \ln^2 N$. A regret bound of $L^* + (e - 2) \cdot \ln N + 2\sqrt{e} \cdot \sqrt{L^* \ln N} + \frac{e}{4} \ln^2 N$ can be directly obtained. However, note that in this regret bound there is an extra L^* term (in particular this bound is not in the canonical form).

Algorithm *iAWM*

- **Initialize**
 - $w_{j,0} \leftarrow 1, L_{j,0} \leftarrow 0, j = 1, \dots, N$
- **For** $i := 1$ to ∞
 - **Calculate** :
 - $W_i \leftarrow \sum_{j=1}^N \alpha_t^{-L_{j,i-1}}$
 - $w_{j,i} \leftarrow \alpha_t^{-L_{j,i-1}} / W_i, j = 1, \dots, N$
 - $L_{i-1}^* \leftarrow \min_{1 \leq j \leq N} (L_{j,i-1})$
 - $\epsilon_i \leftarrow \min\{\frac{1}{4}, \sqrt{2 \frac{\ln N}{L_{i-1}^*}}\}$
 - $\alpha_i \leftarrow \frac{1}{1-\epsilon_i}$
 - **Receive** experts' predictions $\xi \in [0, 1]^N$
 - **Predict** $\hat{y} = \mathbf{w}_i \cdot \xi$
 - **Receive** true outcome $y \in [0, 1]$
 - **Punish** experts' weights:
 - $\forall 1 \leq j \leq N : L_{j,i} \leftarrow L_{j,i-1} + |y - \hat{y}|$
- **endfor**

Figure 4. The incrementally adaptive weighted majority algorithm (IAWM) of Auer et al.

2.6. Our results

Our first result is a simple randomized extension of the “guess and double” DWM algorithm of Cesa-Bianchi et al. (1997) (see Section 2.3). The new algorithm called “random doubling weighted majority” or RDWM for short, chooses one random real number in $[0, 1)$ at the start of the game and then operates deterministically. This random number controls the sequence of guesses it later forms, which should upper bound the loss of the best expert. These randomized guesses allow for a better absolute loss bound than that achieved by the deterministic DWM and reduces its canonical form constant from 3.32 to 2.49. However, the established bound is on the expected regret of RDWM. Like the deterministic DWM, our RDWM algorithm occasionally restarts itself and initializes expert weights. Therefore, it does not preserve what it learned.

Our main results are based on a new approach for operating the WMC, WMR and WMG prediction routines. In this approach the structure of the master algorithm is similar to that of DWM and is conceptually based on “guesses” on the loss of the best expert. However, when the current “guess” is violated (i.e. the loss of the best expert exceeds the guess), our new “weight-shift” technique “corrects” expert weights instead of initializing them. These corrections preserve the relative magnitude of expert weights in accordance with their cumulative losses. Thus, the master algorithm does not forget what it learned.

This technique is applicable in various settings and in particular in the absolute and 0/1 loss games. In the case of the absolute loss the regret of the proposed algorithm (based on WMC) is asymptotically (for large L^*) bounded above by $(\frac{2\alpha + \sqrt{\alpha+1}}{2\sqrt{2\alpha}} + o(1)) \cdot \sqrt{\ln N \cdot L^*}$, where $\alpha > 1$ is any real. The bound is minimized when $\alpha \rightarrow 1$ and approaches $(\sqrt{2} + o(1)) \cdot \sqrt{\ln N \cdot L^*}$. This bound improves on the best known bound (achieved by IAWM) by a factor of approximately 2. For the 0/1-loss the regret of our algorithm (now based on WMR)

Table 2. Summary of the best known bounds so far and our new bounds.

Algorithm	Reference	Loss function	Asymptotic regret “Canonical” bound
WMVP (WMR-based)	Our work	0/1	$(3/\sqrt{2} + o(1)) \cdot \sqrt{\ln N \cdot L^*}$ bounds the expected regret
P^*	(Cesa-Bianchi et al., 1997)	Absolute	$(3.32 + o(1))\sqrt{\ln N \cdot L^*}$
IAWM	(Auer et al., 2002)	Absolute	$(2.83 + o(1))\sqrt{\ln N \cdot L^*}$
RDWM	Our work	Absolute and 0/1	$(2.49 + o(1))\sqrt{\ln N \cdot L^*}$ bounds the expected regret
WMVP WMC-based	Our work	Absolute	$(\sqrt{2} + o(1)) \cdot \sqrt{\ln N \cdot L^*}$

is bounded above by $(\frac{\alpha + \sqrt{\alpha} + 1}{\sqrt{2\alpha}} + o(1)) \cdot \sqrt{\ln N \cdot L^*}$. Here again, the bound is minimized when $\alpha \rightarrow 1$ and in this case approaches $(3/\sqrt{2} + o(1)) \cdot \sqrt{\ln N \cdot L^*}$. This bound improves the bound attained by P-NORM algorithms. In particular, the bound we achieve is of the form $L_{\text{ALG}} - L^* \leq C \cdot \sqrt{\ln N \cdot L^*}$ with $C \approx 2.12$. In comparison, the best known bound so far, achieved by the P-NORM algorithms of Gentile and Littlestone (1999, see Section 2.5 for more details) is, $L_{\text{ALG}} - L^* \leq L^* + C_1 \ln N + C_2 \sqrt{L^* \ln N} + \frac{\epsilon}{4} \ln^2 N$, where $C_1 = e - 2$ and $C_2 = 2\sqrt{e}$. Note that our 0/1-loss result is based on the (randomized) WMR and therefore our performance guarantee in this case bounds the *expected* regret. Table 2 summarizes the best known *canonical form* bounds so far as well as our new bounds for the 0/1 and absolute loss functions.

3. The Random Doubling Weighted Majority (RDWM) algorithm

In this section we show how to use a simple “randomized guess and double” technique to improve the bound of the DWM doubling algorithm of Cesa-Bianchi et al. (1997, see Section 2.3). We call the resulting algorithm “randomized doubling weighted majority” or RDWM for short. The RDWM algorithm makes a single random choice at initialization and thereafter continues its execution deterministically very similarly to DWM.⁹

The master RDWM algorithm is given in figure 5. The RDWM algorithm uses the VPREDICT prediction routine given in figure 6. VPREDICT is essentially the P^* routine (which, as noted in Cesa-Bianchi et al., 1997, pp. 429–430, is motivated by a related algorithm proposed in Vovk, 1990).¹⁰

Remark. As in algorithm P^* of Cesa-Bianchi et al. (1997), the VPREDICT routine leaves open the choices of the prediction function $F_\beta(r_t)$ and the update constant determined by the function $U_\beta(\cdot)$. In figure 6 we specify the constraints on the prediction function F . These constraints are identical to those used by Algorithm P^* and any choice within the specified ranges will allow for the performance bounds we establish in this paper.

RDWM(α)

Input: α , any real ≥ 1 .

- **Initialize**
 - Pick a number δ uniformly at random from $[0, 1)$
 - Set $F_0 \leftarrow 0$
 - Set first round number $t \leftarrow 1$
- **For** $i := 1$ to ∞
 - Set upper bound guess on best expert loss:

$$F_i = \alpha^{\delta+i-1} \ln N$$

$$U = F_i - F_{i-1}$$
 - Calculate next learning rate:

$$\beta_i \leftarrow g(\sqrt{\frac{2 \ln N}{U}})$$
 where

$$g(z) = (1 + 2z + z^2 / \ln 2)^{-1}$$
 - Initialize weights and losses: $w_j \leftarrow 1, L_j = 0, j = 1, \dots, N$
 - Operate VPREDICT: $[t_i, \{w_j\}, \{L_j\}] \leftarrow \text{VPREDICT}(t, U, \beta_i, \{w_{i,j}\}, \{L_j\})$
 - Increment phase and round numbers: $i \leftarrow i + 1, t \leftarrow t_i + 1$
- **endfor**

Figure 5. The RDWM algorithm.

VPredict($t, T, \beta, \{w_j\}, \{L_j\}$)

Inputs: t : the next trial number; T : threshold on the loss of the best expert (that is, the VPREDICT routine will run as long as the best expert loss $\leq T$); β : an update factor; and $\{w_j\}, \{L_j\}$ are, respectively, the initial weights and losses of the experts. **Outputs:** termination trial number, the sets of updated expert weights $\{w_j\}$ and losses $\{L_j\}$

- **Repeat**
 - **Receive** t^{th} trial expert predictions, $\xi_{1,t}, \dots, \xi_{N,t}$
 - $r_t \leftarrow \frac{\sum_{i=1}^N w_i \xi_{i,t}}{\sum_{i=1}^N w_i}$
 - **Predict** $\hat{y}_t = F_\beta(r_t)$ (see below)
 - **Receive** the true t^{th} trial outcome y_t
 - **Update** expert cumulative losses:

$$L_{\xi_j} \leftarrow L_j + |\xi_{j,t} - y_t|, \quad j = 1, \dots, N$$
 - **Update** (punish) expert weights:

$$w_j \leftarrow w_j \cdot U_\beta(|\xi_{j,t} - y_t|) \quad j = 1, \dots, N$$
 (see below)
 - $t \leftarrow t + 1$
- **Until** $\min_j(L_j) \geq T$
- **Return** $t - 1, \{w_j\}, \{L_j\}$

$F_\beta(r)$ and $U_\beta(q)$ are functions that satisfy:

$$\forall 0 \leq r \leq 1, \quad 1 + \frac{\ln((1-r)\beta + r)}{2 \ln \frac{2}{1+\beta}} \leq F_\beta(r) \leq \frac{\ln(1-r+r\beta)}{2 \ln \frac{2}{1+\beta}} \quad (8)$$

$$\forall 0 \leq q \leq 1, \quad U_\beta(q) = \beta^q \quad (9)$$

Figure 6. The VPREDICT routine.

RDWM receives a parameter $\alpha \geq 1$ which will be optimized later. For real numbers $\alpha \geq 1$ and $\delta \in [0, 1)$ define

$$F_i = \begin{cases} 0, & \text{if } i = 0; \\ \alpha^{\delta+i-1} \ln N, & \text{otherwise.} \end{cases} \quad (2)$$

Before the first trial the algorithm picks δ uniformly at random from $[0, 1)$. The algorithm operates in a sequence of *phases*. The i th phase starts by invoking the VPREDICT using an upper bound guess $U = F_i - F_{i-1}$ on the loss of the best expert. Throughout the i th phase VPREDICT operates using an appropriate learning rate β_i , which is optimized as a function of the upper bound U using the techniques of Cesa-Bianchi et al. (1997). The i th phase continues as long as the loss of the best expert is smaller than or equal to U . When the i th phase ends, a new guess is formed, a new appropriate learning rate is calculated and then a new phase starts.

We now provide a regret bound for RDWM. Our analysis relies on the following theorem from Cesa-Bianchi et al. (1997), which gives a bound on the total loss of VPREDICT when applied with an upper bound on the loss of the best expert (with appropriate learning rate, as specified in Algorithm RDWM).

Theorem 3.1 (Cesa-Bianchi et al., 1997, Section 4.4.3). *Let \mathbf{y} be any outcome sequence and let U be an upper bound on the loss L^* of the best expert with respect to \mathbf{y} . Then, the total loss of VPREDICT when applied with the learning rate $g(\sqrt{\ln N/U})$ (where g is defined in Algorithm RDWM in figure 5), is bounded above by*

$$L_{\text{RDWM}}(\mathbf{y}) \leq L^* + \sqrt{U \ln N} + \frac{\ln N}{2 \ln 2}. \quad (3)$$

The following theorem provides a regret bound for RDWM as a function of its parameter α . Later on in Corollary 3.1 we optimize and find the α that minimizes the regret.

Theorem 3.2. *Let \mathbf{y} be any outcome sequence and let L^* be the loss of the best expert with respect to \mathbf{y} . Then, for any $\alpha \geq 1$,*

$$E[L_{\text{RDWM}}(\mathbf{y})] - L^* \leq \frac{2\sqrt{(\alpha-1)L^* \ln N}}{\ln \alpha} + O(\ln L^* \ln N),$$

where the expectation is with respect to the random choice made by RDWM.

Proof: Let T be the integer satisfying

$$\alpha^{T-1} \ln N \leq L^* \leq \alpha^T \ln N.$$

Let $x \in [0, 1)$ be the solution of

$$L^* = \alpha^{T+x-1} \ln N.$$

Thus,

$$T = \log_{\alpha}(L^*/\ln N) - x + 1. \quad (4)$$

Note that for $i \geq 2$,

$$\begin{aligned} \sqrt{F_i - F_{i-1}} &= \sqrt{\alpha^{\delta+i-1} \ln N - \alpha^{\delta+i-2} \ln N} \\ &= \sqrt{(\alpha - 1)\alpha^{\delta+i-2} \ln N}. \end{aligned} \quad (5)$$

We consider two cases: (i) $\delta \geq x$, and (ii) $\delta < x$. In the first case, using (4), we have

$$\alpha^{T+\delta} = \frac{L^*}{\ln N} \alpha^{1-x+\delta} \geq \frac{L^*}{\ln N} \alpha^{1-x+x} \geq \frac{L^*}{\ln N} \alpha.$$

Therefore, $L^* \leq \alpha^{\delta+T-1} \ln N = F_T$ and considering the algorithm definition (and in particular, definition (2)), we deduce that the last phase is the T th.

