Theoretical Computer Science 411 (2010) 2670–2683

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Learning with continuous experts using drifting games

Indraneel Mukherjee *, Robert E. Schapire

Department of Computer Science, Princeton University, 35 Olden Street, Princeton, NJ 08540, United States

ARTICLE INFO

Keywords:
On-line learning
Expert advice
Drifting game

ABSTRACT

We consider the problem of learning to predict as well as the best in a group of experts making continuous predictions. We assume the learning algorithm has prior knowledge of the maximum number of mistakes of the best expert. We propose a new master strategy that achieves the best known performance for on-line learning with continuous experts in the mistake bounded model. Our ideas are based on drifting games, a generalization of boosting and on-line learning algorithms. We prove new lower bounds based on the drifting games framework which, though not as tight as previous bounds, have simpler proofs and do not require an enormous number of experts. We also extend previous lower bounds to show that our upper bounds are exactly tight for sufficiently many experts. A surprising consequence of our work is that continuous experts are only as powerful as experts making binary or no prediction in each round.

1. Introduction

We consider the problem of learning to predict as well as the best in a group of experts. Our model consists of a series of rounds. In each round, experts make predictions in $[-1, +1]$. This can be interpreted as giving a binary prediction, for example, if it will rain or not, with a certain degree of *confidence*. In particular, this means that an expert can choose to *abstain* from giving any prediction at all, in which case it predicts 0. The problem is to design a master algorithm that combines the expert predictions in each round to give its own binary prediction in $\{-1, +1\}$. At the end of each round, nature (or an adversary) reveals the truth, which is a value in $\{-1, +1\}$. The experts and the master suffer loss that depends on the amount by which their predictions deviated from the truth. Our goal is to ensure that our master algorithm does not suffer much loss relative to the best expert.

An important feature of our model, which we define rigorously in Section 2, is that we assume the master algorithm has prior knowledge of a bound k on the total loss that the best expert will suffer. With binary experts outputting predictions in $\{-1, +1\}$, this problem was essentially solved entirely by Cesa-Bianchi et al. [3] who proposed the Binomial Weights (BW) algorithm. However, their work cannot be applied to our setting since here the experts are continuous, with predictions in $[-1, +1]$. In such a setting, other methods, notably exponential-weight algorithms [2,6,7], can be used instead. However, such algorithms do not enjoy the same level of tight optimality of the BW algorithm, and it has been an open problem since the introduction of BW as to whether this method can be generalized to continuous experts.

In this paper, we present just such a generalization. In Section 3, we propose a new master strategy that gives the best possible performance for this problem. Our algorithm predicts using a weighted majority of the experts' predictions in each round, where the weights are carefully chosen to ensure that the master's loss is small relative to k. We also show that our algorithm runs in polynomial time.

* Corresponding author.
E-mail addresses: imukherj@cs.princeton.edu (I. Mukherjee), schapire@cs.princeton.edu (R.E. Schapire).

0304-3975/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2010.04.004

Our algorithm is based on the *drifting games*' framework introduced by Schapire [9]. This framework generalizes a number of on-line and boosting learning algorithms, including boost-by-majority [4], AdaBoost [6], the weighted majority algorithm [7] and Binomial Weights [3]. We apply the drifting games' framework directly both to derive our algorithm and to analyze its performance which, as seen in Section 4, relies heavily on properties of drifting games.

We also provide in Section 5 new lower bound constructions for master algorithms which employ weighted-majority predictions. These are slightly weaker than those provided by Cesa-Bianchi et al. [3] which already show that our algorithm is nearly the best possible when the number of experts is very large. However, their techniques are based on Spencer's [10] sophisticated results for Ulam's game and require an enormous number of experts. In contrast, our lower bounds use simpler arguments based on the drifting games' framework and are meaningful for any number of experts. In the Appendix, we show how to extend the Spencer's [10] and Cesa-Bianchi et al.'s [3] approaches to achieve exactly tight lower bounds.

A consequence of our results is that learning in our framework with continuous experts is no harder than learning with abstaining experts, i.e., experts whose predictions are restricted to be in $\{-1, 0, +1\}$ (assuming that $2k$ is an integer), although there is a small gap between abstaining and binary experts.

Other related work. Abernethy et al. [1] extended the BW algorithm to the setting where the experts remain binary, but the master is allowed to predict continuously. Their results do not directly apply to our setting, but with some effort can be used to obtain sub-optimal bounds. For completeness, a sketch is included in Section 2.1.

Continuous-time versions of drifting games, with potential applications to on-line learning, were studied by Freund and Opper [5]. In their setting, learning rounds are no longer discrete, but are instead continuous.

2. Expert learning model

Our expert learning model can be viewed as the following game. The players of the game are a fixed set of m experts, a master algorithm, and an adversary. The game proceeds through T rounds. In each round t, the following procedure take place.

- The master chooses real weights $w_1^t, \ldots, w_m^t$ over the experts.
- Each expert i makes a prediction $x_i^t \in [-1, +1]$. The experts' predictions are controlled by the adversary; we distinguish between the experts and the adversary for clarity of exposition.
- The master predicts $\hat{y}^t \triangleq \text{sign}(\sum_i w_i^t x_i^t) \in \{-1, 0, +1\}$. The sign function maps positive reals to 1, negative reals to -1 and 0 to itself.
- The adversary then chooses a label $y^t \in \{-1, +1\}$, causing expert i to suffer loss $\frac{1}{2}|y^t - x_i^t|$, and the master to suffer loss $\mathbf{1}(y^t \neq \hat{y})$, where $\mathbf{1}$ is the indicator function. Note that predicting 0 counts as a mistake.

The total loss of any player is the sum of the losses in each round. It is guaranteed that some expert will suffer at most k total loss, where k is known ahead of time to the master. The goal of the master is to come up with a strategy to choose distributions $\mathbf{w}^t$ in each round, so as to minimize its loss against the worst possible adversary. The performance of every fixed strategy will thus be a function of m and k.

We will only consider *conservative* master algorithms, i.e., algorithms that *ignore* rounds where it does not make a mistake, so that the weights it chooses in a certain round depend only on past rounds where it made a mistake. This will also allow us to assume that as long as the game can continue, the master makes a mistake in every round. Since one can easily convert any master algorithm in a mistake bounded model like ours to a conservative one without loss of performance, we do not lose generality with this assumption.

2.1. The binning algorithm

Abernethy et al. [1] study a similar game, where the Master predicts continuously, but the expert predictions are binary. Experts in their model may split themselves into two parts of varying masses; each part independently makes a prediction and suffers integral loss. The game terminates when the total mass of experts with error at most k is less than 1. They develop an optimal algorithm for their model, called binning. Superficially, the two models appear to be the same, but in fact, critical differences exist.

The first main difference between the models assumed by binning and ours is that experts suffer integral losses in binning, whereas in our case experts suffer continuous losses. However, their experts are amorphous and can split themselves into different parts which make different predictions and suffer different integral losses. The game state in the binning model is captured by the total mass of experts that have suffered losses $0, \ldots, k$, respectively. Mathematically, their state-space is $[0, m]^{k+1}$. Their terminal states are those in which the sum of coordinates is less than 1. In contrast, the state of our expert learning game in any given round is given by the (possibly fractional) cumulative losses of each expert. Mathematically, our state space is $[0, k + 1]^m$; the terminal states are the ones where each coordinate is greater than k.

The next main difference is that whereas we find the optimal *deterministic* Master, Abernethy et al. [1] bound the *expected* error of a Master that predicts *randomly*. The very point of their work was to compute the improvement that a random Master could achieve over the best possible deterministic master, which, in their setting, was known to be the Binomial Weights algorithm of Cesa-Bianchi et al. [3]. Hence, their techniques are suited for finding optimal random Master algorithms, which is not the question we study here.

