Continuous Experts and the Binning Algorithm

Jacob Abernethy[1], John Langford[1], and Manfred K. Warmuth[2,3]

[1] Toyota Technological Institute, Chicago
[2] University of California at Santa Cruz
[3] Supported by NSF grant CCR CCR 9821087
{jabernethy,jl}@tti-c.org, manfred@cse.ucsc.edu

Abstract. We consider the design of online master algorithms for combining the predictions from a set of experts where the absolute loss of the master is to be close to the absolute loss of the best expert. For the case when the master must produce binary predictions, the Binomial Weighting algorithm is known to be optimal when the number of experts is large. It has remained an open problem how to design master algorithms based on binomial weights when the predictions of the master are allowed to be real valued. In this paper we provide such an algorithm and call it the Binning algorithm because it maintains experts in an array of bins. We show that this algorithm is optimal in a relaxed setting in which we consider experts as continuous quantities. The algorithm is efficient and near-optimal in the standard experts setting.

1 Introduction

A large number of on-line learning algorithms have been developed for the so-called *expert setting* [LW94,Vov90,CBFH+97,CBFHW96,HKW98]: learning proceeds in trials; at each trial the master algorithm combines the predictions from the experts to form its own prediction; finally a label is received and both the experts and master incur a loss that quantifies the discrepancy between the predictions and the label. The goal of the master is to predict as well as the best expert. In this paper we focus on the absolute loss when the predictions of the experts and the labels are binary in $\{0, 1\}$, but the prediction of the master can be continuous in the range $[0, 1]$.[4]

Perhaps the simplest expert algorithm is the Halving algorithm: the master predicts with the majority of experts and, whenever the majority is wrong, the incorrect experts are eliminated. If there is at least one expert that never errs then this algorithm makes at most $\log_2 n$ mistakes, where n is the number of experts.

Master algorithms often maintain a weight for each expert that represent the "belief" that the expert is best. In the Halving algorithm the weights of all consistent experts are uniform and the weights of inconsistent experts immediately drop to zero. When there is no consistent expert for the sequence of trials, then a

[4] The loss $|\hat{y} - y|$ (for prediction $\hat{y} \in [0, 1]$ and label $y \in \{0, 1\}$) equals $\mathbb{E}(|Z - y|)$ for a binary valued prediction random variable Z for which $\mathbb{E}(Z) = \hat{y}$.

more gradual decay of the weights is needed. Most expert algorithms (such as the Weighted Majority (WM) algorithm [LW94] and Vovk's Aggregate algorithms [Vov90]) use exponentially decaying weights, i.e. the weights are proportional to $\exp(-\eta L_i)$, where L_i is the total loss of expert i and η a non-negative learning rate. If we assume that there is an expert that makes at most k mistakes, then large k (high noise) requires small η and small k (low noise) high η. In the Halving algorithm $k = 0$ and $\eta = \infty$.

A superior algorithm, Binomial Weighting (BW), uses binomial weights for the experts [CBFHW96]. These weights are motivated by a version space argument: if an expert has $k' \leq k$ mistakes left, then it is expanded into $\binom{q^*+1}{\leq k'}$ experts[5], where q^* is a bound on the number of remaining mistakes of the master. In each of the expansions, at most k' of the q^* trials are chosen in which the expanded expert negates its prediction. We now can run the Halving algorithm on the set of all expanded experts. However this argument requires that the number of mistakes q^* of the master is bounded. This is easily achieved when the master makes binary predictions and incurs units of loss. In that case, all trials in which the master predicts correctly can be ignored and in trials when the master makes a mistake, at least half of the expanded experts are eliminated and there can't be too many such trials.

Restricting the master to use binary predictions is a significant handicap as it does not allow the algorithm to hedge effectively when the experts produce a relatively even vote. In this case, the master prefers to predict .5 instead of predicting[6] 0 or 1. The main open problem posed in [CBFHW96] is the question of how the fancier binomial weights can be used in the case when the master's predictions lie in $[0, 1]$. In that case there are no good bounds on the number of trials because now all trials in which the master incurs any loss need to be counted.

In this paper we provide such a prediction strategy, called `Binning`, and we show that this strategy is essentially optimal. We generalize the standard experts setting to consider experts as continuous quantities: we allow each expert to split itself into parts r and $1-r$, where part r of the expert predicts 1 and part $1-r$ predicts 0. Intuitively, the relaxation to continuous quantities of experts removes the discretization effects that make the computation of the optimal strategy difficult.

In our approach we consider an associated game where the master plays against an adversary who controls the predictions of the experts and the outcomes of every round to maximize the master's total loss. We show that for this relaxed setting the adversary can always play optimally by splitting all remaining experts in half.

[5] This number of expansions is the current binomial weight of the expert. Exponential weights always change by a fixed factor $exp(-\eta)$ in case of a mistake. However the update factors to the binomial weights "cool down" in subtle ways as k' gets close to k and q^* decreases.

[6] As discussed before predicting .5 is equivalent to predicting randomly.

`Binning` is very similar to the Binomial Weighting algorithm (BW) in that it implicitly uses binomial weights. In the case of exponential weights, the bound for the algorithm with predictions in $[0,1]$ (i.e. Vovks aggregating algorithm for the absolute loss) is half of the bound for the algorithm with binary predictions (i.e. the WM algorithm). Essentially the same happens for binomial weights. The bound we prove for the new Binning algorithm (predictions in $[0,1]$) is a constant plus half of the bound of the BW algorithm (predictions in $\{0,1\}$) and the additive constant is well behaved. It is already known that for large n, BW is optimal when the prediction of the master must be binary. Since no algorithm with predictions in $[0,1]$ can be better that half of the best binary prediction algorithm, our new `Binning` algorithm is essentially optimal.

Summary of paper: We begin in Section 2 by introducing our notation and formulating the optimal prediction strategy of the expert's game when the master uses predictions in $[0,1]$. In Section 3 we define the continuous game in which we allow continuous quantities of experts. We show the following is always an optimal split in each trial of the continuous game: all experts split themselves in half (resulting in experts of size $\frac{1}{2^i}$). We also relate the optimal algorithm for this game (i.e. the new Binning algorithm) to the BW algorithm. In Section 4 we give some experimental results showing `Binning`'s performance and its worst case bound on real world datasets. Finally, in Section 5 we discuss various open problems as well as high level goals that might be attainable with our new continuous-experts technique.

2 The Optimal Prediction Strategy

We have n experts that make binary predictions and we are to design a master algorithm that combines the predictions of the experts with the goal of performing well compared to the best expert. Our on-line learning model can be viewed as a game that proceeds in trials. At each trial the following occurs: first, each expert i produces a prediction $x_i \in \{0,1\}$; then the master produces a prediction $\hat{y} \in [0,1]$ and finally, the true label $y \in \{0,1\}$ is received and both the experts and the master incur a loss: expert i incurs loss $|x_i - y|$ and the master loss $|\hat{y} - y|$. Recall that the experts' predictions and the labels are binary, but the prediction of the master lies in $[0,1]$. (Generalizations are discussed in the conclusion).