In the second case ($\delta < x$) we similarly have,

$$\alpha^{T+\delta} = \frac{L^*}{\ln N} \alpha^{1-x+\delta} \geq \frac{L^*}{\ln N} \alpha^{\delta},$$

and therefore, $L^* \leq \alpha^T \ln N \leq \alpha^{\delta+T} \ln N = F_{T+1}$. Thus, in this case the last phase is at most the $(T+1)$ st.

Let R_i and L_i^* denote the loss of algorithm RDWM and respectively, the loss of the best expert during the i th phase. Recalling that the guessed upper bound set by RDWM for the i th phase is $(F_i - F_{i-1}) \ln N$, for each phase i we can apply Theorem 3.1 to bound

$$R_i \leq L_i^* + \sqrt{(F_i - F_{i-1}) \ln N} + \frac{\ln N}{2 \ln 2} + 1,$$

where the extra 1 is due to the possible delay (of at most 1 loss unit) in detecting that the best expert loss exceeded the threshold. Therefore, in the first case ($\delta \geq x$) where there are T phases we have

$$\begin{aligned} L_{\text{RDWM}} &= \sum_{i=1}^T R_i \leq \sum_{i=1}^T \left(L_i^* + \sqrt{(F_i - F_{i-1}) \ln N} + \frac{\ln N}{2 \ln 2} + 1 \right) \\ &= L^* + T \cdot \left(\frac{\ln N}{2 \ln 2} + 1 \right) + \sum_{i=1}^T \sqrt{(F_i - F_{i-1}) \ln N} \\ &= L^* + (\log_{\alpha}(L^*/\ln N) - x + 1) \cdot \left(\frac{\ln N}{2 \ln 2} + 1 \right) \\ &\quad + \sum_{i=1}^T \sqrt{(F_i - F_{i-1}) \ln N} \\ &\leq L^* + \sum_{i=1}^T \sqrt{(F_i - F_{i-1}) \ln N} + O(\log L^* \log N). \end{aligned}$$

Using (5) we have

$$\begin{aligned}
\sum_{i=1}^T \sqrt{(F_i - F_{i-1}) \ln N} &= \sqrt{\ln N} \left(\sqrt{\alpha^\delta \ln N} + \sum_{i=2}^T \sqrt{(\alpha - 1) \alpha^{\delta+i-2} \ln N} \right) \\
&= \alpha^{\delta/2} \ln N \left(1 + \sqrt{\alpha - 1} \sum_{i=0}^{T-2} \alpha^{i/2} \right) \\
&= \alpha^{\delta/2} \ln N \left(1 + \frac{\sqrt{\alpha - 1}}{\sqrt{\alpha - 1}} \alpha^{(T-1)/2} - \frac{\sqrt{\alpha - 1}}{\sqrt{\alpha - 1}} \right) \\
&= \alpha^{\delta/2} \ln N \left(1 + \frac{\sqrt{\alpha - 1}}{\sqrt{\alpha - 1}} \alpha^{(\log_\alpha(L^*/\ln N) - x)/2} - \frac{\sqrt{\alpha - 1}}{\sqrt{\alpha - 1}} \right) \\
&= \alpha^{\delta/2} \ln N \left(1 + \frac{\sqrt{(\alpha - 1)L^*/\ln N}}{\sqrt{\alpha - 1}} \alpha^{-x/2} - \frac{\sqrt{\alpha - 1}}{\sqrt{\alpha - 1}} \right) \\
&< \frac{\sqrt{(\alpha - 1)L^* \ln N}}{\sqrt{\alpha - 1}} \alpha^{(\delta-x)/2},
\end{aligned}$$

where the last inequality follows from the inequality $1 - \frac{\sqrt{\alpha-1}}{\sqrt{\alpha-1}} < 0$, which holds since $\alpha > 1$. Overall, in the first case ($\delta \geq x$) we have

$$L_{\text{RDWM}} \leq L^* + \frac{\sqrt{(\alpha - 1)L^* \ln N}}{\sqrt{\alpha - 1}} \alpha^{(\delta-x)/2} + O(\log L^* \log N). \quad (6)$$

In the second case ($\delta < x$) we can obtain, using a very similar derivation,

$$L_{\text{RDWM}} \leq L^* + \frac{\sqrt{\alpha(\alpha - 1)L^* \ln N}}{\sqrt{\alpha - 1}} \alpha^{(\delta-x)/2} + O(\log L^* \log N). \quad (7)$$

Noting that δ is uniformly distributed in $[0, 1)$, we can bound the expected loss of RDWM by combining (6) and (7) as follows.

$$\begin{aligned}
E[L_{\text{RDWM}}] &< \int_0^x \left(L^* + \frac{\sqrt{(\alpha - 1)L^* \ln N}}{\sqrt{\alpha - 1}} \alpha^{(\delta-x)/2} + O(\log L^* \log N) \right) d\delta \\
&\quad + \int_x^1 \left(L^* + \frac{\sqrt{\alpha(\alpha - 1)L^* \ln N}}{\sqrt{\alpha - 1}} \alpha^{(\delta-x)/2} + O(\log L^* \log N) \right) d\delta \\
&= L^* + \frac{\sqrt{(\alpha - 1)L^* \ln N}}{\sqrt{\alpha - 1}} \left(\int_0^x \alpha^{(\delta-x+1)/2} d\delta + \int_x^1 \alpha^{(\delta-x)/2} d\delta \right) \\
&\quad + O(\log L^* \log N) \\
&= L^* + \frac{\sqrt{(\alpha - 1)L^* \ln N}}{\sqrt{\alpha - 1}} \cdot \frac{2(\sqrt{\alpha} - 1)}{\ln \alpha} + O(\log L^* \log N) \\
&= L^* + \frac{2\sqrt{(\alpha - 1)L^* \ln N}}{\ln \alpha} + O(\log L^* \log N)
\end{aligned}$$

□

Corollary 3.1. *The expected regret of RDWM is bounded above by*

$$2.49\sqrt{L^* \ln N} + O(\ln L^* \ln N).$$

Proof: We use the regret bound of Theorem 3.2. The function $f(\alpha) = 2\sqrt{\alpha - 1}/\ln \alpha$ is minimized when $2 + \alpha \ln \alpha = 2\alpha$. In terms of Lambert's W function this equation has the solution

$$\alpha^* = -2/W\left(\frac{-2}{e^2}\right) \approx 4.92155,$$

and it is not hard to see that $f(\alpha^*) < 2.48527$. □

4. The WMVP algorithm—absolute loss setting

In this section we present our second expert advice algorithm for the absolute loss setting. The algorithm is called “Weighted Majority with Vovk-Phases” or WMVP for short. A pseudo-code of the algorithm is given in figures 7 and 6. Figure 7 contains the master routine called WMVP. WMVP operates in phases where at the start of each phase it calls the prediction routine VPREDICT given in figure 6. WMVP is parameterized by a sequence of monotone increasing positive reals T_1, T_2, \dots , which specify “loss thresholds”. These thresholds determine the phases, where the i th phase lasts as long as the loss of best expert

Wmvp($\{T_i\}$)

Input: T_1, T_2, \dots , is any monotone increasing sequence of positive reals

- **Initialize**
 - $w_j \leftarrow 1, L_j \leftarrow 0, j = 1, \dots, N$
 - $\beta_1 \leftarrow \frac{1}{1 + \sqrt{\frac{2 \ln N}{T_1}}}$
 - $t \leftarrow 1$
- **For** $i := 1$ to ∞
 - **Interval run:**
 $[t_i, \{w_j\}\{L_j\}] \leftarrow \text{VPREDICT}(t, T_i, \beta_i, \{w_{l,i}\}, \{L_j\})$
 - **Compute next learning rate:**
 $\beta_{i+1} \leftarrow \frac{1}{1 + \sqrt{\frac{2 \ln N}{T_{i+1}}}}$
 - **If** $i > 1$
 - **Adjust weights** $w_j \leftarrow w_j \cdot \frac{U_{\beta_{i+1}}(L_j - T_i)}{U_{\beta_i}(L_j - T_i)}$
 (where U is the same as used in VPREDICT; (Eq. (9) at Fig 6)
 - $i \leftarrow i + 1, t \leftarrow t_i + 1$
- **endfor**

Figure 7. The WMVP algorithm.

so far is smaller than T_i . Although the algorithm given in figure 7 receives pre-specified loss thresholds, T_i may be chosen online.

Like RDWM of Section 3, predictions are formed only within phases and only by the VPREDICT routine. Note, however, that unlike algorithm RDWM we do not initialize expert weights when we call VPREDICT from WMVP. The expert weights and losses input to VPREDICT always reflect expert performance so far. At the end of each phase, VPREDICT returns to WMVP the last trial number, and updated expert weights and losses. WMVP then uses these losses to adjust expert weights. Then, it computes a new appropriate learning rate β and starts a new phase by calling VPREDICT again, etc.

Expert punishments are controlled via the β parameter which remains fixed throughout a phase, but is always increased by WMVP in the next phase (which means that experts are punished less severely as phases advance).

The expert weight adjustments performed by WMVP (at the end of a phase i) call for some attention. Notice that these adjustments always raise the weight of an expert whose loss exceeds the last loss threshold T_i (this follows from the fact that the β_i are increasing with i). Through these adjustments the algorithm performs “loss shifts” that coordinate expert weights so that at all times, the weights associated with experts reflect a (fictitious) situation where all past expert mistake rates were identical during each of the previous phases. Nevertheless, these adjustments always keep a set of weights which reflect the relative success of the experts so far. Before we give the following lemma, which formally establishes this fact, we introduce some notation that will be used throughout the paper.

First, j always denotes the j th expert ($1 \leq j \leq N$), t represents a trial number and i , a phase index. We denote by t_i the trial number where the i th phase ended. For all $1 \leq j \leq N$ and $t \geq 1$ we denote by $w_{j,t}$ and (resp. $L_{j,t}$) the weight (resp. loss) of the j th expert just after the t th trial (as computed by VPREDICT). The adjusted weight (of the j th expert) after the i th phase ends (at trial t_i) is denoted by $w'_{j,t_i} = w_{j,t_i} \cdot (\frac{\beta_{i+1}}{\beta_i})^{L_{j,t_i} - T_i}$. We set $W_i = \sum_{j=1}^N w_{j,t_i}$, the sum of expert weights at the end of the i th interval (before weight adjustments), and similarly $W'_i = \sum_{j=1}^N w'_{j,t_i}$ is the sum of adjusted weights just after the i th phase (and after weight adjustments). Thus, W'_i is the sum of weights at the beginning of the $(i+1)$ st interval (the beginning of the (t_i+1) st trial). By convention we set $W_0 = W'_0 = N$. For any outcome sequence \mathbf{y} we denote by \mathbf{y}_i the segment of \mathbf{y} corresponding to the i th phase; that is, $\mathbf{y}_i = y_{t_{i-1}+1}, \dots, y_{t_i}$. For each (phase) i we set $\tau_i = T_i - T_{i-1}$ where $T_0 = 0$.

Lemma 4.1 (*Weight adjustments*). *Let i be any phase index. During this phase, the j th expert weight w_j , just after this expert incurred loss L_j , is*

$$w_j = \left(\prod_{k=1}^{i-1} \beta_k^{\tau_k} \right) \cdot \beta_i^{L_j - T_{i-1}}$$

Proof: We prove the lemma by induction on i . The base case, $i = 1$, holds since the weight of the j th expert is simply $w_j = \beta_1^{L_j}$. Assume that the induction hypothesis holds for the i th phase. We prove that it holds for the $(i+1)$ st phase. By inspection of the algorithm, when the i th phase ends, for all $j = 1, \dots, n$, $L_j \geq T_i$, and by the induction hypothesis, at this moment (before the weight adjustments) $w_j = (\prod_{k=1}^{i-1} \beta_k^{\tau_k}) \cdot \beta_i^{L_j - T_{i-1}}$. But

$L_j - T_{i-1} = (L_j - T_i) + (T_i - T_{i-1}) = (L_j - T_i) + \tau_i$, so $w_j = (\prod_{k=1}^{i-1} \beta_k^{\tau_k}) \cdot \beta_i^{\tau_i} \cdot \beta_i^{L_j - T_i} = (\prod_{k=1}^i \beta_k^{\tau_k}) \cdot \beta_i^{L_j - T_i}$.