3. Choosing weights

We describe a strategy of the master for choosing a distribution on the experts in each round. Computing this strategy requires playing a different type of game called a *drifting game* introduced by Schapire [9]. We begin with an abstract definition of the game, and then go on to show how such games can be used to derive the BW algorithm [3], which is the optimal (in the broadest possible sense) master strategy in the case of binary experts. We then show how similar ideas can be used to derive a master algorithm for continuous experts.

3.1. Drifting games

A drifting game is played by a *shepherd* and *m sheep* (also known as *chips* in [9]). In general, sheep exist in $\mathbb{R}^d$; however, in this paper, we only consider drifting games in which $d = 1$. The game proceeds through T rounds. In each round t, the following procedure takes place.

- The shepherd chooses a weight $w_i^t \in \mathbb{R}$ for each sheep i. The sign indicates the direction he intends the sheep to move, and the magnitude encodes the importance he places on that sheep.
- Sheep i responds by shifting by z_i^t, where z_i^t belongs to a fixed set of directions B. Additionally, the following drifting constraint is obeyed

$$\sum_{i=1}^{m} w_i^t z_i^t \geq \delta \sum_{i=1}^{m} |w_i|. \tag{1}$$

Here $\delta \geq 0$ and $B \subseteq \mathbb{R}$ are parameters of the game.

The shepherd suffers a loss $L(s)$ for every sheep that is at location $s \in \mathbb{R}$ at the end of the game; here $L : \mathbb{R} \to \mathbb{R}$ is a real function on the space. Initially, all the sheep are at the origin, so at the end of the game, sheep i is at $\sum_{t=1}^{T} z_i^t$. The goal of the shepherd is to choose weights in a way that would minimize its average loss $\frac{1}{m}\sum_i L(\sum_t z_i^t)$, assuming the worst behavior from the sheep.

Schapire [9] suggests a shepherd strategy OS based on a set of potential functions $\phi_t : \mathbb{R}^d \to \mathbb{R}$ defined recursively as follows:

- $\phi_T(x) = L(x)$
- $\phi_{t-1}(x) = \min_{w\in\mathbb{R}} \max_{z\in B}(\phi_t(x+z) + wz - \delta|w|)$.

Denoting by s_i^t the position of sheep i at time t, the OS algorithm chooses w_i^t as follows:

$$w_i^t \in \arg\min_{w\in\mathbb{R}} \max_{z\in B}(\phi_{t+1}(s_i^t + z) + wz - \delta|w|). \tag{2}$$

(In this paper, we regard arg min or arg max as returning the set of all values realizing the minimum or maximum.) Schapire [9] provides an upper bound on the performance of the OS algorithm and argues that (under some natural assumptions) it is optimal when the number of sheep m is very large. We record the results in the theorem below.

Theorem 1 (*Drifting games [9]*). *If B is a bounded subset of $\mathbb{R}$, and L is locally bounded, then the loss suffered by the OS algorithm is upper bounded by $\phi_0(0)$, where ϕ is defined as above. Additionally, if B contains both positive and negative numbers of magnitude greater than δ, and L is globally bounded, then, given any $\epsilon > 0$, for sufficiently large m, the sheep can force any shepherd algorithm to suffer at least $\phi_0(0) - \epsilon$ loss at the end of the game.*

3.2. Learning with binary experts using drifting games

Consider our expert learning model with the change that experts make $\{-1, +1\}$ instead of continuous predictions. The binomial weights algorithm [3] is the best possible master strategy for this problem, even among master algorithms not restricted to predicting a weighted majority of the experts' predictions at each stage. We show how a master can simulate a drifting game to derive a strategy for choosing weights on the experts so as to perform as well as the BW algorithm.

The drifting game parameters are $B = \{-1, +1\}$ and $\delta = 0$, and the loss function is $L(s) = \mathbf{1}(x \leq 2k - T)$. The number of rounds is $T = T_0 + 1$ where T_0 will be specified later. For every expert, there is a sheep. At the beginning of a round, the master uses the shepherd's choice $w_1, \ldots, w_m$ for that round to assign weights to experts. After seeing the expert predictions x_i and the label y produced by the adversary, the master causes sheep i to drift by $z_i = -yx_i$. The drifting constraint (1) holds since we are in the conservative setting and assume that a mistake is made by the master in each round. Schapire [9] shows that the resulting algorithm is equivalent to BW, for a certain choice of T_0.

We use the notation $\binom{q}{\leq k}$ to denote $\sum_{i=0}^{k} \binom{q}{i}$.

Theorem 2 (*Learning with binary experts [3,9]*). *Consider the expert learning model described in Section 2, with the change that the expert predictions lie in* $\{-1, +1\}$. *For this problem, when* T_0 *is set to be*

$$\max\left\{q \in \mathbb{N} : q \leq \lg m + \lg \binom{q}{\leq k}\right\}, \tag{3}$$

the number of mistakes made by the master algorithm described in this section is upper bounded by T_0. *Further, the resulting algorithm can be computed efficiently.*

Proof. At the heart of the proof, and the reason behind our choice of T_0, lies the following result which was proved by Schapire [9]: if a drifting game with parameters $\delta = 0$, $B = \{-1, +1\}$ and loss function $L(x) = \mathbf{1}(x \leq 2k - T)$ is played for T rounds, then ϕ_t can be computed exactly, yielding

$$\phi_0(0) = 2^{-T} \binom{T}{\leq k}. \tag{4}$$

Note that the position of a sheep after t rounds is $2M - t$, where M is the loss suffered by the corresponding expert until then. Since we are guaranteed a mistake bound of at most k on some expert, we always have $\sum_i L(s_i^t) \geq 1$, where s_i^t is sheep i's position after t rounds of play. If the game could continue for $T = 1 + T_0$ rounds, using Theorem 1 and (4) we would have the following contradiction:

$$\frac{1}{m} \leq \frac{1}{m} \sum_i L(s_i^T) \leq \phi_0(0) = 2^{-T} \binom{T}{\leq k} < \frac{1}{m}.$$

The last inequality follows from the fact that T_0 was chosen to satisfy $T_0 = \max\{\text{number of rounds} : \phi_0(\mathbf{0}) \geq \frac{1}{m}\}$. This upper bounds the maximum number of rounds for which the game can continue, or equivalently, the maximum number of mistakes our master algorithm makes, by T_0. □

3.3. Drifting games for continuous experts

The same approach from the previous section can be applied to our expert learning model, where experts make $[-1, +1]$ predictions. The drifting game parameter B changes to $B = [-1, +1]$, and a new expression for T_0 has to be chosen; everything else remains the same. We summarize the master strategy in Algorithm 1, where we choose $T_0 = \max\left\{q \in \mathbb{N} : q \leq \lg m + \lg \binom{T+1}{\leq k}\right\}$. We can now state our first main result.

