Machine Learning 35, 247–282 (1999)

Derandomizing Stochastic Prediction Strategies

V. VOVK vovk@dcs.rhbnc.ac.uk
Department of Computer Science, Royal Holloway, University of London, Egham, Surrey TW20 0EX, UK

Editor: John Shawe-Taylor

Abstract. In this paper we continue study of the games of prediction with expert advice with uncountably many experts. A convenient interpretation of such games is to construe the pool of experts as one “stochastic predictor”, who chooses one of the experts in the pool at random according to the prior distribution on the experts and then replicates the (deterministic) predictions of the chosen expert. We notice that if the stochastic predictor’s total loss is at most L with probability at least p then the learner’s loss can be bounded by $cL + a \ln \frac{1}{p}$ for the usual constants c and a. This interpretation is used to revamp known results and obtain new results on tracking the best expert. It is also applied to merging overconfident experts and to fitting polynomials to data.

Keywords: on-line learning, prediction with expert advice, tracking the best expert, regression

1. Introduction

Making rational decisions is a central problem of both science and everyday life. Unfortunately, it is often difficult to readily choose the best course of action, and in such cases we are left with a more or less extensive family of potentially good strategies. At the end of the day, however, one strategy must be chosen (which may or may not be an element of this family); and the goal of the theory of prediction with expert advice is to give procedures for replacing a family of decision strategies with one strategy which never performs much worse than the best strategy in the family. At first this goal might seem too ambitious, but it has turned out that it is feasible for surprisingly wide classes of loss functions and families of strategies (DeSantis, Markowsky, & Wegman, 1988; Littlestone & Warmuth, 1994; Vovk, 1990, etc.).

In the usual terminology, the family of strategies to be merged is called the “pool of experts”; we can imagine that each strategy in this family is advocated by some “expert”, and our problem becomes that of merging opinions of different experts. Often “experts” are just elements of this metaphorical picture, and the theory of prediction with expert advice is applicable much more widely than its name suggests.

Remark 1. This “expert” terminology is particularly inconvenient for the purposes of this paper since we will need to use the word “expert” in a different, though related, sense; we will usually use the phrase “elementary predictor” to refer to the elements of our family of strategies. The name “theory of prediction with expert advice”, which we will still use, is also misleading in another respect. Although the results of this theory have been mostly applied to the problems of prediction, actually they are more widely applicable: for example, Cover’s (Cover & Ordentlich, 1996) universal portfolio algorithm deals with merging strategies for trading in securities markets.

A standard algorithm for predicting with expert advice is the Aggregating Algorithm (AA) proposed in Vovk (1990, 1998); this algorithm generalizes, e.g., the Bayesian merging scheme (DeSantis, Markowsky, & Wegman, 1988), the Weighted Majority Algorithm (Littlestone & Warmuth, 1994; Vovk, 1992), and Cover's (Cover & Ordentlich, 1996) universal portfolio algorithm. The AA works well in many situations; it was felt, however, that for the important problem of tracking the best expert (Littlestone & Warmuth, 1994; Herbster & Warmuth, 1995; Freund et al., 1997) the AA was not sufficient, and modifications of the AA or entirely different algorithms were used. In this paper we show that a general form of the AA (described in, e.g., Vovk (1998), Appendix A) can be applied to the problem of tracking the best expert (Section 3.3) and to several other related problems. This general form involves uncountably many elementary predictors; applications of the AA to uncountable pools of elementary predictors have been considered earlier by, e.g., Cover and Ordentlich (1996) and Freund (1996). (Cover developed the universal portfolio algorithm used by Cover & Ordentlich (1996) independently of the general AA.)

The main technical results of this paper are Theorems 2–6, but we consider its principal contribution to be the stochastic interpretation (described in Section 2.3) of the AA.

For further information on the theory of prediction with expert advice, the reader can also consult, e.g., (Auer & Long, 1994; Cesa-Bianchi, Helmbold, & Panizza, 1996; Feder, Merhav, & Gutman, 1992; Yamanishi, 1995).