The adjusted weight is obtained by multiplying the weight by $(\frac{\beta_{i+1}}{\beta_i})^{L_j - T_i}$. Therefore, during the $(i + 1)$ st phase, $w_j = (\prod_{k=1}^i \beta_k^{\tau_k}) \cdot \beta_i^{L_j - T_i} \cdot (\frac{\beta_{i+1}}{\beta_i})^{L_j - T_i} = (\prod_{k=1}^i \beta_k^{\tau_k}) \cdot \beta_{i+1}^{L_j - T_i}$. \square

Let j and j' be two expert indices. A straightforward corollary of Lemma 4.1 is that during any phase i , while the experts j and j' have weights w_j and $w_{j'}$, and losses L_j and $L_{j'}$,

$$w_j > w_{j'} \Leftrightarrow \left(\prod_{k=1}^{i-1} \beta_k^{\tau_k} \right) \cdot \beta_i^{L_j - T_{i-1}} > \left(\prod_{k=1}^{i-1} \beta_k^{\tau_k} \right) \cdot \beta_i^{L_{j'} - T_{i-1}} \Leftrightarrow L_j < L_{j'}$$

The following lemma concerns a basic property of VPREDICT and will be used in our analysis for proving Lemma 4.3. The lemma is an adaptation of a result that was first established by Vovk (1990). The proof we provide in Appendix A is adapted from Cesa-Bianchi et al. (1997).

Lemma 4.2. *For any $0 < \beta < 1$, and any outcome sequence \mathbf{y} of length ℓ ,*

$$L_{\text{VPREDICT}}(\mathbf{y}) \leq \frac{1}{2 \cdot \ln \frac{2}{1+\beta}} \ln \left(\frac{\sum_{j=1}^N w_{j,0}}{\sum_{j=1}^N w_{j,\ell}} \right), \quad (10)$$

where, for each $1 \leq j \leq N$ and $1 \leq t \leq \ell$, $w_{j,t}$ is the weight of the j th expert after the t th trial, and the $w_{j,0}$ are the initial weights.

Let $B_i = \frac{1}{2 \ln \frac{2}{1+\beta_i}}$. Using Lemma 4.2 we can now upper bound the loss incurred during each phase.

Lemma 4.3 (Phase loss). *For any phase i and any $0 < \beta_{i-1} < \beta_i < 1$,*

$$L_{\text{WMVP}}(\mathbf{y}_i) \leq B_i (\ln W'_{i-1} - \ln W_i).$$

Proof: By definition of the W'_{i-1} and W_i ,

$$\begin{aligned} B_i (\ln W'_{i-1} - \ln W_i) &= B_i \left(\ln \sum_{j=1}^N w_{j,t_{i-1}} \cdot \left(\frac{\beta_i}{\beta_{i-1}} \right)^{L_{j,t_{i-1}} - T_{i-1}} - \ln \sum_{j=1}^N w_{j,t_i} \right) \\ &= B_i \ln \frac{\sum_{j=1}^N w_{j,t_{i-1}} \cdot \left(\frac{\beta_i}{\beta_{i-1}} \right)^{L_{j,t_{i-1}} - T_{i-1}}}{\sum_{j=1}^N w_{j,t_i}}. \end{aligned}$$

Noting that in the numerator we have the weights at the beginning of the phase and in the denominator, the weights at the end of the phase, we can complete the proof by applying Lemma 4.2. \square

The following lemma bounds the loss of algorithm WMVP with respect to finite outcome sequences.

Lemma 4.4 (*Finite sequence loss*). *For any outcome sequence \mathbf{y} and expert responses,*

$$L_{\text{WMVP}}(\mathbf{y}) < \ln N \cdot \left(B_k + \sum_{i=1}^{k-1} \left(B_i - \frac{\ln \beta_{i+1}}{\ln \beta_i} \cdot B_{i+1} \right) \right) - \sum_{i=1}^{k-1} (\tau_i \ln \beta_i B_i) \\ - z \ln \beta_k B_k - \ln \beta_1 B_1, \quad (11)$$

where k is the consequent number of phases, with a possibly incomplete last phase, and $0 \leq z \leq \tau_k$ is the loss incurred by the best expert during the last phase.

Proof: Decomposing \mathbf{y} into phases $\mathbf{y} = \mathbf{y}_1, \dots, \mathbf{y}_k$ (and the last phase may be incomplete) we have $L_{\text{WMVP}} = \sum_{i=1}^k L_{\text{WMVP}}(\mathbf{y}_i)$. By Lemma 4.3,

$$\sum_{i=1}^k L_{\text{WMVP}}(\mathbf{y}_i) \leq \sum_{i=1}^k B_i (\ln W'_{i-1} - \ln W_i), \quad (12)$$

so for an upper bound on the algorithm's loss it is sufficient to maximize the upper bounds on the individual phase losses (the right hand side of (12)), where the maximization is over feasible expert weights. Recalling that $W'_0 = N$, the right hand side of (12) can be rewritten as

$$B_1 \ln N - B_k \ln W_k + \sum_{i=1}^{k-1} (B_{i+1} \ln W'_i - B_i \ln W_i). \quad (13)$$

For each phase i we define

$$W_i^{\max} = \sum_{j=1}^N \prod_{k=1}^i \beta_k^{\tau_k} = N \prod_{k=1}^i \beta_k^{\tau_k}. \quad (14)$$

W_i^{\max} is the maximum sum of possible expert weights after phase i . Also, define $\gamma_i = W_i / W_i^{\max}$. Let χ be the solution of

$$W_i = \sum_{j=1}^N \prod_{k=1}^i \beta_k^{\tau_k} \beta_i^{L_j - T_i} = \sum_{j=1}^N \prod_{k=1}^i \beta_k^{\tau_k} \beta_i^{\chi} = N \cdot \prod_{k=1}^i \beta_k^{\tau_k} \beta_i^{\chi}. \quad (15)$$

Clearly, $\gamma_i = \beta_i^{\chi}$. Note that $\chi = \log_{\beta_i} \gamma_i$ represents a (uniform) extra loss above T_i in a scenario where all experts (uniformly) incur exactly the same loss during the i th phase. Similarly, let χ' be the solution of

$$W'_i = \sum_{j=1}^N \prod_{k=1}^i \beta_k^{\tau_k} \beta_{i+1}^{L_j - T_i} = \sum_{j=1}^N \prod_{k=1}^i \beta_k^{\tau_k} \beta_{i+1}^{\chi'} = N \cdot \prod_{k=1}^i \beta_k^{\tau_k} \beta_{i+1}^{\chi'}. \quad (16)$$

By Lemma A.1 (appearing in the appendix), $\chi' \geq \chi$ and we conclude that

$$W'_i = N \cdot \prod_{k=1}^i \beta_k^{\tau_k} \beta_{i+1}^{\chi'} < N \cdot \prod_{k=1}^i \beta_k^{\tau_k} \beta_{i+1}^{\chi}. \quad (17)$$

The summation in (13) (the r.h. term) can now be bounded (from above) as follows.

r.h. term of (13)

$$\begin{aligned} &= \sum_{i=1}^{k-1} (B_{i+1} \ln W'_i - B_i \ln W_i) \\ &= \sum_{i=1}^{k-1} \left\{ B_{i+1} \ln \left(N \prod_{k=1}^i \beta_k^{\tau_k} \beta_{i+1}^{\chi'} \right) - B_i \ln W_i \right\} \\ &\leq \sum_{i=1}^{k-1} \left\{ B_{i+1} \ln \left(N \prod_{k=1}^i \beta_k^{\tau_k} \beta_{i+1}^{\chi} \right) - B_i \ln W_i \right\} \quad (18) \\ &= \sum_{i=1}^{k-1} \left\{ B_{i+1} \ln \left(\left(N \prod_{k=1}^i \beta_k^{\tau_k} \beta_{i+1}^{\chi} \right) \left(\frac{\beta_{i+1}}{\beta_i} \right)^{\chi} \right) - B_i \ln W_i \right\} \\ &= \sum_{i=1}^{k-1} \left\{ B_{i+1} \ln \left(W_i \left(\frac{\beta_{i+1}}{\beta_i} \right)^{\chi} \right) - B_i \ln W_i \right\} \\ &= \sum_{i=1}^{k-1} \left\{ B_{i+1} \ln \left(W_i^{\max} \gamma_i \left(\frac{\beta_{i+1}}{\beta_i} \right)^{\log_{\beta_i} \gamma_i} \right) - B_i \ln (W_i^{\max} \gamma_i) \right\} \\ &= \sum_{i=1}^{k-1} \left\{ B_{i+1} \left(\ln W_i^{\max} + \ln \gamma_i + \frac{\ln \gamma_i}{\ln \beta_i} \ln \left(\frac{\beta_{i+1}}{\beta_i} \right) \right) - B_i (\ln W_i^{\max} + \ln \gamma_i) \right\} \\ &= \sum_{i=1}^{k-1} \left\{ \ln W_i^{\max} (B_{i+1} - B_i) + \ln \gamma_i \left(B_{i+1} - B_i + B_{i+1} \frac{\ln \beta_{i+1}}{\ln \beta_i} - B_{i+1} \frac{\ln \beta_i}{\ln \beta_i} \right) \right\} \\ &= \sum_{i=1}^{k-1} \left\{ \ln W_i^{\max} (B_{i+1} - B_i) + \ln \gamma_i \left(B_{i+1} \frac{\ln \beta_{i+1}}{\ln \beta_i} - B_i \right) \right\}, \quad (19) \end{aligned}$$

where (18) follows from (17). Using (19) we can write an upper bound on (13) as $\mathcal{J}_1 + \mathcal{J}_2$, where

$$\mathcal{J}_1 = \sum_{i=1}^{k-1} \ln W_i^{\max} (B_{i+1} - B_i) + B_1 \ln N - B_k \ln W_k \quad (20)$$

$$\mathcal{J}_2 = \sum_{i=1}^{k-1} \ln \gamma_i \left(B_{i+1} \frac{\ln \beta_{i+1}}{\ln \beta_i} - B_i \right). \quad (21)$$

Taking $W_0^{\max} = N$, we have

$$\mathcal{J}_1 = B_k (\ln W_{k-1}^{\max} - \ln W_k) + \sum_{i=1}^{k-1} B_i (\ln W_{i-1}^{\max} - \ln W_i^{\max}). \quad (22)$$

We first consider the r.h. term. (The reason we partition \mathcal{J}_1 is that the last phase may be incomplete.) By definition of W^{\max} (see Eq. (14)),

$$\ln W_{i-1}^{\max} - \ln W_i^{\max} = \ln N + \sum_{q=1}^{i-1} \tau_q \ln \beta_q - \left(\ln N + \sum_{q=1}^i \tau_q \ln \beta_q \right) = -\tau_i \ln \beta_i.$$

Therefore, the r.h. term of \mathcal{J}_1 equals $-\sum_{i=1}^{k-1} B_i \tau_i \ln \beta_i$. The only free parameters in \mathcal{J}_1 (which can be controlled by the adversary) are k and W_k . For each choice of k by the adversary, the best choice of expert response (that maximizes the regret) would minimize W_k (appearing in the l.h. term of \mathcal{J}_1). Clearly, the sum of all weights is not smaller than the weight of the best expert, which we denote by W^{\min} (W_i^{\min} after the i th phase),

$$W_k > W_k^{\min} \equiv \beta_k^{z+1} \prod_{i=1}^{k-1} \beta_i^{\tau_i}, \quad (23)$$

where the extra β_k factor in W_k^{\min} is due to a maximum delay (of one loss unit) in detecting that the loss of the best expert increased above the threshold T_{k-1} . Overall, we have, for the l.h. term of \mathcal{J}_1 ,

$$\begin{aligned} B_k (\ln W_{k-1}^{\max} - \ln W_k) &< B_k \left(\ln W_{k-1}^{\max} - \ln \left(\beta_k^{z+1} \prod_{i=1}^{k-1} \beta_i^{\tau_i} \right) \right) \\ &= B_k \left(\ln N + \sum_{q=1}^{k-1} \tau_q \ln \beta_q - (z+1) \ln \beta_k - \sum_{q=1}^k \tau_q \ln \beta_q \right) \\ &= B_k (\ln N - \ln \beta_k (1+z)). \end{aligned} \quad (24)$$

Thus, we can bound \mathcal{J}_1 as follows:

$$\mathcal{J}_1 < -\sum_{i=1}^{k-1} B_i \tau_i \ln \beta_i + B_k (\ln N - \ln \beta_k (1+z)). \quad (25)$$

We now consider \mathcal{J}_2 . Here the free parameters are γ_i 's. It is not hard to see¹¹ that $B_{i+1} \frac{\ln \beta_{i+1}}{\ln \beta_i} - B_i < 0$, and since $0 < \gamma_i \leq 1$, by minimizing the γ_i 's in \mathcal{J}_2 we will get an upper bound on \mathcal{J}_2 .