Algorithm 1 Master algorithm for continuous experts

Require: k—mistake bound, m—number of experts

$T_0 \leftarrow \max\left\{q \in \mathbb{N} : q \leq \lg m + \lg \binom{T+1}{\leq k}\right\}$
$B \leftarrow [-1, +1]$, $L \leftarrow \mathbf{1}(x \leq 2k - T)$
$\delta \leftarrow 0$
Setup drifting game with $T = 1 + T_0$, B, δ, L, and shepherd OS

{**Note**: Game cannot continue beyond T_0 rounds}
for $t = 1$ to T_0 **do**
  Accept $w_1^t, \ldots, w_m^t$ from shepherd (Algorithm 2)
  Accept predictions $x_1^t, \ldots, x_m^t$ from experts.
  Predict $\hat{y}^t = \text{sign}(\sum_i w_i x_i^t)$
  Accept label y^t from adversary.
  For each i, make sheep i drift by $z_i^t \triangleq -y^t x_i^t$
end for

Theorem 3 (*Learning with continuous experts*). *Consider the expert learning model described in Section 2. For that problem, the loss of the master algorithm described in Algorithm 1 is upper bounded by* T_0, *which is equal to*

$$\max\left\{q \in \mathbb{N} : q \leq \lg m + \lg \binom{q+1}{\leq k}\right\}. \tag{5}$$

The bound for continuous experts looks very similar to the one for binary experts, except that $\lg\binom{q}{\le k}$ in (3) is replaced by $\lg\binom{q+1}{\le k}$ in (5). Roughly, this means the continuous bound is twice as much as the binary one.

As in learning with binary experts, the choice of T_0 in Theorem 3 is dictated by the analysis of the drifting game used for playing with continuous experts. This analysis also constitutes our main technical contribution and is summarized in the next theorem, but we defer a proof until the next section.

Theorem 4 (*Drifting games for* $[-1, +1]$ *experts*)**.** *Consider the drifting game with parameters* $\delta = 0$, $B = [-1, +1]$, *total number of rounds* T *and loss function* $L(x) = \mathbf{1}(x \le 2k - T)$. *The value of the potential function for this game at any integer point* $s + 2k - T$ *is given by*

$$\phi_{T-t}(s + 2k - T) = \begin{cases} 1 & \text{if } s \le 0 \\ 1 - 2^{-t}\sum_{i=0}^{s-1}\binom{t}{\lceil\frac{t+i}{2}\rceil} & \text{else.} \end{cases} \tag{6}$$

In particular, we have

$$\phi_0(0) = 2^{-T}\binom{T+1}{\le k}.$$

Further, the OS strategy for this game can be computed efficiently.

Proof of Theorem 3. Observe that $T_0 = \max\{\text{number of rounds} : \phi_0(0) \ge \frac{1}{m}\}$, where ϕ_0 is the potential associated with the drifting game in Theorem 4. The rest of the proof is the same as that for Theorem 2. □

We can loosely upper bound the expression for the number of mistakes in Theorem 3 by

$$2k + \ln m\left(1 + \sqrt{1 + \frac{4k}{\ln m}}\right) - 1.$$

In Section 5 we will prove that, when the number of experts m is around 2^k, the mistake guarantee given in Theorem 3 is the best possible, up to an additive $O(\log k)$ term, when considering master algorithms that predict a weighted majority of the experts' predictions in each round. When $m \ge 2^{2^k}$, our upper bound is exactly tight as shown in the Appendix.

4. Analysis of drifting games for continuous experts

In this section we analyze the continuous drifting game and prove Theorem 4. Throughout we will be using the following two facts: ϕ_t is decreasing and takes values in $[0, 1]$. These facts were proved more generally by Schapire [9].

We begin with a technical result necessary for proving Theorem 4.

Theorem 5 (*Piecewise convexity*)**.** *For every round* t, ϕ_t *is piecewise convex with pieces breaking at integers, i.e., for every integer* n, ϕ_t *is convex in* $[n, n + 1]$.

The proof of this theorem is complicated and we defer it to Section 6. The proof relies on Lemma 1, which will also be useful otherwise. The lemma gives us a tool for recursively computing the potentials ϕ_t. It can be proved using a more general result in [9], but here we give a direct proof for the case of interest.

Lemma 1. *If* ϕ_t *is piecewise convex with pieces breaking at integers, then for* $s \notin \mathbb{Z}$,

$$\phi_{t-1}(s) = \max\left\{\frac{z\phi_t(s+z') - z'\phi_t(s+z)}{z - z'} : z, z' \in Z, zz' < 0\right\} \tag{7}$$

where

$$Z = \{z \in [-1, +1] : s + z \in \mathbb{Z}\} \cup \{-1, +1\}. \tag{8}$$

For s *integral,* $\phi_{t-1}(s)$ *is the maximum of* $\phi_t(s)$ *and the above expression.*

Proof. By definition

$$\phi_{t-1}(s) = \min_w \max_{z\in[-1,+1]} (\phi_t(s + z) + wz).$$

For fixed s and w, our assumptions imply that $\phi_t(s + z) + wz$ is piecewise convex in z. As z varies over the convex set $[-1, +1]$, the maximum will be realized either at an endpoint, -1 or 1, or when $s + z$ lies at one of the endpoints of the convex pieces, which happens at the integers. This shows that we can restrict z to Z while evaluating $\phi_{t-1}(s)$.

Denote by Δ the simplex of distributions over Z. By the discussion above,

$$\begin{aligned}\phi_{t-1}(s) &= \min_w \max_{z\in Z}(\phi_t(s+z)+wz)\\ &= \min_w \max_{p\in\Delta} \mathsf{E}_{z\sim p}\left[\phi_t(s+z)+wz\right]\\ &= \max_{p\in\Delta} \min_w \mathsf{E}_{z\sim p}\left[\phi_t(s+z)+wz\right]\end{aligned}$$

where the last equality comes from Corollary 37.3.2 of [8]. Interpreting the right side as the Lagrangian dual we may compute $\phi_{t-1}(s)$ as the solution to the following optimization problem

$$\begin{aligned}&\max_{p\in\Delta} \ \mathsf{E}_{z\sim p}\left[\phi_t(s+z)\right]\\ &s.t.\quad \mathsf{E}_{z\sim p}[z]=0.\end{aligned}$$

The above is a linear program and is hence optimized at vertices of the polytope $\{p\in\Delta : \mathsf{E}_{z\sim p}[z]=0\}$, which are mean-zero distributions supported on two points z, z' of opposite signs, or concentrated on 0 when feasible, i.e., when $s\in Z$. Maximizing $\mathsf{E}_{z\sim p}\left[\phi_t(s+z)\right]$ over such vertices p yields the lemma. □

As a corollary, we show how the OS algorithm may use the potentials to compute weights according to (2).

Corollary 1. *In round $t-1$, the OS algorithm puts the following weight on a sheep at location s:*

$$w^* \triangleq -\left(\frac{\phi_t(s+z_2)-\phi_t(s+z_1)}{z_2-z_1}\right), \tag{9}$$

where $z_1<0<z_2$ realize the maximum in the right-hand side of (7).

Proof. We need to show

$$\forall z\in[-1,+1]:\phi_t(s+z)+w^*z\le\phi_{t-1}(s).$$

As in the proof of Lemma 1, the maximum of the left-hand side is attained by some z lying in Z (defined in (8)). Positive and negative z need to be handled separately but are symmetric (the case $z=0\in Z$ occurs only when s is integral; but for such s, Lemma 1 tells us $\phi_{t-1}(s)\ge\phi_t(s)=\phi_t(s+0)+w^*\cdot 0$ anyway). Therefore, it suffices to show

$$\text{if } z\in Z, z>0,\ \text{then } \phi_t(s+z)+w^*z\le\phi_{t-1}(s).$$

Call the expression being maximized in (7) $f(z',z)$ and rewrite it as

$$f(z',z)=\phi_t(s+z')-s(z',z)z'.$$

where $s(z',z)$ denotes the slope $\left(\phi_t(s+z)-\phi_t(s+z')\right)/\left(z-z'\right)$. Note both s, f are symmetric functions of their arguments. Since

$$z_1<0<z_2\in\arg\max_{z,z'\in Z:z'<0<z}\phi_t(s+z')-s(z',z)z',$$

we may conclude that

$$w^*=-\max_{0<z\in Z}s(z_1,z). \tag{10}$$

Finish by observing

$$\begin{aligned}\phi_{t-1}(s) &\ge \max_{0<z\in Z} f(z,z_1)\ \text{(Lemma 1)}\\ &= \max_{0<z\in Z}\phi_t(s+z)-s(z_1,z)z\\ &\ge \max_{0<z\in Z}\phi_t(s+z)+w^*z \quad \text{(by (10))}. \quad \square\end{aligned}$$

It is now straightforward to prove Theorem 4.