The two parties of the game are nature and the master: nature provides the predictions of the experts, the master gives a prediction in each trial, and finally nature provides the true label. We must restrict the adversary, since an unrestricted adversary can continue to inflict loss at least $\frac{1}{2}$ in each round. A common restriction is the following: the true labels and the choices of the experts' predictions have to be such that at least one expert has total loss at most k. We call this restriction the *k-mistake rule*. It is assumed that k is known to both the master and the adversary before the game starts.

Notice that, with the addition of the k-mistake rule, the game is essentially finite. If, after many rounds, all but one expert has made more than k mistakes, and the last expert has made exactly k, then this expert is required to predict correctly from this point on. In this case, the master can simply mimic the prediction of this expert, and the adversary cannot allow this expert to err. Since this simple strategy of the master assures that the master incurs no future loss, the game has ended.

Observe that, after a number of trials, the only relevant information about the past is the number of experts that made $0, 1, \ldots, k$ mistakes. The experts can then be partitioned into $k+1$ bins and the past can therefore be summarized by the state vector $\mathbf{s} = (s_0, \ldots, s_k)$, where s_i is the number of experts that has made i mistakes. We also use the notation $|\mathbf{s}| := |\mathbf{s}|_1 = \sum_{i=0}^{k} s_i$ for the total number of experts. It is possible that $|\mathbf{s}| \leq n$, as some experts might have incurred already more than k mistakes and therefore will be ignored. Our state space is therefore the set $\mathcal{S} = \{\mathbf{s} \in \{0, 1, \ldots, n\}^k : |\mathbf{s}| \leq n\}$.

By choosing binary predictions for the experts, the adversary splits the state vector $\mathbf{s}$ into $\mathbf{r}$ and $\mathbf{s} - \mathbf{r}$ such that $\mathbf{r} \leq \mathbf{s}$ and $\mathbf{r} \in \mathcal{S}$. The vector $\mathbf{r}$ represents the experts that predict one and the vector $\mathbf{s} - \mathbf{r}$ the experts that predict zero. After the adversary provides the binary label y, the experts that predict wrongly advance by one bin. For any state vector $\mathbf{z} \in \mathcal{S}$, we use $\mathbf{z}^+$ to denote the shifted vector $(0, z_0, z_1, \ldots, z_{k-1})$. When $y = 0$, then $\mathbf{r}$ advances, and when $y = 1$, then $\mathbf{s} - \mathbf{r}$ does, and the successor state is

$$\mathbf{s}^{\mathbf{r},y} = \begin{cases} \mathbf{r}^+ + (\mathbf{s} - \mathbf{r}) \text{ if } y = 0 \\ \mathbf{r} + (\mathbf{s} - \mathbf{r})^+ \text{ if } y = 1. \end{cases}$$

Let's give an example of one round of our game. Assume the mistake bound k is 2 and the number of experts n is 6. Initially, our state vector is $(6, 0, 0)$. However, after several rounds, 2 experts have made no mistakes, 1 expert has made 2 mistakes, and 3 experts have made more than 2 mistakes. Our state vector is now $(2, 0, 1)$. On the next round we receive predictions from the experts, and we find that the only expert to predict 1 was one of the experts with no mistakes. In this case, the split of the state vector is

$$\mathbf{s} + (2, 0, 1) = \overbrace{(1, 0, 0)}^{\mathbf{r}} + \overbrace{(1, 0, 1)}^{\mathbf{s}-\mathbf{r}},$$

and the two resulting possible states would be

$$\begin{aligned} \mathbf{s}^{\mathbf{r},0} &= \overbrace{(0, 1, 0)}^{\mathbf{r}^+} + \overbrace{(1, 0, 1)}^{\mathbf{s}-\mathbf{r}} = (1, 1, 1) \\ \mathbf{s}^{\mathbf{r},1} &= \underbrace{(1, 0, 0)}_{\mathbf{r}} + \underbrace{(0, 1, 0)}_{(\mathbf{s}-\mathbf{r})^+} = (1, 1, 0). \end{aligned}$$

2.1 The Value of the Game

We define the value of our game at a state $\mathbf{s}$ as the total loss the adversary can force the master to incur if its choices satisfy the k-mistake rule. With the above notation we can express the value of the game as:

$$\ell(\mathbf{s}) := \begin{cases} -\infty & \text{if } \mathbf{s} = \mathbf{0} \\ 0 & \text{if } \mathbf{s} = (0, \ldots, 0, 1) \\ \max\limits_{\mathbf{r} \le \mathbf{s}, \mathbf{r} \in \mathcal{S}} \min\limits_{\hat{y} \in [0,1]} \max\limits_{y \in \{0,1\}} \left(|\hat{y} - y| + \ell(\mathbf{s}^{\mathbf{r},y}) \right) & \mathbf{s} \in \text{ rest of } \mathcal{S}. \end{cases} \tag{1}$$

At various points in the paper we induct on the *mistake budget* $B(\mathbf{s})$ of a state, which we define as the total number of mistakes that can be made by all of the experts before arriving in the final state $\mathbf{b}_k = (0, ..., 0, 1)$. Explicitly, $B(\mathbf{s}) := -1 + \sum_{i=0}^{k} (k - i + 1) s_i$. Notice, $B(\mathbf{0}) = -1$ and if $B(\mathbf{s}) = 0$ then $\mathbf{s}$ must be the final state $\mathbf{b}_k$.

Some care needs to be taken to assure that the above game is well defined because of the possibility that all of the experts make the same prediction. If a split is *unanimous*, i.e. all experts predict $u = 1$ and $\mathbf{r} = \mathbf{s}$, or all experts predict $u = 0$ and $\mathbf{r} = \mathbf{0}$, then the master must choose $\hat{y}$ as the unanimous label u. If the master chose $\hat{y} \neq u$, the adversary would simply choose $y = u$, inflicting positive loss $|\hat{y} - u|$ on the master while the experts incur no mistakes. Therefore whenever all experts predict with some unanimous label u, the optimal choice of the master is $\hat{y} = u$ also.

How should the adversary choose its label y when all experts predict with some unanimous label u and $\hat{y} = u$? If $y = u$ then the current trial is vacuous because $\mathbf{s}^{\mathbf{r},y} = \mathbf{s}$ and none of the parties incur any loss. We need to make the mild assumption that such vacuous trials are disallowed. Therefore $y \neq u$ in unanimous trials and in this case the successor state is $\mathbf{s}^{\mathbf{r},y} = \mathbf{s}^{+}$. In summary,

$$\mathbf{r} \in \{\mathbf{0}, \mathbf{s}\} \Longrightarrow \min_{\hat{y} \in [0,1]} \max_{y \in \{0,1\}} \left(|\hat{y} - y| + \ell(\mathbf{s}^{\mathbf{r},y}) \right) = 1 + \ell(\mathbf{s}^{+}).$$