By the definitions of γ_i and of W^{\min} , $W_i^{\max} \gamma_i = W_i > W_i^{\min}$. Thus,

$$\gamma_i > \frac{W_i^{\min}}{W_i^{\max}} = \frac{\beta_i \prod_{q=1}^i \beta_q^{\tau_q}}{N \prod_{q=1}^i \beta_q^{\tau_q}} = \frac{\beta_i}{N}. \quad (26)$$

Therefore,

$$\begin{aligned}
\mathcal{J}_2 &= \sum_{i=1}^{k-1} \ln \gamma_i \left(B_{i+1} \frac{\ln \beta_{i+1}}{\ln \beta_i} - B_i \right) \\
&< \sum_{i=1}^{k-1} \ln \frac{\beta_i}{N} \left(B_{i+1} \frac{\ln \beta_{i+1}}{\ln \beta_i} - B_i \right) \\
&= \ln N \sum_{i=1}^{k-1} \left(B_i - B_{i+1} \frac{\ln \beta_{i+1}}{\ln \beta_i} \right) + \sum_{i=1}^{k-1} \ln \beta_i \left(B_{i+1} \frac{\ln \beta_{i+1}}{\ln \beta_i} - B_i \right) \\
&= \ln N \sum_{i=1}^{k-1} \left(B_i - B_{i+1} \frac{\ln \beta_{i+1}}{\ln \beta_i} \right) + \sum_{i=1}^{k-1} (B_{i+1} \ln \beta_{i+1} - B_i \ln \beta_i) \\
&= \ln N \sum_{i=1}^{k-1} \left(B_i - B_{i+1} \frac{\ln \beta_{i+1}}{\ln \beta_i} \right) + B_k \ln \beta_k - B_1 \ln \beta_1. \quad \square
\end{aligned}$$

In the expert advice game the adversary's goal, when playing against WMVP, is to choose an outcome sequence \mathbf{y} and expert predictions so as to maximize the regret, defined to be $L_{\text{WMVP}}(\mathbf{y}) - L^*(\mathbf{y})$, with $L^*(\mathbf{y})$ being the loss of the best expert. One interesting property of an expert advice algorithm is the growth rate of its regret, as the loss of the best expert increases. The following theorem determines this asymptotic rate for WMVP applied with "geometric" thresholds.

Theorem 4.1. *Let T_1, T_2, \dots be any geometric loss threshold sequence satisfying $T_{i+1} = \alpha T_i$ where $\alpha > 1$. Then, as L^* , the loss of the best expert increases,*

$$L_{\text{WMVP}} - L^* < \left(\frac{2\alpha + \sqrt{\alpha} + 1}{2\sqrt{2\alpha}} + o(1) \right) \cdot \sqrt{\ln N \cdot L^*}.$$

Proof: Consider

$$\frac{L_{\text{WMVP}} - L^*}{\sqrt{L^* \cdot \ln N}}. \quad (27)$$

Our goal is to show that the asymptotic value of this fraction, as L^* increases, is at most $\frac{2\alpha + \sqrt{\alpha} + 1}{2\sqrt{2\alpha}}$. Consider a finite game with k phases where the k th phase may be incomplete. By Lemma 4.4,

$$\begin{aligned}
L_{\text{WMVP}} &< \ln N \cdot \left(B_k + \sum_{i=1}^{k-1} \left(B_i - \frac{\ln \beta_{i+1}}{\ln \beta_i} \cdot B_{i+1} \right) \right) - \sum_{i=1}^{k-1} \tau_i \cdot (\ln \beta_i B_i) \\
&\quad - z \cdot (\ln \beta_k B_k) - \ln \beta_1 B_1.
\end{aligned}$$

The proof is straightforward: In (27) we substitute this upper bound for L_{WMVP} (ignoring the constant $\ln \beta_1 B_1$) and substitute $T_{k-1} + z$ for L^* (recall the definition of z in Lemma 4.4). Specifically, let

$$\begin{aligned} u_k(z) &= \ln N \cdot \left(B_k + \sum_{i=1}^{k-1} \left(B_i - \frac{\ln \beta_{i+1}}{\ln \beta_i} \cdot B_{i+1} \right) \right) \\ &\quad - \sum_{i=1}^{k-1} \tau_i \cdot (\ln \beta_i B_i) - z \cdot (\ln \beta_k B_k); \\ C_k(z) &= \frac{u_k(z) - (T_{k-1} + z)}{\sqrt{(T_{k-1} + z) \ln N}}. \end{aligned}$$

To prove the theorem we need to show that

$$\sup_z \lim_{k \rightarrow \infty} C_k(z) \leq \frac{2\alpha + \sqrt{\alpha} + 1}{2\sqrt{2\alpha}}.$$

We write $C_k(z) = \mathcal{I}_1 + \mathcal{I}_2 + \mathcal{I}_3$, where

$$\mathcal{I}_1 = \frac{\ln N \cdot B_k}{\sqrt{(T_{k-1} + z) \cdot \ln N}} = \frac{\sqrt{\ln N} \cdot B_k}{\sqrt{(T_{k-1} + z)}} \quad (28)$$

$$\mathcal{I}_2 = \frac{\ln N \cdot \sum_{i=1}^{k-1} \left(B_i - \frac{\ln \beta_{i+1}}{\ln \beta_i} \cdot B_{i+1} \right)}{\sqrt{(T_{k-1} + z) \cdot \ln N}} = \frac{\sqrt{\ln N} \cdot \sum_{i=1}^{k-1} \left(B_i - \frac{\ln \beta_{i+1}}{\ln \beta_i} \cdot B_{i+1} \right)}{\sqrt{(T_{k-1} + z)}} \quad (29)$$

$$\mathcal{I}_3 = \frac{-\sum_{i=1}^{k-1} \tau_i \cdot (\ln \beta_i B_i + 1) - z \cdot (\ln \beta_k B_k + 1)}{\sqrt{(T_{k-1} + z) \cdot \ln N}}. \quad (30)$$

By Lemma A.2 (appearing in the appendix),

$$\lim_{k \rightarrow \infty} \mathcal{I}_1 \leq \sqrt{\alpha/2};$$

$$\lim_{k \rightarrow \infty} \mathcal{I}_2 = 0;$$

$$\lim_{k \rightarrow \infty} \mathcal{I}_3 \leq \frac{\sqrt{\alpha} + 1}{2\sqrt{2\alpha}}.$$

To conclude, we have

$$\begin{aligned} \sup_z \lim_{k \rightarrow \infty} C_k(z) &= \sup_z \lim_{k \rightarrow \infty} \mathcal{I}_1 + \mathcal{I}_2 + \mathcal{I}_3 \\ &\leq \frac{\sqrt{\alpha}}{\sqrt{2}} + \frac{\sqrt{\alpha} + 1}{2\sqrt{2\alpha}} \\ &= \frac{2\alpha + \sqrt{\alpha} + 1}{2\sqrt{2\alpha}}. \end{aligned}$$

□

Corollary 4.1. *Let $\epsilon > 0$ be given. Then, algorithm WMVP can be operated such that its asymptotic regret (for sufficiently large L^*) is bounded above as*

$$L_{\text{WMVP}} - L^* < (\sqrt{2} + \epsilon) \cdot \sqrt{\ln NL^*}.$$

Proof: By Theorem 4.1, for any threshold sequence of the form $T_{i+1} = \alpha T_i$, $\alpha > 1$, an asymptotic bound of $\frac{2\alpha + \sqrt{\alpha+1}}{2\sqrt{2\alpha}} \sqrt{\ln NL^*}$ on the regret is guaranteed. It is easy to see that this bound is minimized when $\alpha \rightarrow 1$. In particular, with $\alpha = 1$, $\frac{2\alpha + \sqrt{\alpha+1}}{2\sqrt{2\alpha}} = \sqrt{2}$. It is not hard to see that for any $\epsilon > 0$ there exists α_ϵ such that the bound of the corollary holds. Note that operating WMVP with arbitrarily small $\alpha > 1$ is equivalent to terminating each application of the VPREDICT routine immediately after an increase in the loss of the best expert is detected. \square

5. The WMVP algorithm—0/1 loss setting

In this section we present two variants of the WMVP algorithm for the 0/1-loss setting. For one of these variants we are able to provide an asymptotic bound of $(\frac{\alpha + \sqrt{\alpha+1}}{2\sqrt{\alpha}} + o(1)) \cdot \sqrt{\ln N \cdot L^*}$ on the regret of WMVP. As it is in the absolute loss case, this performance guarantee is optimized when $\alpha \rightarrow 1$ in which case the asymptotic bound is $(\frac{3}{\sqrt{2}} + o(1)) \cdot \sqrt{\ln N \cdot L^*}$.

In the 0/1-loss setting we consider outcome values that can be elements of a finite alphabet (i.e. “nominal” values). That is, no non-trivial metric or even an order relation are assumed to hold over the outcome set. Both the expert predictions and master algorithm predictions are required to be elements of this outcome set. Although the 0/1 loss function is an extreme special case of the absolute loss, the above WMVP algorithm, based on Vovk’s (absolute loss) prediction routine (VPREDICT), does not generate nominal predictions and is therefore not suitable for the 0/1-loss setting. In order to generate nominal predictions our master algorithm must rely on appropriate prediction routines. Whenever the alphabet is binary, two prediction routines, proposed by Littlestone and Warmuth (1994), can be utilized:

- **Weighted Majority General (WMG):** Uses the weighted average of experts’ predictions (which can be elements of $[0, 1]$) to form its prediction by thresholding this average against $1/2$ (i.e. it is 0 iff the average is $< 1/2$). See figure 1.
- **Weighted Majority Randomized (WMR):** Randomly chooses one expert, with probability proportional to its weight, and uses this expert’s prediction. See figure 2.

Notice that Wmr can be directly applied when the alphabet (outcome set) has cardinality larger than 2.¹² In this section we discuss and analyze two variants of the WMVP algorithm which are based on the WMG and Wmr prediction routines.

When we measure prediction discrepancies using the 0/1-loss, the total loss of experts (and that of the master algorithm) equal the number of mistakes made by the experts (resp. the master algorithm). Thus, throughout this section we often use m_j and M for the number of mistakes made by the j th expert and number of mistakes made by the master algorithm, respectively, and we note that always $m_j = L_j$ and $M = L_{\text{WMVP}}$.

Table 3. Upper bounds attained by three Weighted Majority prediction variants.

	VPREDICT	Wmr	WMG
Bound	$\frac{\ln N+m^* \cdot \ln \frac{1}{\beta}}{2 \ln \frac{2}{1+\beta}}$	$\frac{\ln N+m^* \cdot \ln \frac{1}{\beta}}{1-\beta}$	$\frac{\ln N+m^* \cdot \ln \frac{1}{\beta}}{\ln \frac{2}{1+\beta}}$

It is interesting to compare the (known) worst case bounds that can be attained by the three prediction routines we consider: VPREDICT, Wmr and WMG (which all depend on a particular learning rate β). These bounds, which were established in (Vovk, 1990; Littlestone and Warmuth, 1994), are summarized in Table 3, where m^* denotes the number of mistakes (total loss) of the best expert and, as usual, N is the number of experts. As can be seen, the three bounds differ in their denominators. In addition, as will be later shown, when using our techniques, the WMG bound is not sufficient to guarantee an asymptotic bound on the regret. In figure 8 we plot the three denominators. For any $\beta \in [0, 1]$, the denominator of VPREDICT is the largest, thus producing the best (smallest) bound. As can be observed, the second best bound is attained by Wmr.

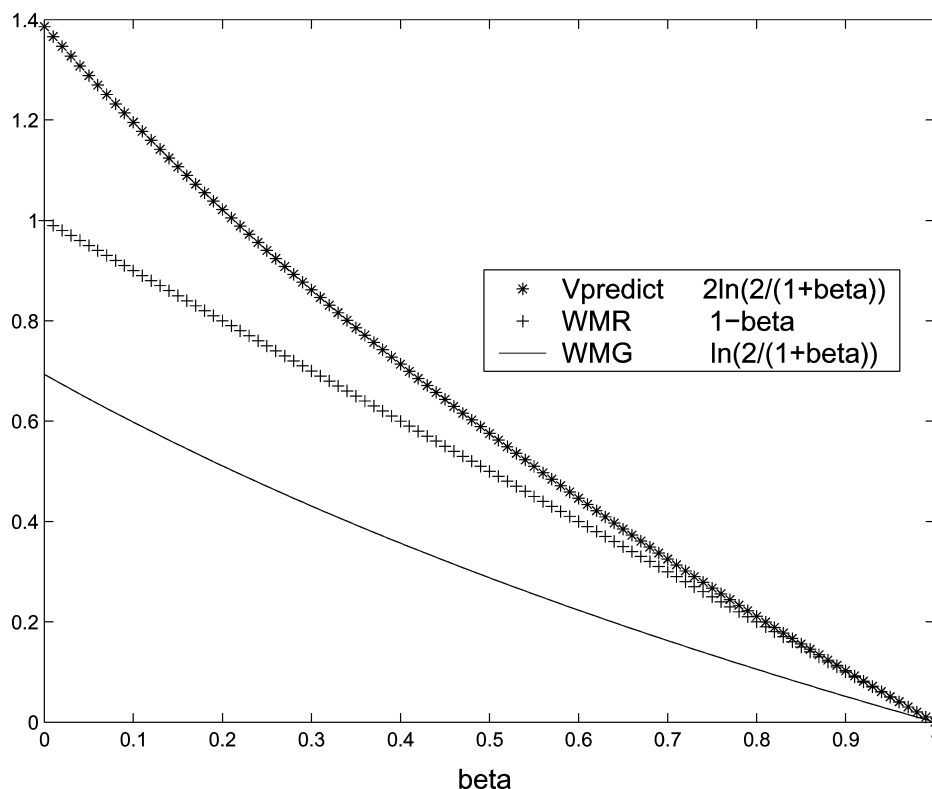


Figure 8. Denominators of the bounds for VPREDICT, WMR and WMG.