Proof of Theorem 4. Theorem 5 and Lemma 1 imply that for integer points s

$$\phi_{t-1}(s)=\max\left\{\phi_t(s),\frac{\phi_t(s-1)+\phi_t(s+1)}{2}\right\}. \tag{11}$$

One can finish the proof by directly substituting into (11) the expression for $\phi_t(s)$ given in (6) and verifying that the inequality holds. We omit calculations.

Corollary 1 shows how the shepherd computes weights in each round. The OS routine is summarized in Algorithm 2. All we now need is a way to efficiently compute the potentials using recurrences (7) and (11). Note the value of ϕ_t at point s depends upon values at $\mathbb{Z}$ and $s+\mathbb{Z}$. An easy induction yields for all t, $\phi_t(s)=1$ for $s\le 2k-T$ and $\phi_t(s)=0$ for $s\ge T$. Standard dynamic programming techniques can now be used to compute $\phi_t(s)$ in time polynomial in T. □

Algorithm 2 OS subroutine: initialized with T, B, δ, L from Algorithm 1

Require: Current round t and positions $s_1^t, \dots, s_m^t$ of each sheep : $s_i^t = \sum_{t'<t} z_i^t$ (see Algorithm 1 for a definition of z_i^t)

for each sheep i **do**
  $Z \leftarrow \left\{-1, \lfloor s_i^t \rfloor - s_i^t, \lceil s_i^t \rceil - s_i^t, +1\right\}$ as in (8)
  Pick $z, z' \in Z$ maximizing, as in (7),
  $$\frac{z\phi_{T-(t-1)}(s+z') - z'\phi_{T-(t-1)}(s+z)}{z-z'}$$
  Set $w_i^t \leftarrow \frac{\phi_{T-(t-1)}(s+z')-\phi_{T-(t-1)}(s+z)}{z-z'}$ as in (9)
end for
return $w_1^t, \dots, w_m^t$

Notice that (11) is the same as what we would get if the sheep were allowed to drift only by $-1, 0, +1$ at each time step. Correspondingly, in terms of provable upper bounds, our algorithm performs no worse with continuous experts than it does with abstaining experts, while with binary experts, the upper bound on performance is a tiny bit better. Combined with our lower bound results, this implies that, for sufficiently many experts, abstaining experts are exactly as powerful as continuous ones, while binary experts are only slightly less powerful.

5. Lower bounds

In this section we provide lower bounds for on-line learning with continuous experts which almost match the upper bounds of Theorem 3, thus showing that the drifting game based on Algorithm 1 is near optimal. The bounds we obtain are weaker than those provided in the Appendix (which exactly match the upper bounds derived in the previous section). We nevertheless include these, since the arguments are much simpler, and do not require the number of experts to depend on the mistake bound. The main result of this section is the following theorem.

Theorem 6 (*Lower bound for expert learning*)**.** *Consider the expert learning model defined in Section 2. For every master algorithm, the adversary can choose labels and cause the experts to make predictions in each round in a manner so as to force the master algorithm to suffer a loss of*

$$\max\left\{q \in \mathbb{N} : q < \lg\left(\frac{m}{\sqrt{k}}\right) + \lg\binom{q+1}{\leq k} + \Theta(1)\right\}, \tag{12}$$

where $\Theta(1)$ is a quantity bounded between some absolute constants c_1 and c_2.

The loss bound given above and the upper bound in (5) define the smallest integer T such that $2^{-T}\binom{T+1}{\leq k}$ is less than $O(\frac{\sqrt{k}}{m})$ and $\frac{1}{m}$, respectively. Since $O(\log m)$ rounds will always be necessary, and $2^{-T}\binom{T+1}{\leq k}$ decreases exponentially fast when $T > 3k$, we see that the gap between the upper and lower bounds is only $O(\log k)$ when the number of experts m is around 2^k. When the number of experts is much larger ($m > 2^{2^k}$), a very different and highly involved analysis included in the Appendix shows that there is essentially no gap between the upper and lower bounds.

The proof of Theorem 6 consists of showing how an adversary in the expert model can exploit adversarial sheep movement in the drifting game to force any master algorithm to suffer high loss. This is the converse of what we saw in Section 3.3, where a well-performing shepherd algorithm gave rise to master algorithms suffering low loss. At the heart of the proof of Theorem 6 is the following result on drifting games, showing how the sheep may drift without violating the drifting constraint, and yet cause any shepherd a large amount of loss.

Theorem 7. *Consider the drifting game with parameters $\delta = 0$, $B = [-1, +1]$, number of rounds T and loss function $\mathbf{1}(x \leq 2k - T)$. For any shepherd algorithm, there exists a strategy for the sheep that causes the shepherd to suffer a loss of*

$$\phi_0(0) - \frac{\Theta(\sqrt{k})}{m}$$

at the end of the game.

We prove Theorem 7 in the next section, but first we show how it can be used to prove Theorem 6.

Proof of Theorem 6. The adversary in our expert model (defined in Section 2) simulates a drifting game in $\mathbb{R}$, with parameters as above. The drifting game is played for T rounds, where T is given by the expression (12). For every expert, there is a sheep. At the beginning of a round, if the master places weights $w_1, \dots, w_m$ on the experts, the adversary causes the shepherd to drive each sheep i in the direction w_i. If the sheep drift in the direction $z_1, \dots, z_m$, the adversary causes expert i to predict $x_i = z_i$ (remember the adversary controls expert predictions). The drifting constraint $\sum_i w_i z_i \geq \delta = 0$ ensures that the weighted majority prediction of the master is 0 or 1. The adversary then outputs the label $y = -1$, causing the master to make a mistake in each round.

Note that the position of a sheep after t rounds is $2M - t$ where M is the loss suffered by the corresponding expert until then; thus an expert has suffered at most k loss if and only if the corresponding sheep lies at a point less than $2k - T$ at the end of the game. Hence, by our choice of loss function, the mistake bound on the experts is equivalent to ensuring the constraint that the loss suffered by the shepherd algorithm is strictly positive at the end of the game, so that at least one sheep has a final loss of 1.

Theorem 7 guarantees that the sheep can drift in a way so that the shepherd suffers at least $\phi_0(0) - \Theta(\sqrt{k})/m$ loss, where we know from Theorem 4 that $\phi_0(0) = \binom{T+1}{\leq k}$. Our choice of T satisfies $\phi_0(0) - \Theta(\sqrt{k})/m > 0$, completing the proof. □

5.1. Lower bound for drifting game

We prove Theorem 7. Schapire [9] provides a similar though slightly weaker lower bound ($\phi_0(0) - O(T/\sqrt{m})$ instead of $\phi_0(0) - (\sqrt{k}/m)$) which leads to considerably weaker expert learning lower bounds. The reason is that Schapire's arguments hold for much more general drifting games. By carefully tailoring his proof to our specific learning model, we achieve significant improvements.