We now expand the recurrence slightly by rewriting (1) as follows. As before, $\ell(\mathbf{0}) = -\infty$ and $\ell(\mathbf{b}_k) = 0$, and for every other state $\mathbf{s} \in \mathcal{S}$,

$$\ell(\mathbf{s}) = \max_{\substack{\mathbf{r} \le \mathbf{s} \\ \mathbf{r} \in S}} \begin{cases} 1 + \ell(\mathbf{s}^{+}) & \text{if } \mathbf{r} \in \{\mathbf{0}, \mathbf{s}\} \\ \max\{\ell(\mathbf{s}^{\mathbf{r},0}), \ell(\mathbf{s}^{\mathbf{r},1})\} & \text{if } \mathbf{r} \notin \{\mathbf{0}, \mathbf{s}\}, |\ell(\mathbf{s}^{\mathbf{r},1}) - \ell(\mathbf{s}^{\mathbf{r},0})| > 1 \\ \frac{\ell(\mathbf{s}^{\mathbf{r},1}) + \ell(\mathbf{s}^{\mathbf{r},0}) + 1}{2} & \text{if } \mathbf{r} \notin \{\mathbf{0}, \mathbf{s}\}, |\ell(\mathbf{s}^{\mathbf{r},1}) - \ell(\mathbf{s}^{\mathbf{r},0})| \le 1 \end{cases} \tag{2}$$

The unanimous case (i.e. when $\mathbf{r} = 0, \mathbf{s}$) follows from the above discussion. The remaining two cases arise when we consider how the game is played once $\mathbf{r}$ has been chosen. The master algorithm wants to choose $\hat{y}$, while knowing that the adversary can simply choose the larger of $|\hat{y} - y| + \ell(\mathbf{s}^{\mathbf{r},y})$ for $y \in \{0, 1\}$. Thus it would like to minimize $L(\mathbf{r}, \hat{y}) := \max\{\hat{y} + \ell(\mathbf{s}^{\mathbf{r},0}), 1 - \hat{y} + \ell(\mathbf{s}^{\mathbf{r},1})\}$. It can accomplish this by making these two quantities as close as possible. When $|\ell(\mathbf{s}^{\mathbf{r},0}) - \ell(\mathbf{s}^{\mathbf{r},1})| \le 1$, it can make them exactly equal by setting $\hat{y} = \frac{\ell(\mathbf{s}^{\mathbf{r},1}) - \ell(\mathbf{s}^{\mathbf{r},0}) + 1}{2}$. However, when $\ell(\mathbf{s}^{\mathbf{r},1}) > \ell(\mathbf{s}^{\mathbf{r},0}) + 1$, the master should choose $\hat{y} = 1$ and $L(\mathbf{r}, \hat{y}) = \ell(\mathbf{s}^{\mathbf{r},1})$. Similarly when $\ell(\mathbf{s}^{\mathbf{r},0}) > \ell(\mathbf{s}^{\mathbf{r},1}) + 1$ then $\hat{y} = 0$ and $L(\mathbf{r}, \hat{y}) = \ell(\mathbf{s}^{\mathbf{r},0})$. The latter two cases are summarized in line 2 of above recursion, completing the argument that the above recursion is equivalent to (1).

Algorithm 1 `OptPredict`$_{\mathtt{k}}$

At round t, the number of past mistakes of the experts are tallied in the state vector $\mathbf{s}$ and the experts that predict 1 (respectively 0) are tallied in the split vector $\mathbf{r}$ (respectively $\mathbf{s} - \mathbf{r}$). The master algorithm `OptPredict`$_{\mathtt{k}}$ outputs prediction:

$$\hat{y} = clip\left(\frac{\ell(\mathbf{s}^{\mathbf{r},1}) - \ell(\mathbf{s}^{\mathbf{r},0}) + 1}{2}\right), \tag{4}$$

where $clip(x)$ is the point in $[0, 1]$ closest to x.

A further and more subtle simplification of the above recurrence is provided by the following lemma which rules out the first two lines of (2). In particular, it implies two useful facts: (a) unanimous splits only occur when there is a single expert and (b) when $|\mathbf{s}| > 1$, the adversary will never choose $\mathbf{r}$ so that $|\ell(\mathbf{s}^{\mathbf{r},1}) - \ell(\mathbf{s}^{\mathbf{r},0})| > 1$.

Note that when there is exactly one expert left which has made $i \leq k$ mistakes, i.e. $\mathbf{s}$ is the standard basis vector $\mathbf{b}_i$, then all splits must be unanimous and $\ell(\mathbf{b}_i) = k - i$.

Lemma 1. *For any state* $\mathbf{s} \in \mathcal{S}$ *s.t.* $|\mathbf{s}| > 1$,

$$\ell(\mathbf{s}) = \max_{0<\mathbf{r}<\mathbf{s}} \left\{\frac{\ell(\mathbf{s}^{\mathbf{r},0}) + \ell(\mathbf{s}^{\mathbf{r},1}) + 1}{2}\right\}. \tag{3}$$

The proof requires a rather technical induction and is given in the appendix.

We now have a very simple recursion for computing ℓ, and we can easily define the optimal prediction strategy `OptPredict`$_{\mathtt{k}}$, as in (4), with oracle access to this function. Unfortunately, ℓ is still too expensive to compute: it requires the solution of a dynamic programming problem over $O(|\mathcal{S}|) = O((n + 1)^k)$ states.

3 The Continuous Game

Computing the value function ℓ using (3) is too difficult: at every round we must consider all possible splits $\mathbf{r}$. However, empirical observations have suggested that splits $\mathbf{r}$ close to $\frac{\mathbf{s}}{2}$ are optimal for the adversary. In particular, we have observed that, whenever $\mathbf{s}$ has only even entries, the split $\mathbf{r} = \frac{\mathbf{s}}{2}$ is always optimal. This evidence leads one to believe that an optimal adversarial strategy is to evenly divide the experts, thus balancing the loss value of the two successor states as much as possible. Unfortunately, this is not always possible when the experts come in discrete units.

We therefore develop a continuous game that always allows for such "even splits" and show that the value of this continuous game is easy to compute and tightly upper bounds the value function of the original game. The continuous game follows the same on-line protocol given in the first paragraph of Section 2, however now each expert has a mass in $[0, 1]$ and at each trial each expert is allowed to split itself into two parts which predict with opposite labels. The total mass of the two parts must equal the original mass.

3.1 The Continuous Experts Setting

As in the discrete game, the state of the new game is again summarized by a vector, i.e. it is not necessary to keep track of the identities of the individual experts of mass in $[0,1]$. The state space of the new game is $\widetilde{\mathcal{S}} = \{\mathbf{s} \in [0,n]^{k+1} : |\mathbf{s}| \leq n\}$. The initial state vector is again $(n, 0, \dots, 0)$, but the split $\mathbf{r}$ may have non-negative real valued components.

We now define a new value function $\widetilde{\ell}$ on $\widetilde{\mathcal{S}}$. We would like the definition of $\widetilde{\ell}$ to mimic the recursive definition of ℓ in (3), yet we must be careful to define the "base case" of this recursion. In particular, how do we redefine the k-mistake rule within this "continuous experts" setting? We require the following constraints on $\widetilde{\ell}$: when $|\mathbf{s}| < 1$, which we call an *infeasible* state, we define $\widetilde{\ell}(\mathbf{s}) := -\infty$; when $|\mathbf{s}| \geq 1$, i.e. a total of one unit of experts remains, then $\widetilde{\ell}(\mathbf{s})$ should be non-negative.