2. Aggregating algorithm

2.1. Protocol

Our learning protocol involves three players, *Learner* (the decision maker), *Pool* (the family of elementary predictors), and *Reality*. Learner is required to make *predictions* (or, more generally, to take actions) in some *prediction space* Γ and Reality is required to choose *outcomes* in an *outcome space* Ω; the loss suffered by Learner who makes prediction γ when the outcome is ω is $\lambda(\omega, \gamma)$, where $\lambda : \Omega \times \Gamma \to [0, \infty]$ is a fixed *loss function*. We will call the triple $(\Omega, \Gamma, \lambda)$ our *game*. Our assumptions about the game $(\Omega, \Gamma, \lambda)$ are the same as in (Vovk, 1998):

1. Γ is a compact topological space.
2. For each ω, the function $\gamma \mapsto \lambda(\omega, \gamma)$ is continuous.
3. There exists γ such that, for all ω, $\lambda(\omega, \gamma) < \infty$.
4. There exists no γ such that, for all ω, $\lambda(\omega, \gamma) = 0$.

The final parameter of our protocol is a measurable set Θ (i.e., Θ is a set equipped with a σ-algebra $\mathcal{F}$), which will also be called the *pool*. Intuitively, the elements $\theta \in \Theta$ of this set are the names of the elementary predictors whose predictions we would like to combine.

Learner interacts with Pool and Reality in the following way. At each *trial* t, $t = 1, 2, \ldots$.

- Pool makes a prediction ξ_t, which is a measurable function $\xi_t : \Theta \to \Gamma$ (Γ is equipped with the σ-algebra generated by the open sets). The value $\xi_t(\theta)$, $\theta \in \Theta$, is interpreted as the prediction made by the elementary predictor θ in the pool Θ.

- Learner makes his own prediction $\gamma_t \in \Gamma$.
- Reality chooses some outcome $\omega_t \in \Omega$.
- Every elementary predictor $\theta \in \Theta$ incurs loss $\lambda(\omega_t, \xi_t(\theta))$ and Learner incurs loss $\lambda(\omega_t, \gamma_t)$.

Remark 2. We have merged all elementary predictors into a single player, Pool. Actually Learner can be considered to be playing against all other players who are able to collude; therefore, we could have gone even further merging Pool and Reality into a single player. Such merging, however, would be counterintuitive (as noted by a referee), since the learner (in a broader sense) usually *chooses* a suitable pool of elementary predictors to solve her problems.

2.2. Description of the AA

We begin with a brief description of the idea behind the AA. For simplicity, suppose there are only finitely many elementary predictors (or “experts”). Initially every elementary predictor is assigned some prior weight (for example, if we do not have any *a priori* information about the elementary predictors, it is natural to take all weights equal). After every trial the weights of the elementary predictors are recomputed so as to reflect their performance: the larger an elementary predictor’s loss the more his weight decreases. (Specifically, the weight of the elementary predictor who suffers loss l is multiplied by β^l, where β is a fixed constant between 0 and 1.) When making a prediction, Learner “merges” the predictions suggested by the elementary predictors for this trial, taking into account both the elementary predictors’ advice and the current weights of the elementary predictors. Merging is done in two steps:

- first Learner computes a “weighted average” (here “average” does not mean an arithmetic mean) of the suggested predictions; this weighted average might fail to qualify as an “allowed prediction”;
- after that, the “pseudoprediction” computed at the first step is converted to an allowed prediction (i.e., an element of Γ).

Next we will give a formal description of the AA; after that we will demonstrate how it works on a very simple example.

2.2.1. Formal description. To run the AA, we need to specify three elements. The first of them, which we will call the *exponential learning rate*, is a constant $\beta \in \,]0, 1[$; the parameter β determines how fast the AA learns. The second is the *prior distribution* $\mathbf{P}$ on Θ; it specifies the initial weights assigned to the elementary predictors. The final element is the *substitution function*. To explain what a substitution function is, we need one more definition. A *pseudoprediction* is defined to be any function of the type $\Omega \to [0, \infty]$. An allowed prediction $\gamma \in \Gamma$ is identified with the pseudoprediction g defined by $g(\omega) := \lambda(\omega, \gamma)$; we will be interested in pseudopredictions which are mixtures, in some sense, of allowed predictions. A *substitution function* is a function Σ that maps every pseudoprediction

$g:\Omega\to[0,\infty]$ into an allowed prediction $\Sigma(g)\in\Gamma$. The AA imposes strong restrictions on the choice of the substitution function, which we will discuss later; for the time being, we assume that some substitution function Σ is fixed. Now we have all we need to describe how the AA works.