Proof of Theorem 7. We will show that on round t, the sheep can choose to drift in directions z_i so that

$$\frac{1}{m}\sum_i \phi_{t+1}(s_i^{t+1}) \geq \frac{1}{m}\sum_i \phi_t(s_i^t) - \frac{U_t}{m}. \tag{13}$$

Here s_i^t is the position of sheep i in round t, and

$$U_t \triangleq \max_{s^t} \frac{\phi_{t+1}(s^t - 1) - \phi_{t+1}(s^t + 1)}{2} \tag{14}$$

where the maximum is taken over all possible *integral* positions s^t of any sheep in round t. Note that this is different from the set of all possible positions, since the movement of the sheep is restricted to change by at most $+1$ or -1 in each round. Among the possible positions, we take supremum over only those positions which happen to lie at an integer.

Repeatedly applying the above yields

$$\frac{1}{m}\sum_i L(s_i^T) \geq \phi_0(0) - \frac{1}{m}\sum_t U_t.$$

Appealing to Lemma 4 will then produce the desired bound.

For each i, $s_i^0 = 0$. Our sheep strategy will choose every drift to be in $\{-1, 0, 1\}$. Hence we may assume $s_i^t \in \mathbb{Z}$ for each i, t.

Fix a round t. From Lemma 1 we have

$$\phi_t(s) = \max\left\{\phi_{t+1}(s), \frac{\phi_{t+1}(s-1) + \phi_{t+1}(s+1)}{2}\right\}.$$

Let $I = \{1, \ldots, m\}$, $I_0 = \{i : \phi_t(s_i^t) = \phi_{t+1}(s_i^t)\}$, $I_1 = I \setminus I_0$. For each $i \in I_0$ we set $z_i^t = 0$. This ensures

$$\sum_{i \in I_0} \phi_{t+1}(s_i^{t+1}) = \sum_{i \in I_0} \phi_t(s_i^t).$$

For each $i \in I_1$ we must have

$$\phi_t(s_i^t) = \frac{\phi_{t+1}(s_i^t - 1) + \phi_{t+1}(s_i^t + 1)}{2}.$$

For such i we will choose z_i^t in $\{-1, +1\}$. Define $a_i^t \triangleq \frac{\phi_{t+1}(s_i^t-1) - \phi_{t+1}(s_i^t+1)}{2}$. Then, for each $i \in I_1$,

$$\phi_{t+1}(s_i^{t+1}) = \phi_t(s_i^t) - z_i^t a_i^t$$

since $s_i^{t+1} = s_i^t + z_i^t$. Thus

$$\sum_{i \in I_1} \phi_{t+1}(s_i^{t+1}) = \sum_{i \in I_1} \phi_t(s_i^t) - \sum_{i \in I_1} z_i^t a_i^t.$$

Note that $a_i^t \in [0, U_t]$ by definition. If the shepherd weights for this round are $w_1^t, \ldots, w_m^t$, it suffices to ensure that $\sum_{i \in I_1} w_i^t z_i^t \geq 0$ while keeping $\sum_{i \in I_1} a_i^t z_i^t$ below U_t.

By Lemma 2, there exists a subset $P \subseteq I_1$ such that

$$\left| \sum_{i \in P} a_i^t - \sum_{j \in I_1 \setminus P} a_j^t \right| \leq U_t.$$

Assume without loss of generality that $\sum_{i \in P} w_i^t - \sum_{i \in I_1 \setminus P} w_j^t \geq 0$. Then, assigning $z_i^t = +1$ for $i \in P$ and $z_i^t = -1$ for $i \in I_1 \setminus P$ would ensure the drifting constraints as well as (13), completing our proof. □

Lemma 2. *For any sequence $a_1, \ldots, a_n$ of numbers in $[0, U]$*

$$\min_{P\subseteq I}\left|\sum_{i\in P} a_i - \sum_{j\in I\setminus P} a_j\right| \le U$$

where $I = \{1, \ldots, n\}$.

Proof. Define discrepancy to be the argument of the min, and let P^* realize the minimum. If P^*'s discrepancy were greater than U, we could transfer any a_i from the heavier group to get a partition with lower discrepancy, a contradiction. □

Notice that the U_t can be trivially bounded by 1, since the ϕ_t take values in $[0, 1]$. That would give us a lower bound of $\phi(0) - \frac{T}{m}$. By being more careful, we get the following improvement.

Lemma 3. *Define U_t as in (14). Then,*

$$U_{T-t} = \begin{cases} 2^{-t}\binom{t}{k} & \text{if } t > 2k \\ 2^{-t}\binom{t}{\lceil \frac{t}{2}\rceil} & \text{if } t \le 2k. \end{cases}$$

Proof. Using (6), we have, for $s \ge 0$

$$\frac{1}{2}[\phi_{T-t}(s-1+2k-T) - \phi_{T-t}(s+1+2k-T)] = 2^{-t-1}\left(\binom{t}{\lceil\frac{t+s-1}{2}\rceil} + \binom{t}{\lceil\frac{t+s}{2}\rceil}\right). \tag{15}$$

Let $s + 2k - T$ be the position of a sheep at the end of $T - t$ rounds. Since it can drift by at most -1 in the negative direction in any round, we have $s + 2k - T \ge t - T$ so that $s \ge t - 2k$. We take two cases, depending on the value of k.

Suppose $t > 2k$. Then, $s \ge t - 2k \ge 1$. Since (15) is larger for smaller (and non-negative) s, we can plug in $s = t - 2k$ to compute U_{T-t}:

$$U_{T-t} = 2^{-t-1}\left(\binom{t}{\lceil\frac{2t-2k-1}{2}\rceil} + \binom{t}{\lceil\frac{2t-2k}{2}\rceil}\right) = 2^{-t}\binom{t}{t-k} = 2^{-t}\binom{t}{k}.$$

When $t \le 2k$, s can be less than 1. If $s < 0$, the left-hand side of (15) is zero. If $s = 0$, the right-hand side of the same equation equals $2^{-t}\binom{t}{\lceil\frac{t}{2}\rceil}$. Hence, for $t \le 2k$, $U_{T-t} = 2^{-t}\binom{t}{\lceil\frac{t}{2}\rceil}$. □

Lemma 4. *Define U_t as in (14). Then, $\sum_t U_t \le \Theta(\sqrt{k})$.*

Proof. Lemma 3 yields

$$\sum_t U_t = \sum_{t>2k} U_t + \sum_{t\le 2k} U_t = \sum_{t>2k} 2^{-t}\binom{t}{k} + \sum_{t\le 2k} 2^{-t}\binom{t}{\lceil\frac{t}{2}\rceil}.$$

The terms in the first summation decrease by at least a factor of 3/4 successively, so that we can upper bound it by $4\binom{2k}{k}$. Stirling's approximation yields $\binom{t}{\lceil\frac{t}{2}\rceil} < \frac{O(1)}{\sqrt{t}}$ for all positive integers t. Hence, we have

$$\sum_t U_t \le \frac{4}{\sqrt{2k}} + \sum_{t\le 2k}\frac{O(1)}{\sqrt{t}} = \Theta(\sqrt{k})$$

completing our proof. □

6. Proof of Theorem 5

In this section we prove Theorem 5, which states that the potentials arising out of drifting games against continuous experts are not too complicated, but are in fact piecewise convex. Without this result, none of the computations would be possible. This is also the main reason lying behind the surprising fact that continuous experts are no stronger than abstaining ones, given a large number of experts.

We prove the Theorem by (backward) induction on time steps. Our proofs would be far simpler if we could inductively assume convex, rather than piecewise-convex, potentials at later time steps. Unfortunately, this is not the case, e.g. the end-potential is the non-convex 0–1 loss function. The considerably weaker piecewise-convex condition causes many technical complications, and our proof is rather long and messy. The main ingredient is Lemma 5, proved below. We first show how this lemma suffices to prove the theorem.