These two requirements are not sufficient: we must consider the case when we are at a feasible state $\mathbf{s}$ in which, given any split $\mathbf{r}$ that the adversary chooses, at least one of the successor states is infeasible. This would be a state where the game has effectively ended, and we will therefore consider it a base case of our recursion. That is, the recursive definition in (3) would not be appropriate on such a state, for $\widetilde{\ell}(\mathbf{s}^{\mathbf{r},0})$ or $\widetilde{\ell}(\mathbf{s}^{\mathbf{r},1})$ is $-\infty$ (for any $\mathbf{r}$), and we must therefore fix the value $\widetilde{\ell}(\mathbf{s})$. We let S_0 denote the set of such base-case states:

$$\begin{aligned}
S_0 &= \left\{\mathbf{s} : |\mathbf{s}| \geq 1 \text{ and } \forall \mathbf{0} < \mathbf{r} < \mathbf{s} : |\mathbf{s}^{\mathbf{r},0}| < 1 \text{ or } |\mathbf{s}^{\mathbf{r},1}| < 1\right\} \\
&= \{\mathbf{s} : |\mathbf{s}| \geq 1 \text{ and } \forall \mathbf{0} < \mathbf{r} < \mathbf{s} : |\mathbf{s}| - r_k < 1 \text{ or } |\mathbf{s}| - s_k + r_k < 1\} \\
&= \left\{\mathbf{s} : |\mathbf{s}| \geq 1 \text{ and } |\mathbf{s}| - \frac{s_k}{2} < 1\right\} \\
&= \text{convex-hull}\{\mathbf{b}_0, \dots, \mathbf{b}_k, 2\mathbf{b}_k\} \setminus \left\{\mathbf{s} : |\mathbf{s}| - \frac{s_k}{2} = 1\right\}
\end{aligned}$$

We obtain the convex hull representation because S_0 is described by $k+3$ linear constraints: $s_0 \geq 0, \dots, s_k \geq 0$, $\sum s_i \geq 1$ and $\frac{s_k}{2} + \sum_{i<k} s_i < 1$. The subsequent polytop has corners $\mathbf{b}_0, \dots, \mathbf{b}_k$ and $2\mathbf{b}_k$. The region is not exactly the convex hull since the last constraint is a strict inequality, and thus we must subtract one face. Notice that, while $\mathbf{b}_k$ lies within S_0, the remaining states $\mathbf{b}_0, \dots, \mathbf{b}_{k-1}$, and $2\mathbf{b}_k$ all lie on the subtracted face.

3.2 The Value of the Continuous Game

We are now in position to define our recursion for $\widetilde{\ell}$. For any state $\mathbf{s} \in \widetilde{\mathcal{S}}$,

$$\widetilde{\ell}(\mathbf{s}) := \begin{cases} \widetilde{\ell}_0(\mathbf{s}) & \text{if } \mathbf{s} \in S_0 \\ \displaystyle\max_{\substack{\mathbf{r} \leq \mathbf{s},\, \mathbf{r} \in \widetilde{\mathcal{S}} \\ \frac{1}{2} \leq |\mathbf{r}| \leq |\mathbf{s}| - \frac{1}{2}}} \frac{\widetilde{\ell}(\mathbf{s}^{\mathbf{r},0}) + \widetilde{\ell}(\mathbf{s}^{\mathbf{r},1}) + 1}{2} & \mathbf{s} \in \text{ rest of } \widetilde{\mathcal{S}}. \end{cases} \tag{5}$$

Note that it is crucial that in the above recurrence for $\widetilde{\ell}$ we bound[7] the split $\mathbf{r}$ away from $\mathbf{0}$ and $\mathbf{s}$. Thus whenever we recurse, at least a total of half a unit of experts is advanced and the depth of the recurrence is bounded by $2n(k+1)$.

We still need a natural definition of $\widetilde{\ell}_0(\mathbf{s})$ for states $\mathbf{s} \in S_0$. The simplest definition would be to define $\widetilde{\ell}_0$ as zero. However, this would cause the value function $\widetilde{\ell}$ to be discontinuous at the corners of the base region $\mathcal{S}_0$. Intuitively, we want $\widetilde{\ell}$ to mimic ℓ as much as possible. Thus, the two properties that we require of $\widetilde{\ell}_0$ is (a) that it agrees with ℓ on the discrete corners of S_0, and (b) that it is continuous on the "in between" states. We therefore define $\widetilde{\ell}_0$ on the interior of S_0 as the linear interpolation of ℓ on the corner states $\{\mathbf{b}_0, \ldots, \mathbf{b}_k, 2\mathbf{b}_k\}$. We can explicitly define $\widetilde{\ell}_0(\mathbf{s})$ as follows: write $\mathbf{s} = \alpha_0 \mathbf{b}_0 + \ldots + \alpha_k \mathbf{b}_k + \alpha_{k+1}(2\mathbf{b}_k)$, where $\alpha_i \in [0,1]$ and $\sum \alpha_i = 1$. Let

$$\begin{aligned}\widetilde{\ell}_0(\mathbf{s}) := \left(\sum_{i=0}^{k} \alpha_i \ell(\mathbf{b}_i)\right) + \alpha_{k+1}\ell(2\mathbf{b}_k) &= \left(\sum_{i=0}^{k} \alpha_i (k-i)\right) + \frac{\alpha_{k+1}}{2} \\ &= \sum_{i=0}^{k-1} s_i\,(k-i) + \frac{|\mathbf{s}|-1}{2}.\end{aligned}$$

We chose S_0 and $\widetilde{\ell}_0(\mathbf{s})$ so that the following lemma holds:

Lemma 2. *For all* $\mathbf{s} \in \mathcal{S}$, $\widetilde{\ell}(\mathbf{s}) \geq \ell(\mathbf{s})$.

Proof. The continuous game is identical to the discrete game, except that we have given the adversary a larger space to choose a split $\mathbf{r}$. Essentially, increasing the number of strategies for the adversary can only increase the loss of the game. □

3.3 An Optimal Adversarial Strategy

We now show that for the value function $\widetilde{\ell}$ of the continuous game, $\mathbf{r} = \mathbf{s} - \mathbf{r} = \mathbf{s}/2$ is always an optimal split for the adversary. In this case the successor state is $h(\mathbf{s}) := \frac{\mathbf{s}+\mathbf{s}^+}{2}$ no matter how $y \in \{0,1\}$ is chosen.

Define a function $\mathcal{L}$ as follows:

$$\mathcal{L}(\mathbf{s}) := \begin{cases} -\infty & \text{if } |\mathbf{s}| < 1 \\ \widetilde{\ell}_0(\mathbf{s}) & \text{if } \mathbf{s} \in S_0 \\ \frac{1}{2} + \mathcal{L}(h(\mathbf{s})) & \text{if } \mathbf{s} \text{ rest of } \widetilde{\mathcal{S}}. \end{cases} \tag{6}$$

We prove $\widetilde{\ell} = \mathcal{L}$ in two steps. The first is the following crucial lemma.