Put $\mathbf{P}_0 := \mathbf{P}$. At every trial $t = 1, 2, \ldots$ Learner updates the elementary predictors' weights as follows:

$$\mathbf{P}_t(\mathrm{d}\theta) := \beta^{\lambda(\omega_t,\xi_t(\theta))}\mathbf{P}_{t-1}(\mathrm{d}\theta) \tag{1}$$

(thus sharply decreasing the weights of the elementary predictors θ whose predictions $\xi_t(\theta)$ lead to large losses $\lambda(\omega_t, \xi_t(\theta))$). More explicitly, Eq. (1) can be rewritten as

$$\mathbf{P}_t(A) := \int_A \beta^{\lambda(\omega_t,\xi_t(\theta))}\mathbf{P}_{t-1}(\mathrm{d}\theta),$$

for any measurable $A\subseteq\Theta$.

The prediction made by the AA at trial t is obtained from the weighted average (see (2) below) of the elementary predictors' predictions by applying the substitution function:

$$\gamma_t := \Sigma(g_t),$$

where the pseudoprediction g_t is defined by

$$g_t(\omega) := \log_\beta \int_\Theta \beta^{\lambda(\omega,\xi_t(\theta))}\mathbf{P}^*_{t-1}(\mathrm{d}\theta) \tag{2}$$

and $\mathbf{P}^*_{t-1}$ are the normalized weights, $\mathbf{P}^*_{t-1}(\mathrm{d}\theta) := \mathbf{P}_{t-1}(\mathrm{d}\theta)/\mathbf{P}_{t-1}(\Theta)$ (assuming that the denominator is positive; its being zero means that $\mathbf{P}$-almost all elementary predictors suffer infinite loss).

We will now discuss suitable choices of the substitution function Σ. Our choice of Σ will depend on β. First we define the important notion of the *mixability curve* $c(\beta)$. For any $\beta\in\,]0,1[$ we put

$$c(\beta) := \inf\left\{c \mid \forall P\ \exists\delta\in\Gamma\ \forall\omega\colon \lambda(\omega,\delta)\le c\log_\beta\int_\Gamma \beta^{\lambda(\omega,\gamma)}P(\mathrm{d}\gamma)\right\}, \tag{3}$$

where P ranges over all probability distributions in Γ and $\inf\emptyset := \infty$. (In (Vovk, 1998) we allowed P to range over only *simple* probability distributions, but it can be shown that, under mild regularity conditions, these two definitions are equivalent; e.g., it is sufficient to assume that the topology on Γ is generated by a metric.) Under our assumptions, the infimum in (3) is attained. A related function is

$$a(\beta) := \frac{c(\beta)}{\ln\frac{1}{\beta}}.$$

As shown by Vovk (1998), $c(\beta)$ and $a(\beta)$ are continuous and monotonic functions ($c(\beta)$ nonincreasing and $a(\beta)$ nondecreasing).

Remark 3. Our definition of $c(\beta)$ does not take account of the fact that typically the AA is applied in situations when something is known about Pool's strategy. In such cases it might be beneficial to consider only P which can appear as the images under the mapping $\theta \mapsto \xi_t(\theta)$ of the normalized weights $\mathbf{P}^*_{t-1}$.

We will always assume that our substitution function $\Sigma = \Sigma_\beta$ satisfies

$$\forall\beta\ \forall\omega : \lambda(\omega, \Sigma_\beta(g)) \le c(\beta)g(\omega) \tag{4}$$

for any pseudoprediction

$$g(\omega) := \log_\beta \int_\Gamma \beta^{\lambda(\omega,\gamma)} P(\mathrm{d}\gamma)$$

with P probability distribution in Γ; by the definition of the mixability curve, we can always satisfy this requirement. For all our results (theorems, lemmas, and corollaries) to hold assumption (4) is sufficient.

A natural way to ensure assumption (4) is to require that, for every pseudoprediction g,

$$\Sigma_\beta(g) \in \arg\min_{\gamma\in\Gamma} \sup_{\omega\in\Omega} \frac{\lambda(\omega,\gamma)}{g(\omega)} \tag{5}$$

(where $\frac{0}{0}$ is set to 0); a pleasant feature of such a definition would be the independence of Σ_β from β. This approach was used in Vovk (1998, 1997a).