Proof of Theorem 5. Lemma 5 shows that ϕ_t is convex in $(n, n+1)$. Since ϕ_t is decreasing, it is right convex at n. We are left to show that ϕ_t is left convex at $n+1$; since ϕ_t is convex and decreasing in $(n, n+1)$, this is equivalent to showing that ϕ_t is left continuous at $n+1$. Inductively, ϕ_{t+1} is decreasing and convex in $[n, n+1]$, and hence necessarily left continuous at $n+1$. Since $\phi_t(n+1)$ is decreasing, it suffices to show that

$$\phi_t(n+1) \geq \lim_{s \to (n+1)^-} \phi_t(s).$$

By (7), for $s \in (n, n+1)$, $\phi_t(s)$ is either

$$\frac{\phi_{t+1}(s - z_1) + z_1 \phi_{t+1}(s+1)}{1 + z_1},$$

where $z_1 \in \{1, s-n\}$, or

$$\frac{(n+1-s)\phi_{t+1}(s - z_1) + z_1 \phi_{t+1}(n+1)}{n+1-s+z_1}.$$

As $s \to (n+1)^-$, the first expression tends to

$$\frac{\lim_{s_1 \to n^-} \phi_{t+1}(s_1) + \lim_{s_2 \to (n+2)^-} \phi_{t+1}(s_2)}{2} = \frac{\phi_{t+1}(n) + \phi_{t+1}(n+2)}{2},$$

where the equality holds since by induction, ϕ_{t+1} is left continuous at integers. Similarly, the second expression tends to $\phi_{t+1}(n+1)$. Therefore,

$$\lim_{s \to (n+1)^-} \phi_t(s) \leq \max\left\{\phi_{t+1}(n+1), \frac{\phi_{t+1}(n) + \phi_{t+1}(n+2)}{2}\right\}.$$

But the right-hand side of the above equation is $\phi_t(n+1)$ by (11). This completes the proof. □

Lemma 5. *For every integer n and round t, ϕ_t is convex in $(n, n+1)$.*

Proof. By backwards induction on t. The base case holds since ϕ_T is the loss function $\mathbf{1}(x \leq 2k - T)$. Assume that ϕ_{t+1} is piecewise convex. Fix any integer $n \in \mathbb{Z}$. We have to show that ϕ_t is convex in $(n, n+1)$. Recall that, for non-integral points s, Eqs (7) and (8) state that $\phi_t = \max\{H_{13}, H_{23}, H_{14}, H_{24}\}$ where

$$H_{ij}(s) = \frac{z_i \phi_{t+1}(s + z_j) - z_j \phi_{t+1}(s + z_i)}{z_i - z_j},$$

and $(z_1, z_2, z_3, z_4) = (-1, n-s, n+1-s, +1)$. Checking that H_{14} and H_{23} are convex is straightforward. It turns out that H_{13} and H_{24} need not be convex. However, below we show that $\max\{H_{23}, H_{24}\}$ is convex, and a very similar proof works for showing that $\max\{H_{23}, H_{13}\}$ is convex. As the supremum of convex functions is convex (Theorem 5.5 [8]), and because ϕ_t can be written as $\phi_t = \max\{\max\{H_{23}, H_{13}\}, \max\{H_{23}, H_{24}\}, H_{14}\}$, we are done.

We begin by making our task of showing that $\max\{H_{23}, H_{13}\}$ is convex a little easier. The next lemma shows that it suffices to show only local convexity, meaning that every point in the domain has a neighborhood over which the function is convex. The proofs of this and other technical lemmas are given later.

Lemma 6. *A locally convex function on $(0, 1)$ is convex.*

We eliminate a degenerate case before proceeding. If $\phi_{t+1}(n) = 0$, then $\phi_{t+1}(s) = 0$ for $s \geq n$, and H_{24} ends up being the 0 function. Then, $\max\{H_{23}, H_{24}\} = H_{23}$ is convex. So assume that $\phi_{t+1}(n) > 0$.

If H_{24} were locally convex, then it would immediately follow that $\max\{H_{23}, H_{24}\}$ is also locally convex. Unfortunately, H_{24} may fail to be convex in some neighborhood. We instead show that in any neighborhood, either H_{24} is convex, or $H_{23} \geq H_{24}$, which suffices. The conditions for each fact to hold are given in the next two lemmas. Since we are in the non-degenerate case ($\phi_{t+1}(n) > 0$), we can algebraically simplify the conditions by introducing some notation. To that end, we define functions $f, g : (0, 1) \to \mathbb{R}$

$$f(x) = \frac{\phi_{t+1}(n+1+x)}{\phi_{t+1}(n)}, \qquad g(x) = \frac{f(x) - 1}{1 + x} \tag{16}$$

and we continuously extend them at 0.

Lemma 7. *Let f, g be as in (16). Then, $\max\{H_{23}, H_{24}\} = H_{23}$ at a point $(n + x)$ if $g(0) \geq g(x)$.*

Proof. Using ϕ_{t+1} is decreasing, $f(0) \leq \frac{\phi_{t+1}(n+1)}{\phi_{t+1}(n)}$. The rest is simple algebra. □

Lemma 8. *Let f, g be as in (16). Then, the left derivative f^L of f exists. Further, if $g(x) \leq f^L(x)$ in some open set U, then H_{24} is convex in the neighborhood $n + U$.*

The conditions in Lemmas 7 and 8 motivate the following definition.

Definition 1. Let $f : [0, 1) \to \mathbb{R}$ have a left-derivative f^L at all points, and define $g : [0, 1) \to \mathbb{R}$ as in (16). Then, f satisfies the *max-convex condition* if around every point there is a neighborhood $U \subseteq [0, 1)$ for which at least one of the following holds:

- $\forall x \in U : g(0) \geq g(x)$.
- $\forall x \in U : g(x) \leq f^L(x)$.

If f satisfies the max-convex condition, then our proof is complete, since then either Lemma 7 or Lemma 8 will apply. In either case, $\max\{H_{23}, H_{24}\}$ is locally convex. A picture providing intuition for why this might happen is given in Fig. 1.

Continuing with our proof, observe that (16) defines f to be a positive scaling of ϕ_{t+1}, which is convex by the inductive assumption. By construction, f is continuous at 0 and hence convex in [0, 1). By Theorem 10.1 of [8], convexity of f implies continuity in (0, 1) as well. It turns out that by the next lemma these properties are sufficient.

Lemma 9. *Every convex, continuous $f : [0, 1) \to \mathbb{R}$ satisfies the max-convex condition.*

We are now done showing Lemma 5, except for proving Lemmas 6, 8 and 9, which we do next. We will use the following standard fact about convex functions (Theorems 23.1 and 24.1 in [8]).

Lemma 10. *If f is convex in a neighborhood, then its left derivative f^L exists and may be defined as*

$$f^L(x) = \sup_{y<x} \frac{f(x) - f(y)}{x - y}. \tag{17}$$

Further, f^L is non-decreasing and left continuous.

Proof of Lemma 6. Suppose a function F is convex on (a, b) and (c, d) with $a < c < b < d$. We show F is convex on (a, d). Take any three points $x < y < z \in (a, d)$. It suffices to show $s_{x,y} \leq s_{y,z}$ where $s_{p,q}$ denotes the slope between points $(p, F(p))$ and $(q, F(q))$. Consider any two points $u, v \in (c, b)$. Let the set of five points $\{x, y, z, p, q\}$ in increasing order be $p_1, \ldots, p_5$. Then every three adjacent points lie entirely in (a, b) or (c, d); hence the slopes $s_{p_i,p_{i+1}}$ are increasing, and it follows that $s_{x,y} \leq s_{y,z}$ will hold.