Lemma 3. *$\mathcal{L}$ is concave.*

[7] $\widetilde{\ell}$ does not change if we use instead the constraint $\epsilon \leq |\mathbf{r}| \leq |\mathbf{s}| - \epsilon$ for any ϵ with $0 < \epsilon \leq \frac{1}{2}$.

Proof. We have defined our base region S_0 above. Notice that we can rewrite S_0 as $\{\mathbf{s} \in \widetilde{\mathcal{S}} : |\mathbf{s}| \geq 1, |h(\mathbf{s})| < 1\}$. Now define S_n for $n \geq 2$ as $\{\mathbf{s} : \mathbf{s} \notin S_{n-1}$ and $h(\mathbf{s}) \in S_{n-1}\} = \{\mathbf{s} : |h^n(\mathbf{s})| > 1, |h^{n+1}(\mathbf{s})| < 1\}$. We will show that we can effectively reduce the concavity of $\mathcal{L}$ to the concavity of $\mathcal{L}$ on the region $S_0 \cup S_1$.

It suffices to show that $\mathcal{L}$ is concave on a set of convex open neighbors which cover $\widetilde{\mathcal{S}}$ since the concavity property always fails locally. Let $R_0 := S_0$, and for $n > 0$, define R_n as the interior of $S_{n-1} \cup S_n$, i.e. $\{\mathbf{s} \in \widetilde{\mathcal{S}} : |h^{n-1}(\mathbf{s})| > 1, |h^{n+1}(\mathbf{s})| < 1\}$. Let $h^0(\mathbf{s}) := \mathbf{s}$ by convention. Notice that $\cup R_n = \widetilde{\mathcal{S}}$ and each R_n is open and convex.

Let $\mathcal{L}$ restricted to R_n be denoted $\mathcal{L}|_{R_n}$. We show that $\mathcal{L}|_{R_n}$ is concave for every n. $\mathcal{L}|_{R_0}$ is certainly concave, for $\mathcal{L}$ is defined linearly on $R_0 = S_0$.

We show that $\mathcal{L}|_{R_1}$ is concave by first noting that

$$\mathcal{L}|_{R_1} = \begin{cases} \widetilde{\ell}_0(\mathbf{s}) = \sum s_i\,(k-i) + \frac{|\mathbf{s}|-1}{2} & \text{if } \mathbf{s} \in S_0 \\ \widetilde{\ell}_1(\mathbf{s}) = \widetilde{\ell}_0(h(\mathbf{s})) + \frac{1}{2} = \sum s_i\,(k-i) + \frac{s_k}{4} & \text{if } \mathbf{s} \in S_1. \end{cases} \tag{7}$$

These two linear functions are equal when $\frac{|\mathbf{s}|-1}{2} = \frac{s_k}{4} \implies |\mathbf{s}| - \frac{s_k}{2} = 1$, which is exactly the border between S_0 and S_1. Since S_0 is defined by the constraint $|\mathbf{s}| - s_k/2 < 1$, we see that $\widetilde{\ell}_0(\mathbf{s}) < \widetilde{\ell}_1(\mathbf{s})$ when $\mathbf{s} \in S_0$. These last two statements imply that $\mathcal{L}|_{R_1}(\mathbf{s}) = \min\{\widetilde{\ell}_0(\mathbf{s}), \widetilde{\ell}_1(\mathbf{s})\}$ and the minimum of two linear functions is always concave.

Assume $n > 1$. Notice, $\mathbf{s} \in R_n$ implies that $h(\mathbf{s}) \in R_{n-1}$. Thus $\mathcal{L}|_{R_n} = \mathcal{L}|_{R_{n-1}} \circ h + \frac{1}{2}$. Note that: (a) h is an orientation-preserving linear function ($\det(h) > 0$), (b) addition by a constant preserves concavity, and (c) $\mathcal{L}|_{R_{n-1}}$ is concave by induction. Therefore, $\mathcal{L}|_{R_n}$ is concave as well. □

We are now in position to prove the following theorem.

Theorem 1. *For all* $\mathbf{s} \in \widetilde{\mathcal{S}}$,

$$\widetilde{\ell}(\mathbf{s}) := \begin{cases} -\infty & \textit{if } |\mathbf{s}| < 1 \\ \widetilde{\ell}_0(\mathbf{s}) & \textit{if } \mathbf{s} \in S_0 \\ \frac{1}{2} + \widetilde{\ell}(h(\mathbf{s})) & \textit{if } \mathbf{s} \textit{ rest of } \widetilde{\mathcal{S}}. \end{cases} \tag{8}$$

Proof. We show that, for all $\mathbf{s} \in \widetilde{\mathcal{S}}$, $\widetilde{\ell}(\mathbf{s}) = \mathcal{L}(\mathbf{s})$. We induct on the mistake budget $B(\mathbf{s})$. When $B(\mathbf{s}) < \frac{1}{2}$, then $\mathbf{s} \in S_0$, and $\widetilde{\ell}$ and $\mathcal{L}$ are defined identically on S_0. Now assume that $\frac{p}{2} \leq B(\mathbf{s}) < \frac{p+1}{2}$ for some positive integer p. It is possible that $\mathbf{s} \in S_0$, in which case certainly $\widetilde{\ell}(\mathbf{s}) = \mathcal{L}(\mathbf{s})$. Otherwise,

$$\widetilde{\ell}(\mathbf{s}) = \max_{\substack{\mathbf{r} \leq \mathbf{s},\, \mathbf{r} \in \widetilde{\mathcal{S}} \\ \frac{1}{2} \leq |\mathbf{r}| \leq |\mathbf{s}| - \frac{1}{2}}} \frac{\widetilde{\ell}(\mathbf{s}^{\mathbf{r},0}) + \widetilde{\ell}(\mathbf{s}^{\mathbf{r},1}) + 1}{2}. \tag{9}$$

However, since we may only choose $\mathbf{r}$ such that $\frac{1}{2} \leq |\mathbf{r}| \leq |\mathbf{s}| - \frac{1}{2}$, it must be that $B(\mathbf{s}^{\mathbf{r},1}) < \frac{p}{2}$ and $B(\mathbf{s}^{\mathbf{r},0}) < \frac{p}{2}$. By induction, $\widetilde{\ell}(\mathbf{s}^{\mathbf{r},0}) = \mathcal{L}(\mathbf{s}^{\mathbf{r},0})$ and