A better approach, however, seems to be the following: we require that

$$\Sigma_\beta(g) \in \arg\min_{\gamma\in\Gamma} \sup_{\omega\in\Omega} (\lambda(\omega,\gamma) - c(\beta)g(\omega)) \tag{6}$$

(this min is attained under our assumptions about the game $(\Omega, \Gamma, \lambda)$) and

$$(g_1(\omega) - g_2(\omega) \text{ does not depend on } \omega) \Rightarrow (\forall\beta : \Sigma_\beta(g_1) = \Sigma_\beta(g_2)). \tag{7}$$

(Assumption (7) is always compatible with (6) but is typically incompatible with (5).) A crucial advantage of assumption (7) is that when running the AA we do not need to normalize the weights $\mathbf{P}_t(\mathrm{d}\theta)$, since the pseudoprediction

$$\omega \mapsto \log_\beta \int_\Theta \beta^{\lambda(\omega,\xi_t(\theta))} \mathbf{P}_{t-1}(\mathrm{d}\theta)$$

calculated from the unnormalized weights will differ from the pseudoprediction (2) calculated from the normalized weights by only an additive constant. Besides avoiding normalization of the weights at every trial, which is often computationally difficult, this way of defining the substitution function can lead to significant simplifications of the AA in particular applications; see, e.g., Vovk (1997b) or the algorithms for fitting polynomials (especially Remark 9) below.

In the rest of the paper we will always assume (6) and (7), though these assumptions will only be essential in Section 4. We usually drop the index β of Σ_β.

Example. Let us consider the following very simple game $(\Omega, \Gamma, \lambda)$, which we call the *simple prediction game*:

$$\Omega = \Gamma = \{0, 1\}, \qquad \lambda(\omega, \gamma) = \begin{cases} 0, & \text{if } \omega = \gamma, \\ 1, & \text{otherwise.} \end{cases}$$

(Therefore, in this game Learner is trying to predict a binary classification, 0 or 1; at every trial she suffers a loss of 1 if she makes a mistake.) Suppose we have a finite number n of elementary predictors. To clarify the notions of pseudoprediction, substitution function, etc., in the rest of this subsection we will apply the AA to predicting in the simple prediction game using advice of n elementary predictors. (In this case the AA becomes the Weighted Majority Algorithm; see Littlestone and Warmuth (1994) and Vovk (1992).)

In the case where there are just two possible outcomes, say 0 and 1 (as in the simple prediction game), it is convenient to represent every prediction $\gamma \in \Gamma$ as the point $(\lambda(0, \gamma), \lambda(1, \gamma))$ of the (x, y)-plane. There are two allowed predictions in the simple prediction game, 0 and 1, which are depicted as small filled circles in figure 1. It is also convenient to represent every pseudoprediction $g : \Omega \to [0, \infty]$ as the point $(g(0), g(1))$ of the (x, y)-plane. Possible mixtures

$$(\log_\beta(\beta p + (1 - p)), \log_\beta(p + \beta(1 - p))) \tag{8}$$

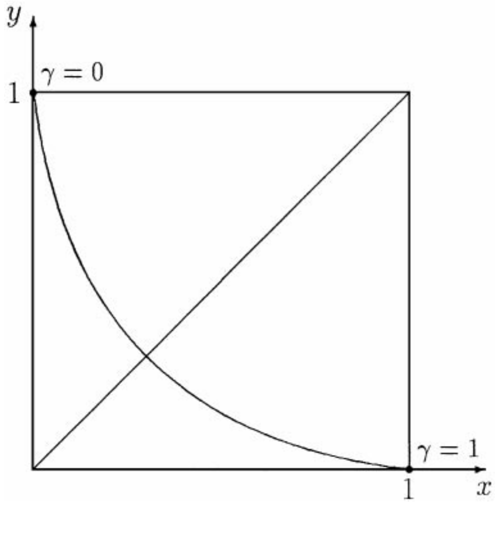


Figure 1. The substitution function for the simple prediction game.

(see (2); that expression only depends on the total weight $p = \mathbf{P}^*_{t-1}(\Theta_t(1))$ of the elementary predictors $\Theta_t(1) = \{\theta \in \Theta \mid \xi_t(\theta) = 1\}$ who predict 1 at trial t; notice that $1 - p$ is the total weight $\mathbf{P}^*_{t-1}(\Theta_t(0))$ of the elementary predictors $\Theta_t(0) = \{\theta \in \Theta \mid \xi_t(\theta) = 0\}$ who predict 0 at trial t) of the allowed predictions are shown in figure 1 by the curve (which we will call the *pseudoprediction curve*) connecting the two allowed predictions; (8) is a parametric equation of this curve (the parameter p ranges between 0 and 1).