Now consider any compact set $[a, b]$ with $0 < a \leq b < 1$. Since F is locally convex, every point in (0, 1) has an open interval containing it where F is convex. These form an open cover of $[a, b]$ and hence there is a minimal finite sub-cover $(a_1, b_1), \ldots, (a_N, b_N)$ with $a_1 < \cdots < a_N$ and $b_1 < \cdots < b_N$ by minimality. Using the procedure outlined above, we may conclude that F is convex over $[a, b]$. Since this holds for arbitrary $0 < a$ and $b < 1$, F is convex over (0, 1). □

Proof of Lemma 8. As noted in the proof of Lemma 5, f is convex, so that by Lemma 10, its left derivative exists. Next observe that function h, defined as

$$h(x) \triangleq \frac{1 + xf(x)}{1 + x} = \frac{H_{24}(n + x)}{\phi_{t+1}(n)},$$

is convex in a neighborhood U if and only if H_{24} is convex in $n + U$ (for geometric intuition about h, refer to Fig. 1). For any points $0 < x < y < 1$, and convex combination $z = \lambda x + \mu y$, we get, after some algebra,

$$\lambda h(x) + \mu h(y) - h(z) = \frac{\lambda\mu(y - x)}{1 + z}(g(y) - g(x)) + \frac{z(\lambda f(x) + \mu f(y) - f(z))}{1 + z}.$$

The second term is non-negative since f is convex, and the first term is non-negative if g is non-decreasing. Hence h, and thus H_{24}, is convex in a region where g is not decreasing, which occurs if $0 \leq g^L(x) = \frac{f^L(x) - g(x)}{1+x}$, i.e., $f^L(x) \geq g(x)$. □

Proof of Lemma 9. We will need the following fact. For any points $x < y \in (0, 1)$

$$g(y) \text{ is a weighted average of } g(x) \text{ and } \frac{f(y) - f(x)}{y - x}. \tag{18}$$

We take cases to show that for every point x, there is a neighborhood U containing it where either $g(0) \geq g(y) \forall y \in U$, or $f^L(y) \geq g(y) \forall y \in U$.

case 1: $g(0) > g(x)$: By the continuity of f and hence g, we get $g(0) \geq g(y)$ for y in an interval containing x.

case 2: $g(0) < g(x)$: We have

$$g(x) < \frac{f(x) - f(0)}{x} \leq f^L(x),$$

since the first inequality follows from (18), and the second one from (17). By left continuity, $f^L(y) > g(y)$ in a left neighborhood of x. For any $y > x$,

$$\begin{aligned} g(y) &\leq \max\left\{g(x), \frac{f(y) - f(x)}{y - x}\right\} \quad \text{(by (18))} \\ &\leq \max\{g(x), f^L(y)\} \quad \text{(by (17))} \\ &= f^L(y). \end{aligned}$$

The last equality holds since f^L is increasing and $f^L(x) > g(x)$.

We have therefore shown that $f^L(y) \geq g(y)$ holds in a neighbourhood of x.

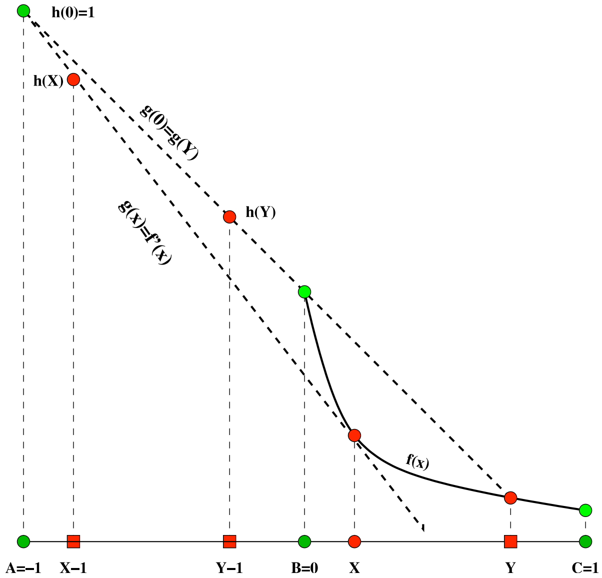

Fig. 1. The diagram shows how any convex, continuous $f : [0, 1) \rightarrow \mathbb{R}$ satisfies the *max-convex condition* in Definition 1. The slopes of the dotted lines trace the function g, while the bold curved line indicates f. X is the x-coordinate of the point of contact of the tangent from $(A, 1)$ to f. The value Y is the x-coordinate of the point where the line joining $(A, 1)$ and $(B, f(0))$ hits the curve f again. For every point z in the region (B, Y), $g(0) \geq g(z)$. The other case, i.e. $g(z) \leq f^L(z)$ occurs for every point in the region (X, C). Also included in the figure is a geometric intuition for the function $h(x) = \frac{1+xf(x)}{1+x}$, used in the proof of Lemma 8.

case 3: $g(0) = g(x)$: We have

$$g(x) = \frac{f(x) - f(0)}{x} \leq f^L(x),$$

since, the equality follows from (18), and the inequality from (17). If strict inequality holds, then we are done as in case 2. Otherwise we have $f^L(x) = \frac{f(x)-f(0)}{x}$. By Lemma 10, $f^L(x) = \sup_{y<x} \frac{f(y)-f(x)}{y-x}$, so that for any $y < x$,

$$\frac{f(x) - f(y)}{x - y} \leq \frac{f(x) - f(0)}{x}.$$

If strict inequality holds then, since $\frac{f(x)-f(0)}{x}$ is a weighted average of $\frac{f(x)-f(y)}{x-y}$ and $\frac{f(y)-f(0)}{y}$, we get

$$f^L(x) = \frac{f(x) - f(0)}{x} < \frac{f(y) - f(0)}{y} \leq f^L(y),$$

a contradiction, since f convex implies that f^L is non-decreasing. Hence $\frac{f(x)-f(y)}{x-y} = \frac{f(x)-f(0)}{x}$ for all $y < x$ and the segment of the curve f between $(0, x)$ is a straight line. It follows from (18) that $g(y) = \frac{f(x)-f(0)}{x}$, which in turn is equal to $f^L(y)$, for y in a left neighborhood around x; since $f^L(x) \geq g(x)$ implies that $f^L(y) \geq g(y)$ for y in a right neighborhood of x (as shown in case 2), we have $f^L(y) \geq g(y)$ in some neighborhood of x.

We have considered all cases. The proof follows. □

7. Conclusion

In this paper we designed the optimal deterministic master algorithm for continuous experts in the mistake bounded model and computed the exact worst case error it suffers. Computing the optimal random master algorithm against continuous experts is an open question. A combination of our techniques and the binning algorithm [1] might prove useful there.

Acknowledgements

Thanks to Jake Abernethy and Yoav Freund for many helpful discussions. This research was supported by NSF grants IIS-0325500.

Appendix. Lower bounds

We establish exact tightness of our upper bound (5). As a consequence we are able to conclusively show that abstaining experts are as powerful as continuous experts for sufficiently many experts, as well as show that binary experts are strictly less powerful (although only slightly) than abstaining experts for infinitely many games. Our arguments are based on minor modifications of the reasoning in Section 2.3 of [3] and [10].