$\widetilde{\ell}(\mathbf{s}^{\mathbf{r},1}) = \mathcal{L}(\mathbf{s}^{\mathbf{r},1})$. However, by the concavity of $\mathcal{L}$, we see that for any $\mathbf{r}$,

$$\begin{aligned}\frac{\widetilde{\ell}(\mathbf{s}^{\mathbf{r},0}) + \widetilde{\ell}(\mathbf{s}^{\mathbf{r},1}) + 1}{2} &= \frac{\mathcal{L}(\mathbf{s}^{\mathbf{r},0}) + \mathcal{L}(\mathbf{s}^{\mathbf{r},1}) + 1}{2} \leq \mathcal{L}\left(\frac{\mathbf{s}^{\mathbf{r},0} + \mathbf{s}^{\mathbf{r},1}}{2}\right) + \frac{1}{2} \\ &= \mathcal{L}\left(\frac{(\mathbf{s}-\mathbf{r}) + (\mathbf{r})^+ + \mathbf{r} + (\mathbf{s}-\mathbf{r})^+}{2}\right) + \frac{1}{2} \\ &= \mathcal{L}\left(\frac{\mathbf{s}+\mathbf{s}^+}{2}\right) + \frac{1}{2} = \mathcal{L}(h(\mathbf{s})) + \frac{1}{2} = \mathcal{L}(\mathbf{s})\end{aligned}$$

and thus $\mathcal{L}$ is an upper bound on $\widetilde{\ell}$. On the other hand, notice also that $B(h(\mathbf{s})) < \frac{p}{2}$ so $\widetilde{\ell}(h(\mathbf{s})) = \mathcal{L}(h(s))$. Thus, for the choice $\mathbf{r} = \frac{\mathbf{s}}{2}$, we see that

$$\widetilde{\ell}(\mathbf{s}) \geq \frac{\widetilde{\ell}(\mathbf{s}^{\mathbf{r},0}) + \widetilde{\ell}(\mathbf{s}^{\mathbf{r},1}) + 1}{2} = \frac{\widetilde{\ell}(h(\mathbf{s})) + \widetilde{\ell}(h(\mathbf{s})) + 1}{2} = \mathcal{L}(h(\mathbf{s})) + \frac{1}{2} = \mathcal{L}(\mathbf{s}),$$

and thus $\mathcal{L}$ is also a lower bound on $\widetilde{\ell}$. Therefore, $\widetilde{\ell}(\mathbf{s}) = \mathcal{L}(\mathbf{s})$ and we are done. □

Corollary 1. *The split* $\mathbf{r} = \frac{\mathbf{s}}{2}$ *is always an*[8] *optimal choice for the adversary in the continuous game.*

3.4 The Binning Algorithm

We can now define our new algorithm $\texttt{Binning}_\texttt{k}$ by simply replacing ℓ in equation (4) with the new function $\widetilde{\ell}$. We will reason below that $\widetilde{\ell}$ can be computed efficiently.

Theorem 1 tells us that, to get the value of $\widetilde{\ell}(\mathbf{s})$, we apply the function h to $\mathbf{s}$ several times until we are in the base region S_0. Let

$$q^{\mathbf{s}} := \min\{q : h^q(\mathbf{s}) \in S_0\} = \max\{q : |h^q(\mathbf{s})| \geq 1\}. \tag{10}$$

This allows us to write $\widetilde{\ell}(\mathbf{s}) = \frac{q^{\mathbf{s}}}{2} + \widetilde{\ell}_0(h^{q^{\mathbf{s}}}(\mathbf{s}))$. The function h is linear on the state space $\widetilde{\mathcal{S}}$ and can be represented as a square matrix of dimension $k+1$ (The matrix for $k=3$ is given below). Thus h^n corresponds to an n-fold power of this matrix and leads to binomial coefficients:

$$h = \begin{bmatrix} 1/2 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 1/2 & 1/2 \end{bmatrix} \qquad h^n = \frac{1}{2^n}\left[\binom{n}{i-j}\right]_{i,j}.$$

From this observation, we see that, for any state $\mathbf{s}$,

$$|h^n(\mathbf{s})| = \frac{1}{2^n}\sum_{i=0}^{k}\binom{n}{\leq (k-i)} s_i,$$

where we define $\binom{a}{\leq b} := \sum_{0 \leq b' \leq b}\binom{a}{b'}$. This notation allows us to rewrite (10) as (11) and concisely define $\texttt{Binning}_\texttt{k}$.

[8] In general, there are multiple optimal splits.

Algorithm 2 $\texttt{Binning}_\texttt{k}$

Summarize the expert performance with $\mathbf{s}$, and current predictions with $\mathbf{r}$. For both $y \in \{0, 1\}$, compute

$$q_y = q^{\mathbf{s}^{\mathbf{r},y}} = \max\left\{ q : \frac{1}{2^q} \sum_{i=0}^{k} \binom{q}{\leq (k-i)} (\mathbf{s}^{\mathbf{r},y})_i \geq 1 \right\}. \tag{11}$$

Using the functions $clip(\cdot)$ defined in equation (4) and $\widetilde{\ell}_0(\cdot)$ defined in (6), output prediction

$$\hat{y} = clip\left(\frac{q_1 - q_0}{4} + \frac{\widetilde{\ell}_0(h^{q_1}(\mathbf{s}^{\mathbf{r},1})) - \widetilde{\ell}_0(h^{q_0}(\mathbf{s}^{\mathbf{r},0})) + 1}{2} \right) \tag{12}$$

The naive complexity of solving (11) at every round is $O(kq^{\mathbf{s}}) = O(k^2 + k \log |\mathbf{s}|)$, and for computing $\widetilde{\ell}_0 \circ h^q$ in (12) is $O(kq^{\mathbf{s}})$. The overall computation for one round requires therefore $O(k^2 + k \log |\mathbf{s}| + n)$, where n is the initial number of experts. Using bookkeeping one can reduce the per-round complexity to $O(n)$ by maintaining binomials $\binom{q}{k-i}$, binomial tails $\binom{q}{\leq k-i}$, and the function $\widetilde{\ell}_0 \circ h^q(\mathbf{b}_i)$ for an expert with i mistakes. These values can be updated in constant time using recurrences.

3.5 Binning and Binomial Weighting

In the introduction we discussed the algorithm Binomial Weighting which makes deterministic predictions ($\hat{y} \in \{0, 1\}$) when a mistake bound k is given. BW finds a bound, q^*, on the number of times the master errs, and considers a set of virtual experts of size $\sum_j \binom{q^*+1}{\leq k-m_j}$ where the sum is taken over all experts j, and m_j denotes the number of mistakes of expert j. In some sense, q^* is computed in hindsight: we make q^* big enough so that we produce "enough" virtual experts, i.e. so that we don't halve the set of virtual experts too many times. It is chosen as

$$q^* = \max\left\{ q : q \leq \log_2 \sum_j \binom{q}{\leq k - m_j} \right\}. \tag{13}$$

Recall that, if we summarize the past performance of our experts with a state vector $\mathbf{s}$, then s_i is the number of experts e that have made i mistakes and therefore $\sum_j \binom{q^*}{k-m_j} = \sum_{i=0}^{k} \binom{q^*}{k-i} s_i$. Interestingly, if we exponentiate the above equation and divide by 2^{q^*} we arrive at equation (11) and thus $q^{\mathbf{s}} = q^*$.