Vpredict_{0/1 Loss}($t, T, \beta, \{w_j\}, \{L_j\}$)

Inputs: t : the next trial number; T : threshold on the loss of the best expert (that is, the VPREDICT routine will run as long as the best expert loss $\leq T$); β : an update factor; $\{w_j\}$ and $\{L_j\}$ are, respectively, the initial weights and losses of the experts. **Outputs:** termination trial number, the sets of updated expert weights $\{w_j\}$ and losses $\{L_j\}$

- **Repeat**
 - **Receive** t^{th} trial expert predictions, $\xi_{1,t}, \dots, \xi_{n,t}$
 - $r_t \leftarrow \frac{\sum_{i=1}^N w_i \xi_{i,t}}{\sum_{i=1}^N w_i}$
 - **Predict**
 - (WMG) $\hat{y}_t = 1$ iff $r_t \geq \frac{1}{2}$ (and 0 otherwise)
 - (WMR) $\hat{y}_t = \xi_{i,t}$ with probability $\frac{w_i}{\sum_{j=1}^N w_j}$
 - **Receive** the true t^{th} trial outcome y_t
 - **Update** expert cumulative losses:
 - $L_{\xi_j} \leftarrow L_j + L_{0/1}(\xi_{j,t}, y_t), \quad j = 1, \dots, n$
 - **Update** (punish) expert weights:
 - $w_j \leftarrow w_j \cdot \beta^{L_{0/1}(\xi_{j,t}, y_t)} \quad j = 1, \dots, n$ (see below)
 - $t \leftarrow t + 1$
- **Until** $\min_j(L_j) \geq T$
- **Return** $t - 1, \{w_j\}, \{L_j\}$

Figure 9. Two versions of the 0/1-VPREDICT routine, based on WMG and Wmr.

In figure 9 we provide a pseudo code for adapted versions of both Wmr and WMG. The routines (called here 0/1-VPREDICT) are adapted to work with our WMVP master algorithm. The only differences of these routines from the absolute loss version of VPREDICT (figure 6) are the prediction functions (which are now based on either WMG or Wmr), and the update function, which is now based on the 0/1-loss, denoted $L_{0/1}(\cdot)$ and defined to be

$$L_{0/1}(x, y) = \begin{cases} 0, & \text{if } x = y \\ 1, & \text{otherwise.} \end{cases}$$

The following lemma, due to Littlestone and Warmuth (1994), bounds the loss of the WMG and Wmr variants of VPREDICT for any finite segment of the outcome sequence. This lemma is analogous to Lemma 4.2 and is given here without a proof.

Lemma 5.1 (WMG and WMR, Littlestone and Warmuth, 1994). *Let \mathbf{y} be any outcome (sub)sequence, and $0 < \beta < 1$ be the update factor used while processing the sequence \mathbf{y} . Let M_{WMG} (resp. M_{WMR}) denote the number of mistakes (total loss) suffered by the WMG (resp. Wmr) variant of VPREDICT when applied with a pool of N experts. Let W_{start} and W_{end} denote the sum of expert weights before and after processing \mathbf{y} , respectively, and let*

m^* be the number of mistakes incurred by the best expert over the sequence \mathbf{y} . Then,

$$M_{\text{WMG}} \leq \frac{\ln \frac{W_{\text{start}}}{W_{\text{end}}}}{\ln \frac{2}{1+\beta}} \leq \frac{\ln N + m^* \cdot \ln \frac{1}{\beta}}{\ln \frac{2}{1+\beta}};$$

$$E[M_{\text{WMr}}] \leq \frac{E[\ln \frac{W_{\text{start}}}{W_{\text{end}}}]}{1-\beta} \leq \frac{\ln N + E[m^*] \cdot \ln \frac{1}{\beta}}{1-\beta}.$$

Using Lemma 5.1 we can easily bound the loss incurred by WMVP during each phase. Denote by $L_{\text{WMVP}}^{\text{WMG}}(\cdot)$ (resp. $L_{\text{WMVP}}^{\text{WMr}}(\cdot)$) the loss suffered by WMVP when applied with the WMG (resp. WMr) variant of VPREDICT.

Lemma 5.2 (Phase loss). *For any phase i and any $0 \leq \beta_i \leq 1$,*

$$L_{\text{WMVP}}^{\text{WMr}}(\mathbf{y}_i) \leq \frac{(\ln W'_{i-1} - \ln W_i)}{1-\beta}$$

$$L_{\text{WMVP}}^{\text{WMG}}(\mathbf{y}_i) \leq \frac{(\ln W'_{i-1} - \ln W_i)}{\ln \frac{2}{1+\beta}}.$$

Proof: By definition of the W'_{i-1} and W_i (see Section 4),

$$B_i(\ln W'_{i-1} - \ln W_i) = B_i \left(\ln \sum_{j=1}^N w_{j,t_{i-1}} \cdot \left(\frac{\beta_i}{\beta_{i-1}} \right)^{L_{j,t_{i-1}} - T_{i-1}} - \ln \sum_{j=1}^N w_{j,t_i} \right)$$

$$= B_i \ln \frac{\sum_{j=1}^N w_{j,t_{i-1}} \cdot \left(\frac{\beta_i}{\beta_{i-1}} \right)^{L_{j,t_{i-1}} - T_{i-1}}}{\sum_{j=1}^N w_{j,t_i}},$$

where $B_i = \frac{1}{1-\beta_i}$ for the WMr variant, and $B_i = \frac{1}{\ln \frac{2}{1+\beta_i}}$ for the WMG variant. The proof is completed by applying Lemma 5.1, with the appropriate bounds for WMG and WMr, respectively. \square

The following lemma bounds the loss of the 0/1-loss variants of the WMVP with respect to a finite outcome sequence. This lemma is analogous to Lemma 4.4.

Lemma 5.3 (Finite sequence 0/1-loss). *For any outcome sequence \mathbf{y} and expert responses,*

$$L_{\text{WMVP}}^{0/1}(\mathbf{y}) < \ln N \cdot \left(B_k + \sum_{i=1}^{k-1} \left(B_i - \frac{\ln \beta_{i+1}}{\ln \beta_i} \cdot B_{i+1} \right) \right)$$

$$- \sum_{i=1}^k B_i \tau_i \ln \beta_i - z \cdot (B_k \ln \beta_k),$$

where k is the consequent number of phases, $0 \leq z \leq \tau_k$ is the loss incurred by the best expert during the last phase, and $B_i = \frac{1}{2} \frac{1}{\ln \frac{1}{1+\beta_i}}$ in the case where WMVP is applied with the WMG variant of VPREDICT, and $B_i = \frac{1}{1-\beta}$ in the case of the Wmr variant.

Proof: The proof closely follows the proof of Lemma 4.4. We only specify the changes. In the 0/1-loss case we can assume, without loss of generality, that the thresholds T_i are natural numbers. Otherwise, we can take instead $T'_i = \lceil T_i \rceil$ without changing the behavior of the algorithm. Thus, the first change is that now

$$W_k^{\min} = \beta_k^z \prod_{l=1}^{k-1} \beta_l^{\tau_l}, \quad (31)$$

instead of (23). This affects (and simplifies) inequality (24) which is now

$$B_k (\ln W_{k-1}^{\max} - \ln W_k^{\max}) \leq B_k \ln N - B_k \ln \beta_k z.$$

Consequently, we have the following bound on \mathcal{J}_1

$$\mathcal{J}_1 \leq - \sum_{i=1}^{k-1} B_i \tau_i \ln \beta_i + B_k (\ln N - \ln \beta_k \tau_k) = B_k \ln N - \sum_{i=1}^{k-1} B_i \tau_i \ln \beta_i - B_k \cdot z \ln \beta_k,$$

instead of the previous bound of Eq. (25).

The updated expression for W_k^{\min} (31) also affects the bound on γ_i (Eq. (26)), which now becomes $\gamma_i > \frac{1}{N}$. Consequently, we have

$$\mathcal{J}_2 < \sum_{i=1}^{k-1} \ln \frac{1}{N} \left(B_{i+1} \frac{\ln \beta_{i+1}}{\ln \beta_i} - B_i \right) = \ln N \sum_{i=1}^{k-1} \left(B_i - B_{i+1} \frac{\ln \beta_{i+1}}{\ln \beta_i} \right),$$

which completes the proof. \square

Remark. Notice that the overall difference between the statements of Lemmas 4.4 and 5.3 is the additional $-\ln \beta_1 B_1$ term (in the bound of Lemma 4.4) and the definitions of β_i and B_i .

For the 0/1 loss variant of WMVP, applied with the Wmr prediction routine, we have the following theorem.

Theorem 5.1. *Let T_1, T_2, \dots be any sequence of thresholds satisfying $T_{i+1} = \alpha T_i$ where $\alpha > 1$. Then, as L^* , the expected loss of the best expert increases,*

$$L_{\text{WMVP}}^{\text{WMR}} - L^* < \left(\frac{\alpha + \sqrt{\alpha} + 1}{\sqrt{2\alpha}} + o(1) \right) \cdot \sqrt{\ln N \cdot L^*}.$$

Proof: The proof very closely follows the proof of Theorem 4.1. The only differences are the bound on the limit of the term \mathcal{I}_3 , and the derivations of the limits of \mathcal{I}_1 , \mathcal{I}_2 and \mathcal{I}_3 , which now depend on a different definition of β_i (and B_i). By Lemma 5.3.

$$L_{\text{WMVP}}^{\text{WMR}}(\mathbf{y}) < \ln N \cdot \left(B_k + \sum_{i=1}^{k-1} \left(B_i - \frac{\ln \beta_{i+1}}{\ln \beta_i} \cdot B_{i+1} \right) \right) - \sum_{i=1}^{k-1} B_i \tau_i \ln \beta_i - z \cdot (B_k \ln \beta_k)$$

We substitute $T_{k-1} + z$ for L^* , and let

$$u_k(z) = \ln N \cdot \left(B_k + \sum_{i=1}^{k-1} \left(B_i - \frac{\ln \beta_{i+1}}{\ln \beta_i} \cdot B_{i+1} \right) \right) - \sum_{i=1}^{k-1} B_i \tau_i \ln \beta_i - z \cdot (B_k \ln \beta_k);$$

$$C_k(z) = \frac{u_k(z) - (T_{k-1} + z)}{\sqrt{(T_{k-1} + z) \ln N}}.$$

To prove the theorem we need to show that

$$\sup_z \lim_{k \rightarrow \infty} C_k(z) \leq \frac{\alpha + \sqrt{\alpha} + 1}{\sqrt{2\alpha}}.$$

We write $C_k(z) = \mathcal{I}_1 + \mathcal{I}_2 + \mathcal{I}_3$, where

$$\mathcal{I}_1 = \frac{\sqrt{\ln N} \cdot B_k}{\sqrt{T_{k-1} + z}};$$

$$\mathcal{I}_2 = \frac{\sqrt{\ln N} \cdot \sum_{i=1}^{k-1} \left(B_i - \frac{\ln \beta_{i+1}}{\ln \beta_i} \cdot B_{i+1} \right)}{\sqrt{T_{k-1} + z}};$$

$$\mathcal{I}_3 = \frac{-\sum_{i=1}^{k-1} \tau_i \cdot (\ln \beta_i B_i + 1) - z \cdot (\ln \beta_k B_k + 1)}{\sqrt{(T_{k-1} + z) \cdot \ln N}}.$$

By Lemma A.3 (appearing in the appendix)

$$\mathcal{I}_1 \leq \frac{\sqrt{\alpha}}{\sqrt{2}};$$

$$\mathcal{I}_2 = 0;$$

$$\mathcal{I}_3 \leq \frac{1}{\sqrt{2}} \left(1 + \frac{1}{\sqrt{\alpha}} \right).$$

To conclude, when using Wmr, we have

$$\sup_z \lim_{k \rightarrow \infty} C_k(z) \leq \frac{\sqrt{\alpha}}{\sqrt{2}} + \frac{1}{\sqrt{2}} \left(1 + \frac{1}{\sqrt{\alpha}} \right) = \left(\frac{\alpha + \sqrt{\alpha} + 1}{\sqrt{2\alpha}} + o(1) \right) \cdot \sqrt{\ln N \cdot L^*}.$$

□

Remark. When using the WMG version of VPREDICT we cannot obtain a finite asymptotic canonic bound on the regret. Specifically, it can be shown that $\mathcal{T}_3 \rightarrow \infty$.

A result analogous to Corollary 4.1 also holds in the 0/1 loss case.