We setup some notation. An (m, k) game is an on-line expert game with m experts and k mistake bound. The optimal number of mistakes made by the Master in an (m, k) game assuming that optimal play is denoted by $q^*_{\text{abs}}(m, k)$ and $q^*_{\text{bin}}(m, k)$ for abstaining and binary experts, respectively. We define that $\text{Abs}(q, k) \triangleq 2^q/\binom{q+1}{\le k}$, and $\text{Bin}(q, k) \triangleq 2^q/\binom{q}{\le k}$. Then, the upper bounds obtained by us (5) and the binomial weights (BW) [3] algorithm (3) are given by $\max\{q : \text{Abs}(q, k) \le m\}$ and $\max\{q : \text{Bin}(q, k) \le m\}$, respectively.

A.1. Bounds are tight for abstaining

Theorem 8. *For any k and $m > 2^{2^k}$, let $\text{Abs}(q, k) \le m < \text{Abs}(q+1, k)$. Then,*

(1) $q \le q^*_{\text{abs}}(m, k) + 1$.
(2) *If $m \ge \text{Abs}(q, k) + 2^k$, then $q = q^*_{\text{abs}}(m, k)$.*

It is not difficult to check, that for $q \gg k$, $\text{Abs}(q+1, k) - \text{Abs}(q, k) \gg 2^k$, so that condition 2 is satisfied by almost all games.

Proof of Theorem 8. We assume familiarity with Section 2.3 of [3] and [10]. Consider the following chip game with two players Paul and Carole. There are $k + 2$ bins on the *non-negative integer line*, initially all at 0. In every round, Paul *abstains* on an arbitrary subset of the chips and splits the remaining chips arbitrarily into two sets. Then, Carole chooses one of the two sets and advances each chip in it by one. Abstaining twice on the same chip causes it to advance by one bin. The game ends when all chips are located beyond k. Carole wants the game to end soon, whereas Paul wants it to drag. Reasoning very similar to that in Section 2.3 of [3] shows that, assuming optimal play, the number of rounds in the chip game is $q^*_{\text{abs}}(m, k)$. In the rest of the proof, we describe a strategy for Paul that ensures that the game lasts for at least $q - 1$ rounds, and when condition 2 is met, for q rounds.

The strategy consists of three stages. In the first two stages, Paul never abstains, so the state of the game can be described by a configuration $I = (I_0, \dots, I_k)$, where I_j denotes the number of chips in position j (chips beyond k do not matter). For any r, define the weight $W_r(I)$ of configuration I to be $\sum_j I_j \binom{r+1}{\le j}$. Chips in position k are called pennies. At the end of the first two stages, there will be at most one non-penny remaining.

By choice of q, the weight of the initial configuration given by W_q is $m2^q$. The first stage lasts for k steps at the end of which, like in the proof of Theorem 5 in [3], Paul can reach a configuration I^k where $W_{\tilde{q}}(I^k) = 2^{\tilde{q}}$, and $I^k_k > c(k)\tilde{q}^k$, where $c(k)$ is sufficiently large. If condition 2 holds then $\tilde{q} = q - k$ or else $\tilde{q} = q - k - 1$. We will show that the game lasts for $\tilde{q}$ more rounds which will complete the proof of the theorem. Henceforth, the weight of a configuration is given by W_r where r is the number of rounds remaining.

Next, in the second stage, we apply fictitious play as described in [10]. The analysis of the First Steps, Middle, Late Middle and Early End stages in the proof of **Main Theorem** in [10] carries through with our modified weight function W as well, as straightforward calculations show. Therefore, we halve the weight of the configuration every round, until, as [10] shows, we reach a configuration with at most one non-penny.

Finally, in the third stage, we modify the **Endgame Lemma** of [10] in Lemma 11 to finish off the proof. This is the only place where Paul might use abstaining moves. □

Lemma 11. *Let $(x_0, \dots, x_k)$ be a configuration with $x_0 \le 1$, $x_1 = \dots = x_{k-1} = 0$ and $W_j(x_0, \dots, x_k) = 2^j$. Then Paul can play for j more rounds.*

Proof. Note that $W_j(x_0, \dots, x_k) = x_0\binom{j+1}{\le k} + x_k$. If $x_0 = 0$, then $x_k = 2^j$, and Paul may continue halving for j more rounds. So assume that $x_0 > 0$. If $j \le 2k$, then Paul may use abstaining moves on the chip at position 0 to make the game last for at least $2k \ge j$ moves. If $j > 2k$, then both $\binom{j}{\le k}, \binom{j}{\le k-1}$, are less than 2^{j-1}. Since $\binom{j+1}{\le k} = \binom{j}{\le k} + \binom{j}{\le k-1}$, there is a way to split the chips into two parts of exactly equal weight. The proof follows by induction. □

A.2. Abstaining is continuous

Whenever condition 2 of Theorem 8 is satisfied, the upper bound for continuous experts matches the lower bound for abstaining experts, and hence both classes of experts are equally powerful. For number-theoretic reasons, the upper-bound might occasionally be one less than the optimum. The same situation occurs for the BW upper bound and the true optimum in games against binary experts. Appendix A of [3] contain an enhanced version of the BW algorithm which always achieves the exact optimum. It is straightforward to check that our algorithms for continuous and abstaining experts may be enhanced similarly to achieve exactly optimum performance for all (m, k) games with $m > 2^{2^k}$. This implies, that whenever m is sufficiently large with respect to k, abstaining and continuous experts are game-theoretically equally powerful.

A.3. Binary less powerful than abstaining

Basic calculations show the next useful Lemma.

Lemma 12. *The following hold.*

(1) *For* $q > 2^k$, $\mathrm{Bin}(q, k) < \mathrm{Abs}(q + 1, k) < \mathrm{Bin}(q + 1, k)$
(2) *For* $q \gg k$, $\mathrm{Bin}(q + 1, k) - \mathrm{Abs}(q + 1, k) \gg 2^k$.

A straightforward application of the above and Theorem 8 now yields the following separation, which is the best possible according to point 1 in Lemma 12.

Corollary 2. *For any k, define*

$$M_k \triangleq \cup_{q>2^k}\{\mathrm{Abs}(q + 1, k) + 2^k, \ldots, \mathrm{Bin}(q + 1, k) - 1\}.$$

Then, M_k *has constant density, and if* $m \in M_k$, $q^*_{\mathrm{abs}}(m, k) = q^*_{\mathrm{bin}}(m, k) + 1$.

References

[1] Jacob Abernethy, John Langford, Manfred K. Warmuth, Continuous experts and the binning algorithm, in: COLT, 2006, pp. 544–558.
[2] Nicolò Cesa-Bianchi, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, Manfred K. Warmuth, How to use expert advice, Journal of the ACM 44 (3) (1997) 427–485.
[3] Nicolò Cesa-Bianchi, Yoav Freund, David P. Helmbold, Manfred K. Warmuth, On-line prediction and conversion strategies, Machine Learning 25 (1996) 71–110.
[4] Yoav Freund, Boosting a weak learning algorithm by majority, Information and Computation 121 (2) (1995) 256–285.
[5] Yoav Freund, Manfred Opper, Continuous drifting games, Journal of Computer and System Sciences (2002) 113–132.
[6] Yoav Freund, Robert E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences 55 (1) (1997) 119–139.
[7] Nick Littlestone, Manfred K. Warmuth, The weighted majority algorithm, Information and Computation 108 (1994) 212–261.
[8] R. Tyrell Rockafellar, Convex Analysis, Princeton University Press, 1970.
[9] Robert E. Schapire, Drifting games, Machine Learning 43 (3) (2001) 265–291.
[10] Joel Spencer, Ulam's searching game with a fixed number of lies, Theoretical Computer Science 95 (2) (1992) 307–321.