The loss bound on $\texttt{Binning}_\texttt{k}$ is $\frac{q^{\mathbf{s}}}{2} + \widetilde{\ell}_0(h^{q^{\mathbf{s}}}(\mathbf{s}))$. Notice, the binomial nature of $h^{q^{\mathbf{s}}}$ forces the majority of the weight in $\mathbf{s}$ to be collected in s_k, yet this term has coefficient $\frac{1}{2}$ in the function $\widetilde{\ell}_0$. However, the term $\widetilde{\ell}_0(h^{q^{\mathbf{s}}}(\mathbf{s}))$ quickly drops to a constant c independent of k as the number of experts goes to ∞. Thus, the loss $\widetilde{\ell}(\mathbf{s}) \leq \frac{q^*}{2} + c$ for large enough n. (The exact bound on $\widetilde{\ell}_0(h^{q^{\mathbf{s}}}(\mathbf{s}))$ requires some computation and is discussed in the full version of this paper.)

The factor of $\frac{1}{2}$ is to be expected: the deterministic algorithm BW suffers loss 1 at every round in the worst case, while $\mathtt{Binning}_k$ will be forced to predict $\hat{y} = \frac{1}{2}$ against an optimal adversary, thus suffering loss $\frac{1}{2}$.

4 Experiments

We ran experiments with several real-world datasets (see table 1) obtained from the UCI Machine Learning Repository. We chose rather simple experts: real valued features were replaced with an "above or below median" expert. Categorical features were replaced with a set of experts, one for each category value. When category value v was encountered, expert v would be set to 1 and all others would be set to 0. We also included the constant expert and for every expert, the complement expert. The original ordering of the datasets was preserved.

data	echo	bupa	hep	wpbc	dia	bcw	Aust	ion	beast	wdbc	kr-kp	cat	a-l
rounds	131	345	155	198	768	699	690	351	286	569	3196	873	8124
experts	28	14	70	72	18	20	30	70	100	64	220	1234	280
mistakes	39	145	32	47	237	65	184	122	79	83	1012	3	920

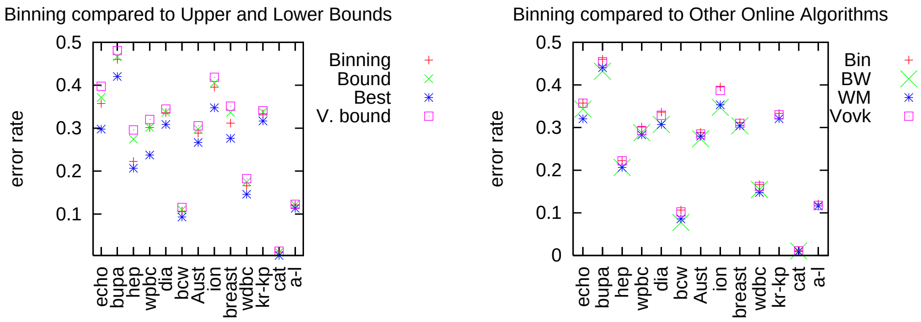

Fig. 1. In the table above we give the number of rounds, the number of experts, and the number of mistakes of the best expert. The datasets are ordered above by byte size. In the graph on the lower left we show how the performance of Binning compares to the upper Bound on Binning performance, the Best single expert, and the bound on Vovk's algorithm. The right graph shows how the performance of Binning compares to Binomial Weighting (BW) (2 entries missing), Weighted Majority (WM), and Vovk's algorithm.

All of the algorithms require the value k (the mistake bound) for tuning. Since we want to compare the algorithms when they are optimally tuned we precomputed this value and provided it to each algorithm. The results are graphed in Figure 1. The left graph shows that the Binning bound is tighter than the bound for Vovk's algorithm and often quite close to the actual Binning performance. The right graph is surprising: the performance appears to worsen with the tighter bound. In fact, BW and WM, both deterministic algorithms, performed better than the best expert on 2 datasets. Perhaps a tighter bound has an associated cost when, in actuality, we are not in an adversarial setting.

5 Conclusion

We discovered a new technique that replaces exponential weights by an optimal algorithm for a continuous game. The key idea is to allow partial experts. Our method uses binomial weights which are more refined then exponentially decaying weights. Note that the latter weights can be derived using a relative entropy as a divergence [KW97,KW99]. This reminds us of the application of entropies in statistical mechanics where the entropy is used to approximate exact counts that are based on binomial tails. In on-line learning we first discovered the entropy based algorithms which use approximate counting by approximating the binomial tails with exponentials. More recently refined algorithms are emerging that are based on binomial counting (See also [Fre95,FO02] for related work on Boosting).

The algorithms based on exponential weights are clearly simpler, but they also require knowledge of k for tuning the learning rate η well. The algorithms that use binomial weights always make use of k. If no tight upper bound of the true k is not known, then simple doubling tricks can be employed (see e.g. Section 4.6 of [CBFH$^+$97]).

Much work has been done in the realm of exponentially decaying weights: shifting experts [HW98], multi-arm bandit problems [ACBFS95] and so forth. In general, the question is whether other cases in which exponential weights have been used are amenable to our new technique of splitting experts. Also, in some settings [FS97] the algorithm needs to commit to a probability vector (w_i) over the experts at the beginning of each trial. It then receives a loss vector $(L_i) \in [0,1]^n$ and incurs a loss $\sum_i w_i L_i$. The question is whether weights can be extracted from our Binning algorithm and the optimal algorithm can be found for the modified setting when the experts are allowed to be continuous quantities.

For a more immediate goal note the following. We assumed that the predictions of the experts and the labels were binary. However in the realm of exponentially decaying weights, Vovk's aggregating algorithm for the absolute loss [Vov90,CBFH$^+$97] can handle expert's predictions in $[0,1]$ and the Vee algorithm of [HKW98] can in addition handle labels in $[0,1]$. We believe that with some additional effort our methods will generalize to handle these cases as well.

A Proof of Lemma 1

To prove the lemma it suffices to prove the following statements:

(a) For all splits $\mathbf{0} < \mathbf{r} < \mathbf{s}$ and $y \in \{0,1\}$, $\ell(\mathbf{s}) > \ell(\mathbf{s}^{\mathbf{r},y})$.
(b) For all $|\mathbf{s}| > 1$, $\ell(\mathbf{s}) > 1 + \ell(\mathbf{s}^+)$ and unanimous splits are not optimal.

Notice that (a) implies that $\ell(\mathbf{s}) > \max\{\ell(\mathbf{s}^{\mathbf{r},0}), \ell(\mathbf{s}^{\mathbf{r},1})\}$, which rules out line 2 of the recursion (2). Also, statement (b) rules out line 1 and together, they imply the lemma.

We prove the above two statements by induction on the mistake budget $B(\mathbf{s})$. Statements (a) and (b) trivially hold for the base case $\mathbf{s} = \mathbf{0}$, i.e. when $B(\mathbf{s}) = -1$. Assume that (a) and (b) holds for all states $\mathbf{s}'$ where $B(\mathbf{s}') < B(\mathbf{s})$. We now show that the statements holds for state $\mathbf{s}$.

For (a), let $\mathbf{r}$ be any non-unanimous split of $\mathbf{s}$ and let $\mathbf{z} := \mathbf{s}^{\mathbf{r},0}$. Consider two cases depending on the value of $|\mathbf{z}|$.