Corollary 5.1. *Let $\epsilon > 0$ be given. Then, the 0/1 loss variant of algorithm WMVP, applied with WMR, can be operated such that its asymptotic regret (for sufficiently large L^*) is bounded above as*

$$L_{\text{WMVP}} - L^* < \left(\frac{3}{\sqrt{2}} + \epsilon \right) \cdot \sqrt{\ln NL^*}.$$

6. Concluding remarks

We would like to point out a number of open questions. Clearly, one of the most important goals in this line of research is to determine a tight bound on expert advice algorithms for both the 0/1 and absolute loss. While such bounds are known for other loss functions such as square loss and log-loss, these two loss functions remain hard to crack. Of course, such tight bounds for *finite sequences* is the main challenge, but even tight *asymptotic* bounds would be challenging and perhaps revealing. The bounds of Lemma 4 (absolute loss case) and Lemma 8 (0/1 loss) provide non-asymptotic finite sequence regret bounds for WMVP. However, these bounds are of quite complex form. It would be interesting to derive non-asymptotic, finite sequence regret *canonical* bounds for WMVP.

The asymptotic bounds we prove for WMVP are based on “geometric” threshold sequences. Nevertheless, based on the asymptotic regret analysis we see that the WMVP algorithms minimize the asymptotic regret when $\alpha \rightarrow 1$ and therefore, they have a much simpler form (they perform expert weight updates after every increase in the loss of the best expert; see Corollary 4.1). It would be interesting to derive direct bounds for these simpler types of WMVP algorithms.

One of the more basic open questions in computational learning theory is how to decompose multi-class classification problems into an equivalent collection of binary problems. Such decompositions allow for using strong discriminant binary classifiers (such as Support Vector Machines) to solve multi-class problems. This problem is also interesting in the context of online learning: Consider 0/1 loss algorithms for finite alphabet predictions which work by operating binary algorithms (or even absolute loss algorithms). Is it possible to obtain for these algorithms regret bounds which are better than our 0/1 loss regret bound?

Appendix A: Some proofs

A.1. Proof of Lemma 4.2

The following proof is by Cesa-Bianchi et al. (1997).

Proof: We show that for $1 \leq t \leq \ell$,

$$|\hat{y}_t - y_t| \leq \ln \left(\frac{\sum_{j=1}^N w_{j,t}}{\sum_{j=1}^N w_{j,t+1}} \right) / 2 \ln \frac{2}{1+\beta}. \quad (32)$$

The lemma then follows by summing the above inequality for all choices of t . By the update rule and the choice of $U_\beta(\cdot)$ (Eq. (9)) we have,

$$\begin{aligned} \ln \left(\frac{\sum_{j=1}^N w_{j,t}}{\sum_{j=1}^N w_{j,t+1}} \right) &= - \ln \left(\frac{\sum_{j=1}^N w_{j,t} U_\beta(|\xi_{j,t} - y_t|)}{\sum_{j=1}^N w_{j,t}} \right) \\ &\geq - \ln \left(\frac{\sum_{j=1}^N w_{j,t} (1 - (1 - \beta)|\xi_{j,t} - y_t|)}{\sum_{j=1}^N w_{j,t}} \right). \end{aligned}$$

By writing the numerator of the right hand side as

$$\sum_{j=1}^N w_{j,t} - (1 - \beta) \left| \sum_{j=1}^N w_{j,t} \xi_{j,t} - y_t \sum_{j=1}^N w_{j,t} \right|,$$

we get that

$$- \ln \left(\frac{\sum_{j=1}^N w_{j,t} (1 - (1 - \beta)|\xi_{j,t} - y_t|)}{\sum_{j=1}^N w_{j,t}} \right) = - \ln(1 - (1 - \beta)|r_t - y_t|).$$

Noting that $y \in \{0, 1\}$, we have the following two cases: (i) $y_t = 0$, in which case $\hat{y}_t \leq -\frac{\ln(1-(1-\beta)r_t)}{2 \ln \frac{2}{1+\beta}}$; and (ii) $y_t = 1$, in which case $1 - \hat{y}_t \leq -\frac{\ln(1-(1-\beta)(1-r_t))}{2 \ln \frac{2}{1+\beta}} = -\frac{\ln(\beta+r_t-\beta r_t)}{2 \ln \frac{2}{1+\beta}}$. Therefore,

$$|\hat{y}_t - y_t| \leq -\frac{\ln(1 - (1 - \beta)|r_t - y_t|)}{2 \ln \frac{2}{1+\beta}},$$

and Eq. (32) holds. \square

A.2. Lemma A.1

Lemma A.1. Let χ be the solution of

$$W_i = \sum_{j=1}^N \prod_{k=1}^i \beta_k^{\tau_k} \beta_i^{L_j - T_i} = N \cdot \prod_{k=1}^i \beta_k^{\tau_k} \beta_i^\chi, \quad (33)$$

and let χ' be the solution of

$$W'_i = \sum_{j=1}^N \prod_{k=1}^i \beta_k^{\tau_k} \beta_{i+1}^{L_j - T_i} = N \cdot \prod_{k=1}^i \beta_k^{\tau_k} \beta_{i+1}^{\chi'}. \quad (34)$$

Then, given that $\beta_i < \beta_{i+1} < 1$, $\chi' \geq \chi$.

Proof: From the equations $N \cdot \beta_i^\chi = \sum_{j=1}^N \beta_i^{m_j}$ and $N \cdot \beta_{i+1}^{\chi'} = \sum_{j=1}^N \beta_{i+1}^{m_j}$, we get

$$\begin{aligned} \chi &= \frac{\ln \frac{\sum_{j=1}^N \beta_i^{m_j}}{N}}{\ln \beta_i} = \frac{\ln N - \ln \sum_{j=1}^N \beta_i^{m_j}}{\ln \frac{1}{\beta_i}}, \\ \chi' &= \frac{\ln N - \ln \sum_{j=1}^N \beta_{i+1}^{m_j}}{\ln \frac{1}{\beta_{i+1}}}. \end{aligned}$$

Consider the function

$$y(x) = \frac{\ln N - \ln \sum_{j=1}^N x^{m_j}}{\ln \frac{1}{x}}.$$

Clearly, $\chi = y(\beta_i)$ and $\chi' = y(\beta_{i+1})$. To complete the proof it is thus sufficient to show that y is monotone non-decreasing in the interval $[0, 1]$. The derivative of y is

$$\begin{aligned} y'(x) &= \frac{-\frac{\sum_{j=1}^N m_j \cdot x^{m_j-1}}{\sum_{j=1}^N x^{m_j}} \cdot \ln \frac{1}{x} + \frac{1}{x} \cdot (\ln N - \ln \sum_{j=1}^N x^{m_j})}{\ln^2 \frac{1}{x}} \\ &= \frac{\frac{1}{x} \cdot \left(\frac{\sum_{j=1}^N m_j \cdot x^{m_j}}{\sum_{j=1}^N x^{m_j}} \cdot \ln x + \ln N - \ln \sum_{j=1}^N x^{m_j} \right)}{\ln^2 \frac{1}{x}}. \end{aligned} \quad (35)$$

Denote by B the numerator of (35),

$$B = \frac{\sum_{j=1}^N m_j \cdot x^{m_j}}{\sum_{j=1}^N x^{m_j}} \cdot \ln x + \ln N - \ln \sum_{j=1}^N x^{m_j}.$$

Let $u_j = x^{m_j}$ and let $p_j = u_j / (u_1 + \dots + u_N)$. Then

$$\begin{aligned} B &= \sum_j p_j \ln u_j - \ln \sum_j u_j + \ln N \\ &= \sum_j p_j \ln p_j + \ln N \\ &= -\mathcal{H}(p) + \ln N \geq 0 \end{aligned}$$

where $\mathcal{H}(\cdot)$ is the Shannon entropy. The last inequality holds since the entropy of a discrete random variable is at most the logarithm of the size of its range. Since B is non negative we get that $y'(x) \geq 0$. \square

A.3. *Lemma A.2 (Applied in Theorem 4.1)*

Lemma A.2.

$$\begin{aligned} \lim_{k \rightarrow \infty} \mathcal{I}_1 &\leq \sqrt{\alpha/2}; \\ \lim_{k \rightarrow \infty} \mathcal{I}_2 &= 0; \\ \lim_{k \rightarrow \infty} \mathcal{I}_3 &\leq \frac{\sqrt{\alpha} + 1}{2\sqrt{2\alpha}}. \end{aligned}$$

Remark. Throughout the limit calculations in this appendix, we use without special mention the following elementary result: Let $f(x)$ be a real-valued function. If $a_n = f(n)$ for every natural n , and $\lim_{x \rightarrow \infty} f(x) = L$, then $\lim_{n \rightarrow \infty} a_n = L$.

Proof: Recall that

$$\begin{aligned} \mathcal{I}_1 &= \frac{\ln N \cdot B_k}{\sqrt{(T_{k-1} + z) \cdot \ln N}} = \frac{\sqrt{\ln N} \cdot B_k}{\sqrt{(T_{k-1} + z)}}; \\ \mathcal{I}_2 &= \frac{\ln N \cdot \sum_{i=1}^{k-1} (B_i - \frac{\ln \beta_{i+1}}{\ln \beta_i} \cdot B_{i+1})}{\sqrt{(T_{k-1} + z) \cdot \ln N}} = \frac{\sqrt{\ln N} \cdot \sum_{i=1}^{k-1} (B_i - \frac{\ln \beta_{i+1}}{\ln \beta_i} \cdot B_{i+1})}{\sqrt{(T_{k-1} + z)}}; \\ \mathcal{I}_3 &= \frac{-\sum_{i=1}^{k-1} \tau_i \cdot (\ln \beta_i B_i + 1) - z \cdot (\ln \beta_k B_k + 1)}{\sqrt{(T_{k-1} + z) \cdot \ln N}}. \end{aligned}$$

The case of \mathcal{I}_3 . Noting that $T_0 = 0$ we represent the denominator of \mathcal{I}_3 as a summation

$$\begin{aligned} &\sqrt{(T_{k-1} + z) \ln N} \\ &= \sum_{i=1}^{k-1} (\sqrt{T_i \ln N} - \sqrt{T_{i-1} \ln N}) + \sqrt{(T_{k-1} + z) \ln N} - \sqrt{T_{k-1} \ln N}. \end{aligned}$$

Thus,

$$\mathcal{I}_3 = \frac{-\sum_{i=1}^{k-1} \tau_i \cdot (\ln \beta_i B_i + 1) - z \cdot (\ln \beta_k B_k + 1)}{\ln N \sum_{i=1}^{k-1} (\sqrt{T_i} - \sqrt{T_{i-1}}) + \sqrt{(T_{k-1} + z) \ln N} - \sqrt{T_{k-1} \ln N}}.$$

Define

$$\begin{aligned} \mathcal{J}_1 &= \frac{-\sum_{i=1}^{k-1} \tau_i \cdot (\ln \beta_i B_i + 1)}{\ln N \sum_{i=1}^{k-1} (\sqrt{T_i} - \sqrt{T_{i-1}})}; \\ \mathcal{J}_2 &= \frac{-z \cdot (\ln \beta_k B_k + 1)}{\sqrt{(T_{k-1} + z) \ln N} - \sqrt{T_{k-1} \ln N}}. \end{aligned}$$

To prove that $\lim \mathcal{I}_3 \leq \frac{\sqrt{\alpha}+1}{2\sqrt{2\alpha}}$, it is sufficient to prove that the limits of both \mathcal{J}_1 and \mathcal{J}_2 are bounded above by $\frac{\sqrt{\alpha}+1}{2\sqrt{2\alpha}}$.¹³ We start with \mathcal{J}_1 and calculate the limit of the ratio of \mathcal{J}_1 's numerator's summand to its denominator's summand,

$$\lim_{T_i \rightarrow \infty} \frac{-\tau_i \cdot (\ln \beta_i \cdot B_i + 1)}{\sqrt{T_i \ln N} - \sqrt{T_{i-1} \ln N}}. \tag{36}$$

If (36) approaches $\frac{\sqrt{\alpha}+1}{2\sqrt{2\alpha}}$, then also \mathcal{J}_1 approaches this constant.¹⁴ Expanding $\beta_i = 1/(1 + \sqrt{\frac{2 \ln N}{T_i}})$ and abbreviating $A = \sqrt{T_i}$ and $B = \sqrt{2 \ln N}$, we have

$$\beta_i = \frac{1}{1 + B/A} = \frac{A}{A + B}; \tag{37}$$

$$B_i = 1 / \left(2 \ln \frac{2}{1 + \beta_i} \right) = 1 / \left(2 \ln \frac{2}{1 + \frac{1}{1+B/A}} \right) = \frac{1}{2 \ln \frac{2(A+B)}{2A+B}}. \tag{38}$$