Assume $|\mathbf{z}| = 1$, then $\mathbf{z} = \mathbf{b}_i$. This implies that $\mathbf{s} = \mathbf{b}_i + m\mathbf{b}_k$ and $\mathbf{r} = m\mathbf{b}_k$ for some $m > 0$. Then $\mathbf{s}^{\mathbf{r},0} = \mathbf{b}_i$ and $\mathbf{s}^{\mathbf{r},1} = \mathbf{b}_{i+1} + m\mathbf{b}_k$, and by (2), we see that

$$\ell(\mathbf{s}) \geq \frac{\ell(\mathbf{b}_i) + \ell(\mathbf{b}_{i+1} + m\mathbf{b}_k) + 1}{2} = \frac{k - i + \ell(\mathbf{b}_{i+1} + m\mathbf{b}_k) + 1}{2}.$$

We now show that $\ell(\mathbf{b}_{i+1}+m\mathbf{b}_k) > k-i-1$ which implies that $\ell(\mathbf{s}) > k-i = \ell(\mathbf{z})$. If $i = k$ then $\mathbf{b}_{i+1} = \mathbf{0}$ and $\mathbf{b}_{i+1} + m\mathbf{b}_k = m\mathbf{b}_k$. Therefore, we can see by induction that $\ell(m\mathbf{b}_k) \geq \ell(\mathbf{b}_k) = 0 > -1$. Also if $i < k$ then by a similar induction, $\ell(\mathbf{b}_{i+1} + m\mathbf{b}_k) > \ell(\mathbf{b}_{i+1}) = k - i - 1$. This implies that $\ell(\mathbf{s}) > k - i$ as desired.

Assume $|\mathbf{z}| \geq 2$. Since $B(\mathbf{z}) < B(\mathbf{s})$, it follows by induction that there is some non-unanimous split $\mathbf{q}$ of $\mathbf{z}$ s.t.

$$\ell(\mathbf{z}) = \frac{\ell(\mathbf{z}^{\mathbf{q},1}) + \ell(\mathbf{z}^{\mathbf{q},0}) + 1}{2} = \frac{\ell((\mathbf{s}^{\mathbf{r},0})^{\mathbf{q},1}) + \ell((\mathbf{s}^{\mathbf{r},0})^{\mathbf{q},0}) + 1}{2}.$$

Since $\mathbf{r}$ is non-unanimous it follows by induction that the above is less than $\frac{\ell(\mathbf{s}^{\mathbf{q},1})+\ell(\mathbf{s}^{\mathbf{q},0})+1}{2} \leq \ell(\mathbf{s})$.

For the proof of (b), let $\mathbf{z} := \mathbf{s}^+$. We consider two cases depending on the value of $|\mathbf{z}|$.

Assume $|\mathbf{z}| = 1$. Then $\mathbf{z} = \mathbf{b}_{i+1}$ for some i, and thus $\mathbf{s} = \mathbf{b}_i + m\mathbf{b}_k$ for some $m \geq 1$. Notice, we already proved above that, when $\mathbf{s}$ is of this form, that $\ell(\mathbf{s}) > k - i = \ell(\mathbf{s}^+) + 1$ as desired.

Assume $|\mathbf{z}| \geq 2$. We prove the statement by induction on the mistake budget $B(\mathbf{s})$. For the base case $B(\mathbf{s}) = 1$, we must be in state $(0, \ldots, 0, 2) = 2\mathbf{b}_k$ and therefore $\mathbf{z} = \mathbf{0}$, so the statement is true. We now proceed to the induction step. Notice that $B(\mathbf{z}) < B(\mathbf{s})$ and $|\mathbf{z}| \geq 2$. By induction, we can therefore find a non-unanimous split $\mathbf{q}'$ of $\mathbf{z}$ where $\ell(\mathbf{z}) = \frac{\ell(\mathbf{z}^{\mathbf{q}',1})+\ell(\mathbf{z}^{\mathbf{q}',0})+1}{2}$. We now choose $\mathbf{q}$ so that $\mathbf{q}^+ = \mathbf{q}'$. Observe that $(\mathbf{s}^{\mathbf{q},y})^+ = \mathbf{z}^{\mathbf{q}',y}$ for $y = 0, 1$. Also, $B(\mathbf{s}^{\mathbf{q},y}) < B(\mathbf{s})$, and thus by induction we can apply (b), giving us that $\ell(\mathbf{s}^{\mathbf{q},y}) > \ell((\mathbf{s}^{\mathbf{q},y})^+)+1 = \ell(\mathbf{z}^{\mathbf{q}',y}) + 1$. Combining these statements, we see that

$$\ell(\mathbf{s}) \geq \frac{\ell(\mathbf{s}^{\mathbf{q},0}) + \ell(\mathbf{s}^{\mathbf{q},1}) + 1}{2} > \frac{\ell(\mathbf{z}^{\mathbf{q}',0}) + \ell(\mathbf{z}^{\mathbf{q}',1}) + 3}{2} = \ell(\mathbf{z}) + 1 = \ell(\mathbf{s}^+) + 1,$$

as desired. This completes the proof of the lemma

References

[ACBFS95] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. Gambling in a rigged casino: the adversarial multi-armed bandit problem. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, pages 322–331. IEEE Computer Society Press, Los Alamitos, CA, 1995.

[CBFH+97] N. Cesa-Bianchi, Y. Freund, D. Haussler, D. P. Helmbold, R. E. Schapire, and M. K. Warmuth. How to use expert advice. *Journal of the ACM*, 44(3):427–485, 1997.

[CBFHW96] Nicolo Cesa-Bianchi, Yoav Freund, David P. Helmbold, and Manfred K. Warmuth. On-line prediction and conversion strategies. *Machine Learning*, 25:71–110, 1996.

[FO02] Y. Freund and M. Opper. Drifting games and Brownian motion. *Journal of Computer and System Sciences*, 64:113–132, 2002.

[Fre95] Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285, September 1995.

[FS97] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, August 1997.

[HKW98] D. Haussler, J. Kivinen, and M. K. Warmuth. Sequential prediction of individual sequences under general loss functions. IEEE Transactions on Information Theory, 44(2):1906–1925, September 1998.

[HW98] M. Herbster and M. K. Warmuth. Tracking the best expert. *Journal of Machine Learning*, 32(2):151–178, August 1998.

[KW97] J. Kivinen and M. K. Warmuth. Additive versus exponentiated gradient updates for linear prediction. *Information and Computation*, 132(1):1–64, January 1997.

[KW99] J. Kivinen and M. K. Warmuth. Averaging expert predictions. In Paul Fischer and Hans Ulrich Simon, editors, *Computational Learning Theory: 4th European Conference (EuroCOLT '99)*, pages 153–167, Berlin, March 1999. Springer.

[LW94] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994. An early version appeared in FOCS 89.

[Vov90] V. Vovk. Aggregating strategies. In *Proc. 3rd Annu. Workshop on Comput. Learning Theory*, pages 371–383. Morgan Kaufmann, 1990.