Using the identities $\tau_i = T_i - T_{i-1} = T_i - \frac{T_i}{\alpha} = A^2 \cdot (1 - \frac{1}{\alpha})$, and $\sqrt{T_i} - \sqrt{T_{i-1}} = A - \frac{A}{\sqrt{\alpha}} = A \cdot (1 - \frac{1}{\sqrt{\alpha}})$, we have

$$\begin{aligned} (36) &= \lim_{A \rightarrow \infty} \frac{-\tau_i \cdot \left(\frac{\ln \frac{A}{2 \ln \frac{2(A+B)}{2A+B}} + 1 \right)}{\frac{B}{\sqrt{2}} \cdot (\sqrt{T_i} - \sqrt{T_{i-1}})} \\ &= \lim_{A \rightarrow \infty} \frac{(1 - \frac{1}{\alpha}) \cdot A^2}{\frac{B}{\sqrt{2}} \cdot (1 - \frac{1}{\sqrt{\alpha}}) \cdot A} \left(-\frac{\ln \frac{A}{2 \ln \frac{2(A+B)}{2A+B}}}{2 \ln \frac{2(A+B)}{2A+B}} - 1 \right) = \frac{\sqrt{\alpha} + 1}{2\sqrt{2\alpha}}, \end{aligned} \tag{39}$$

where (39) can be established using L'Hospital's rule. To deal with \mathcal{J}_2 we introduce the abbreviation $A' = \sqrt{T_k}$. Recalling the abbreviation $B = \sqrt{2 \ln N}$, and observing that $T_{k-1} = \frac{T_k}{\alpha} = \frac{A'^2}{\alpha}$, we can therefore write (as in (37) and (38)),

$$\beta_k = \frac{A'}{A' + B} \tag{40}$$

$$B_k = 1/2 \ln \frac{2(A' + B)}{2A' + B}. \tag{41}$$

Define x to be the solution of $A'x = \sqrt{T_{k-1} + z}$. Thus, $A'x = \sqrt{\frac{A'^2}{\alpha} + z}$. Clearly, x is minimized when $z = 0$ and therefore, $x \geq \sqrt{\frac{A'^2}{\alpha}}/A' = 1/\sqrt{\alpha}$. On the other hand, x is maximized when $z = \tau_k$ and therefore $x \leq 1$. Overall we have $\frac{1}{\sqrt{\alpha}} \leq x \leq 1$.¹⁵ Noting that

$$z = A'^2 x^2 - T_{k-1} = A'^2 \left(x^2 - \frac{1}{\alpha} \right),$$

we have

$$\begin{aligned}
& \lim_{k \rightarrow \infty} \frac{-z \cdot (\ln \beta_k \cdot B_k + 1)}{\sqrt{T_{k-1} + z} \ln N - \sqrt{T_{k-1}} \ln N} \\
&= \lim_{A' \rightarrow \infty} \frac{-z \cdot \left(\frac{\ln \frac{A'}{A'+B}}{2 \ln \frac{2(A'+B)}{2A'+B}} + 1 \right)}{\frac{B}{\sqrt{2}} \cdot (\sqrt{T_{k-1} + z} - \sqrt{T_{k-1}})} \\
&= \lim_{A' \rightarrow \infty} \frac{(x^2 - \frac{1}{\alpha}) \cdot A'^2}{\frac{B}{\sqrt{2}} \cdot (x - \frac{1}{\sqrt{\alpha}}) \cdot A'} \left(-\frac{\ln \frac{A'}{A'+B}}{2 \ln \frac{2(A'+B)}{2A'+B}} - 1 \right) \\
&\leq \frac{\sqrt{\alpha} + 1}{2\sqrt{2\alpha}}, \tag{42}
\end{aligned}$$

where inequality (42) follows from

$$\frac{(x^2 - \frac{1}{\alpha}) \cdot A'^2}{\frac{B}{\sqrt{2}} \cdot (x - \frac{1}{\sqrt{\alpha}}) \cdot A'} = \frac{\sqrt{2}}{B} \cdot \left(x + \frac{1}{\sqrt{\alpha}} \right) A' \leq \frac{\sqrt{2}}{B} \cdot \left(1 + \frac{1}{\sqrt{\alpha}} \right) A'.$$

We can therefore conclude that $\lim_{k \rightarrow \infty} \mathcal{J}_2 \leq \frac{\sqrt{\alpha} + 1}{2\sqrt{2\alpha}}$, and therefore the limit of \mathcal{I}_3 is also bounded above by this limit, as claimed.

The case of \mathcal{I}_1 . We need to show that $\lim_{k \rightarrow \infty} \mathcal{I}_1 \leq \sqrt{\alpha}/\sqrt{2}$. Using the above abbreviations we write \mathcal{I}_1 as follows

$$\mathcal{I}_1 = \frac{\sqrt{\ln N} B_k}{\sqrt{T_{k-1} + z}} = \frac{B}{\sqrt{2}} \frac{1/\left(2 \ln \frac{2(A'+B)}{2A'+B}\right)}{A'x}.$$

Thus, using L'Hospital's rule,

$$\begin{aligned}
\lim_{k \rightarrow \infty} \mathcal{I}_1 &= \frac{1}{x} \lim_{A' \rightarrow \infty} \frac{B}{\sqrt{2}} \cdot \frac{\frac{1}{A'}}{2 \ln \frac{2(A'+B)}{2A'+B}} \\
&= \frac{1}{x} \lim_{A' \rightarrow \infty} \frac{B}{\sqrt{2}} \cdot \frac{-\frac{1}{A'^2}}{2 \cdot -\frac{2B}{2(A'+B)(2A'+B)}} \\
&= \frac{1}{x\sqrt{2}}.
\end{aligned}$$

Noting that $x \geq 1/\sqrt{\alpha}$ we get that $\lim \mathcal{I}_1 \leq \sqrt{\alpha}/\sqrt{2}$.

The case of \mathcal{I}_2 . Write the summand in the numerator of \mathcal{I}_2 as

$$B_i - \frac{\ln \beta_{i+1}}{\ln \beta_i} B_{i+1} = \frac{B_i \cdot \ln \beta_i - B_{i+1} \cdot \ln \beta_{i+1}}{\ln \beta_i}.$$

Using the identities (37), (38), $\beta_{i+1} = 1/(1 + \sqrt{\frac{2 \ln N}{T_{i+1}}})$, and $T_{i+1} = \alpha T_i = \alpha A^2$, we have

$$\begin{aligned} \beta_{i+1} &= \frac{A\sqrt{\alpha}}{A\sqrt{\alpha} + B}; \\ B_{i+1} &= 1/2 \ln \frac{2}{1 + \frac{A\sqrt{\alpha}}{A\sqrt{\alpha} + B}}, \end{aligned}$$

Using L'Hospital's rule it is not hard to see that

$$\lim_{\beta_i \rightarrow 1} \frac{B_i \cdot \ln \beta_i - B_{i+1} \cdot \ln \beta_{i+1}}{\ln \beta_i} = \lim_{A \rightarrow \infty} \frac{\frac{\ln \frac{A}{A+B}}{2 \ln \frac{2(A+B)}{2A+B}} - \frac{\ln \frac{A\sqrt{\alpha}}{A\sqrt{\alpha} + B}}{2 \ln \frac{2(A\sqrt{\alpha} + B)}{2A\sqrt{\alpha} + B}}}{\ln \frac{A}{A+B}} = \frac{1}{4} \left(1 - \frac{1}{\sqrt{\alpha}} \right).$$

Since the limit of the numerator's summand is a constant while the denominator's summand diverges with A , the limit of the ratio is 0. That is, $\mathcal{I}_2 \rightarrow 0$. \square

A.4. Lemma A.3 (Applied in Theorem 5.1)

Lemma A.3.

$$\begin{aligned} \lim_{k \rightarrow \infty} \mathcal{I}_1 &\leq \sqrt{\alpha/2}; \\ \lim_{k \rightarrow \infty} \mathcal{I}_2 &= 0; \\ \lim_{k \rightarrow \infty} \mathcal{I}_3 &\leq \frac{\sqrt{\alpha} + \sqrt{2}}{\sqrt{2\alpha}}. \end{aligned}$$

Proof: The proof is similar to the proof of Lemma A.2 (the differences are entailed by the different definition of B_i used in the 0/1 loss case). Recall that

$$\mathcal{I}_1 = \frac{\ln N \cdot B_k}{\sqrt{T_{k-1} \cdot \ln N}} = \frac{\sqrt{\ln N} \cdot B_k}{\sqrt{T_{k-1} + z}}; \tag{43}$$

$$\mathcal{I}_2 = \frac{\ln N \cdot \sum_{i=1}^{k-1} (B_i - \frac{\ln \beta_{i+1}}{\ln \beta_i} \cdot B_{i+1})}{\sqrt{T_{k-1} \cdot \ln N}} = \frac{\sqrt{\ln N} \cdot \sum_{i=1}^{k-1} (B_i - \frac{\ln \beta_{i+1}}{\ln \beta_i} \cdot B_{i+1})}{\sqrt{T_{k-1} + z}}; \tag{44}$$

$$\mathcal{I}_3 = \frac{-\sum_{i=1}^{k-1} \tau_i \cdot (\ln \beta_i B_i + 1) - z \cdot (\ln \beta_k B_k + 1)}{\sqrt{(T_{k-1} + z) \cdot \ln N}}. \tag{45}$$

The case of \mathcal{I}_3 . Noting that $T_0 = 0$ we represent the denominator as a summation

$$\begin{aligned} &\sqrt{(T_{k-1} + z) \ln N} \\ &= \sum_{i=1}^{k-1} (\sqrt{T_i \ln N} - \sqrt{T_{i-1} \ln N}) + \sqrt{(T_{k-1} + z) \ln N} - \sqrt{T_{k-1} \ln N}. \end{aligned}$$

Thus,

$$\mathcal{I}_3 = \frac{-\sum_{i=1}^{k-1} \tau_i \cdot (\ln \beta_i B_i + 1) - z \cdot (\ln \beta_k B_k + 1)}{\ln N \sum_{i=1}^{k-1} (\sqrt{T_i} - \sqrt{T_{i-1}}) + \sqrt{(T_{k-1} + z) \ln N} - \sqrt{T_{k-1} \ln N}}.$$

Define

$$\begin{aligned} \mathcal{J}_1 &= \frac{-\sum_{i=1}^{k-1} \tau_i \cdot (\ln \beta_i B_i + 1)}{\ln N \sum_{i=1}^{k-1} (\sqrt{T_i} - \sqrt{T_{i-1}})}; \\ \mathcal{J}_2 &= \frac{-z \cdot (\ln \beta_k B_k + 1)}{\sqrt{(T_{k-1} + z) \ln N} - \sqrt{T_{k-1} \ln N}}. \end{aligned}$$

To prove that $\lim \mathcal{I}_3 \leq \frac{\sqrt{\alpha}+1}{2\sqrt{2\alpha}}$, it is sufficient to prove that the limits of both \mathcal{J}_1 and \mathcal{J}_2 are bounded above by $\frac{\sqrt{\alpha}+1}{2\sqrt{2\alpha}}$. We start with \mathcal{J}_1 and calculate the limit of the ratio of \mathcal{J}_1 's numerator's summand to its denominator's summand,

$$\lim_{T_i \rightarrow \infty} \frac{-\tau_i \cdot (\ln \beta_i \cdot B_i + 1)}{\sqrt{T_i \ln N} - \sqrt{T_{i-1} \ln N}}. \quad (46)$$

We show that (46) approaches $\frac{\sqrt{\alpha}+1}{2\sqrt{2\alpha}}$, which means that \mathcal{J}_1 also approaches this constant. Expanding $\beta_i = 1/(1 + \sqrt{\frac{2 \ln N}{T_i}})$ and abbreviating $A = \sqrt{T_i}$ and $B = \sqrt{2 \cdot \ln N}$, we have

$$\begin{aligned} \beta_i &= \frac{1}{1 + B/A} = \frac{A}{A + B}; \\ B_i &= \frac{1}{1 - \beta_i} = \frac{1}{1 - \frac{A}{A+B}}. \end{aligned}$$

Using the identities $\tau_i = T_i - T_{i-1} = T_i - \frac{T_i}{\alpha} = A^2 \cdot (1 - \frac{1}{\alpha})$, and $\sqrt{T_i} - \sqrt{T_{i-1}} = A - \frac{A}{\sqrt{\alpha}} = A \cdot (1 - \frac{1}{\sqrt{\alpha}})$, we have

$$\begin{aligned} (46) &= \lim_{A \rightarrow \infty} \frac{-\tau_i \cdot \left(\ln \frac{A}{A+B} \frac{1}{1 - \frac{A}{A+B}} + 1 \right)}{\frac{B}{\sqrt{2}} \cdot (\sqrt{T_i} - \sqrt{T_{i-1}})} \\ &= \lim_{A \rightarrow \infty} \frac{(1 - \frac{1}{\alpha}) \cdot A^2}{\frac{B}{\sqrt{2}} \cdot (1 - \frac{1}{\sqrt{\alpha}}) \cdot A} \left(-\frac{\ln \frac{A}{A+B}}{\frac{1}{1 - \frac{A}{A+B}}} - 1 \right) = \frac{\sqrt{2}}{2} \cdot \frac{\sqrt{\alpha} + 1}{\sqrt{\alpha}}, \end{aligned} \quad (47)$$

where (47) can be established using L'Hospital's rule. To deal with \mathcal{J}_2 we introduce the abbreviation $A' = \sqrt{T_k}$. Recalling the abbreviation $B = \sqrt{2 \ln N}$, and observing that $T_{k-1} = \frac{T_k}{\alpha} = \frac{A'^2}{\alpha}$, we can therefore write,

$$\beta_k = \frac{A'}{A' + B} \quad (48)$$

$$B_k = \frac{1}{1 - \frac{A'}{A'+B}}. \quad (49)$$

Define x to be the solution of $A'x = \sqrt{T_{k-1} + z}$. Thus, $A'x = \sqrt{\frac{A^2}{\alpha} + z}$. Clearly, x is minimized when $z = 0$ and therefore, $x \geq \sqrt{\frac{A^2}{\alpha}}/A' = 1/\sqrt{\alpha}$. On the other hand, x is maximized when $z = \tau_k$ and therefore $x \leq 1$. Overall we have $\frac{1}{\sqrt{\alpha}} \leq x \leq 1$. Noting that

$$z = A^2x^2 - T_{k-1} = A^2\left(x^2 - \frac{1}{\alpha}\right),$$

we have

$$\begin{aligned} & \lim_{k \rightarrow \infty} \frac{-z \cdot (\ln \beta_k \cdot B_k + 1)}{\sqrt{T_{k-1} + z) \ln N - \sqrt{T_{k-1}} \ln N} \\ &= \lim_{A' \rightarrow \infty} \frac{-z \cdot \left(\frac{\frac{A'}{A'+B}}{1 - \frac{A'}{A'+B}} + 1\right)}{\frac{B}{\sqrt{2}} \cdot (\sqrt{T_{k-1} + z} - \sqrt{T_{k-1}})} \\ &= \lim_{A' \rightarrow \infty} \frac{(x^2 - \frac{1}{\alpha}) \cdot A^2}{\frac{B}{\sqrt{2}} \cdot (x - \frac{1}{\sqrt{\alpha}}) \cdot A'} \left(-\frac{\frac{A'}{A'+B}}{1 - \frac{A'}{A'+B}} - 1\right) \\ &\leq \frac{\sqrt{\alpha} + 1}{\sqrt{2\alpha}}, \end{aligned} \tag{50}$$

where inequality (50) follows from

$$\frac{(x^2 - \frac{1}{\alpha}) \cdot A^2}{\frac{B}{\sqrt{2}} \cdot (x - \frac{1}{\sqrt{\alpha}}) \cdot A'} = \frac{\sqrt{2}}{B} \cdot \left(x + \frac{1}{\sqrt{\alpha}}\right) A' \leq \frac{\sqrt{2}}{B} \cdot \left(1 + \frac{1}{\sqrt{\alpha}}\right) A'.$$

We can therefore conclude that $\lim_{k \rightarrow \infty} \mathcal{J}_2 \leq \frac{\sqrt{\alpha} + 1}{\sqrt{2\alpha}}$, and therefore the limit of \mathcal{I}_3 is also bounded above by this limit, as claimed.

The case of \mathcal{I}_1 . We need to show that $\lim_{k \rightarrow \infty} \mathcal{I}_1 \leq \sqrt{\alpha}/\sqrt{2}$. Using the above abbreviations we write \mathcal{I}_1 as follows

$$\mathcal{I}_1 = \frac{\sqrt{\ln N} B_k}{\sqrt{T_{k-1} + z}} = \frac{B}{\sqrt{2}} \frac{1/\left(1 - \frac{A'}{A'+B}\right)}{A'x}.$$

Thus, using L'Hospital's rule,

$$\begin{aligned} \lim_{k \rightarrow \infty} \mathcal{I}_1 &= \frac{1}{x} \lim_{A' \rightarrow \infty} \frac{B}{\sqrt{2}} \cdot \frac{\frac{1}{A'}}{\frac{B}{A'+B}} \\ &= \frac{1}{x} \lim_{A' \rightarrow \infty} \frac{B}{\sqrt{2}} \cdot \frac{-\frac{1}{A^2}}{-\frac{B}{(A'+B)^2}} \\ &= \frac{1}{x\sqrt{2}}. \end{aligned}$$

Noting that $x \geq 1/\sqrt{\alpha}$ we get that $\lim \mathcal{I}_1 \leq \sqrt{\alpha/2}$.

The case of \mathcal{I}_2 . Write the summand in the numerator of \mathcal{I}_2 as

$$B_i - \frac{\ln \beta_{i+1}}{\ln \beta_i} B_{i+1} = \frac{B_i \cdot \ln \beta_i - B_{i+1} \cdot \ln \beta_{i+1}}{\ln \beta_i}.$$

Noting that

$$\beta_{i+1} = \frac{A\sqrt{\alpha}}{A\sqrt{\alpha} + B};$$

$$B_{i+1} = \frac{1}{1 - \frac{A\sqrt{\alpha}}{A\sqrt{\alpha} + B}},$$

it is not hard calculate, using L'Hospital's rule, that

$$\lim_{A \rightarrow \infty} \frac{B_i \cdot \ln \beta_i - B_{i+1} \cdot \ln \beta_{i+1}}{\ln \beta_i} = \lim_{A \rightarrow \infty} \frac{\frac{\ln \frac{A}{A+B}}{1 - \frac{A}{A+B}} - \frac{\ln \frac{A\sqrt{\alpha}}{A\sqrt{\alpha} + B}}{1 - \frac{A\sqrt{\alpha}}{A\sqrt{\alpha} + B}}}{\ln \frac{A}{A+B}} = \frac{1}{2} \left(1 - \frac{1}{\alpha} \right).$$

Since the limit of the numerator's summand is constant, and the denominator's summand diverges with A , the limit of the ratio is 0. That is, $\mathcal{I}_2 \rightarrow 0$. \square

Acknowledgments

We thank Avrim Blum, Ron Meir and Neri Merhav for their useful comments. We also thank the anonymous referees for their good comments. We are grateful to Sandy Irani for a useful idea, which corrected an error in an earlier version of the proof of Theorem 3.2.

Notes

1. When analyzing randomized expert advice algorithms we assume an "oblivious adversary", which is oblivious of the random choices made by the randomized algorithm; see Borodin and El-Yaniv (1998, Chap. 7) for a more comprehensive discussion on types of adversaries.
2. Although the absolute loss bound is not " (c, η) realizable" (since $c_L = \infty$ for the absolute loss), given the performance of the best expert (L^*) in advance, the η parameter can be calculated for any given c , such that the algorithm will never fail.
3. This a is the C coefficient of the "canonical" bound mentioned earlier.
4. In the case of the generic algorithm of Section 2.2 β is determined by $e^{-\eta}$.
5. The optimal factor is ≈ 2.6 , when selecting the best coefficients for the algorithm.
6. The non-asymptotic (finite sequence) regret bound of this algorithm is $\left(\frac{\phi^{\frac{3}{2}}}{\phi-1} + \frac{0.805\sqrt{\phi}}{4a(\ln 2)(\ln \phi)} + \frac{0.805\sqrt{\phi}}{2a(\ln N)(\ln \phi)} \right) \sqrt{L^* \ln N} + \left(a + \frac{1}{2 \ln 2} \right) \ln N$, where $\phi = \frac{1+\sqrt{5}}{2}$ (the golden ratio), $c = \phi^2$ and $a \geq 0$ is any real.
7. The finite sequence regret bound of this algorithm is $2\sqrt{2L^* \ln N} + \frac{9}{2} \cdot \ln N \cdot \ln(1 + L^*) + 6 \ln N + 1$.
8. P-NORM algorithms are more generally studied in the context of classification and regression.
9. Randomized algorithms that use a single random choice (at the start) are sometimes called "barely random" (see e.g. Borodin and El-Yaniv, 1998, Chap. 2). Such algorithms can be viewed as *mixed* strategies of games in normal form (rather than *behavioral* strategies of games in extensive form).

10. The VPREDICT routine has a slightly more general functionality than is needed here, for the purposes of RDWM (in particular, VPREDICT receives and returns a trial number, and expert weights and losses). This is so because VPREDICT is later used by our stronger WMVP algorithm of Section 4, which requires this extra functionality.
11. Set $f(x) = \frac{\ln x}{2 \ln \frac{1+x}{1-x}}$. Then $f(x)$ is monotone increasing, since it is a ratio of a monotone increasing function and a monotone decreasing function. Now, $\beta_{i+1} > \beta_i \Rightarrow \frac{\ln \beta_{i+1}}{2 \ln \frac{1+\beta_{i+1}}{1-\beta_{i+1}}} > \frac{\ln \beta_i}{2 \ln \frac{1+\beta_i}{1-\beta_i}} \Rightarrow B_{i+1} \cdot \ln \beta_{i+1} > B_i \cdot \ln \beta_i$. Since $0 < \beta_i < \beta_{i+1} < 1$, we know that $B_{i+1} \frac{\ln \beta_{i+1}}{\ln \beta_i} < B_i$.
12. Also WMG can be applied in the multi-valued alphabet setting, but indirectly. This can be done using “standard” or more sophisticated decompositions of the problem into a number of binary subproblems.
13. It is easy to see that for non-negative real sequences a_n, b_n, c_n and d_n , if $\lim_{n \rightarrow \infty} \frac{a_n}{b_n} \leq L_1$ and $\lim_{n \rightarrow \infty} \frac{c_n}{d_n} \leq L_2$, then $\lim_{n \rightarrow \infty} \frac{a_n + c_n}{b_n + d_n} \leq \max\{L_1, L_2\}$.
14. If $\{a_i\}$ and $\{b_i\}$ are two real sequences satisfying (i) $b_i > 0$ for all i ; $\lim_{n \rightarrow \infty} \sum_{i=1}^n b_i = \infty$; and (iii) $\lim_{i \rightarrow \infty} \frac{a_i}{b_i} = L$, then $\lim_{n \rightarrow \infty} \frac{\sum_{i=1}^n a_i}{\sum_{i=1}^n b_i} = L$.
15. The lower bound on x is later used when analyzing \mathcal{I}_1 .

References

- Auer, P., Cesa-Bianchi, N., & Gentile, C. (2002). Adaptive and self-confident on-line learning algorithms. *Journal of Computer and System Sciences*, 64:1, 48–75.
- Blackwell, D. (1956). An analog of the minimax theorem for vector payoffs. *Pacific J. Math.*, 6, 1–8.
- Blum, A. (1996). On-line algorithms in machine learning. Dagstuhl workshop on On-Line algorithms.
- Blum, A. (1997). Empirical support for winnow and weighted-majority algorithms: Results on a calendar scheduling domain. *Machine Learning*, 26, 5–23.
- Blum, A., & Burch, C. (1997). On-line learning and the metrical task system problem. In *Proceedings of the 10th Annual Conference on Computational Learning Theory* (pp. 45–53).
- Blum, A., Burch, C., & Kalai, A. (1989). Finely-competitive paging. *FOCS*.
- Borodin, A., & El-Yaniv, R. (1998). *Online Computation and Competitive Analysis*. Cambridge University Press.
- Cesa-Bianchi, N., Freund, Y., Haussler, D., Helmbold, D., Schapire, R., & Warmuth, M. (1997). How to use expert advice. *Journal of the ACM*, 44:3, 427–485.
- Chernoff, H. (1954). Rational selection of decision functions. *Econometrica*, 22, 422–443.
- Cover, T. (1991). Universal portfolios. *Mathematical Finance*, 1:1, 1–29.
- DeSantis, A., Markowsky, G., & Wegman, M. (1988). Learning probabilistic prediction functions. In *FOCS* (pp. 110–119).
- Feder, M., Merhav, N., & Gutman, M. (1992). Universal prediction of individual sequences. *IEEE Transactions on Information Theory*, 38, 1258–1270.
- Freund, Y., & Schapire R. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:1, 119–139.
- Gentile, C., & Littlestone, N. (1999). *The robustness of the p-norm algorithms* (pp. 1–11).
- Haussler, D., Kivinen, J., & Warmuth, M. (1998). Sequential prediction of individual sequences under general loss functions. *IEEE Transactions on Information Theory*, 44:5, 1906–1925.
- Helmbold, D., Long, D., & Sherrod, B. (1996). A dynamic disk spin-down technique for mobile computing. In *Proceedings of the Second Annual ACM International Conference on Mobile Computing and Networking*. ACM/IEEE.
- Helmbold, D., & Schapire, R. (1995). Predicting nearly as well as the best pruning of a decision tree. In *COLT*.
- Littlestone, N., & Warmuth, M. (1994). The weighted majority algorithm. *Information and Computation*, 108:2, 212–261.
- Milnor, J. (1954). Games against nature. In R. Thrall, C. Coombs, & R. Davis (Eds.), *Decision Processes* (pp. 49–60). New York and London: Wiley and Chapman & Hall.
- Sleator, D., & Tarjan, R. (1985). Amortized efficiency of list update and paging rules. 28:2, 202–208.

Vovk, V. (1990). Aggregating strategies. In *Proceedings of the Third Annual Workshop on Computational Learning Theory* (pp. 371–383).

Received January 21, 2002

Revised July 18, 2003

Accepted July 18, 2003

Final manuscript August 2, 2003