It is clear from figure 1 that in the simple prediction game $1/c(\beta)$ (see (3)) equals the abscissa (equivalently, the ordinate) of the intersection of the pseudoprediction curve and the straight line $x = y$; this intersection corresponds to $p = 1/2$ in (8), which gives

$$\frac{1}{c(\beta)} = \log_\beta\left(\frac{1+\beta}{2}\right);$$

equivalently,

$$c(\beta) = \frac{\ln\frac{1}{\beta}}{\ln\frac{2}{1+\beta}}.$$

Now it is clear that for every β there is essentially only one substitution function Σ_β satisfying (4): if g is above the line $x = y$, Σ_β should map g to the prediction $\gamma = 0$; if g is below the line $x = y$, Σ_β should map g to the prediction $\gamma = 1$; and only if g is exactly on the line $x = y$, $\Sigma_\beta(g)$ can be defined arbitrarily. Notice that (8) being above the line $x = y$ means that $x < y$, i.e.,

$$\log_\beta\left(\beta p + (1-p)\right) < \log_\beta\left(p + \beta(1-p)\right),$$

which is equivalent to $p < 1/2$; analogously, (8) being below the line $x = y$ is equivalent to $p > 1/2$. This means that the AA predicts according to the weighted majority of the elementary predictors, which explains the name “Weighted Majority Algorithm”.

We conclude this subsection with an example of execution of the AA for the simple prediction game. Table 1 describes the AA’s behaviour in the first 3 trials in the situation where: there are three elementary predictors with equal initial weights who give predictions (1, 1, 0), (0, 1, 1), and (0, 1, 0); the actual outcomes (Reality’s moves) are 0, 1, and 1; the exponential learning rate is $\beta = 1/e$ (which corresponds to learning rate 1). The weights given in Table 1 are *unnormalized*, and instead of (8) we use its unnormalized version

$$(\log_\beta(\beta p + q), \log_\beta(p + \beta q)), \tag{9}$$

Table 1. Example of execution of the AA for the simple prediction game.

Trial no.	Pool’s weights	Pool’s predictions	Learner’s pseudoprediction	Learner’s prediction	Outcome	Pool’s losses
1	(1.00, 1.00, 1.00)	(1, 1, 0)	(−0.55, −0.86)	1	0	(1, 1, 0)
2	(0.37, 0.37, 1.00)	(0, 1, 1)	(0.14, −0.41)	1	1	(1, 0, 0)
3	(0.14, 0.37, 1.00)	(0, 1, 0)	(−0.24, 0.24)	0	1	(1, 0, 1)

where $p = \mathbf{P}_{t-1}(\Theta_t(1))$ is the total unnormalized weight of the elementary predictors $\Theta_t(1) = \{\theta \in \Theta \mid \xi_t(\theta) = 1\}$ who predict 1 at trial t and $q = \mathbf{P}_{t-1}(\Theta_t(0))$ is the total unnormalized weight of the elementary predictors $\Theta_t(0) = \{\theta \in \Theta \mid \xi_t(\theta) = 0\}$ who predict 0 at trial t. We simplify (9) to

$$\left(\ln \frac{1}{q + p/e}, \ln \frac{1}{p + q/e}\right).$$

2.3. Stochastic interpretation

In many applications specifying the prior probability distribution **P** is a difficult task. In the context of the AA, this probability distribution describes the prior weights of the elementary predictors rather than a stochastic process, but the stochastic language will still turn out to be very useful for our purposes. (Kolmogorov's axiomatization of probability, which equates the notions of probability and of normalized measure, is usually used to formalize our probabilistic intuition; in this paper we, vice versa, will use our probabilistic intuition to grasp the meaning of a normalized measure which does not describe any stochastic process.)

As our underlying probability space we take $(\Theta, \mathcal{F}, \mathbf{P})$, where $(\Theta, \mathcal{F})$ is our pool of elementary predictors and **P** is the prior distribution on the elementary predictors. Pool's prediction ξ_t, in accordance with the standard definitions of probability theory, becomes a random element of the prediction space Γ; therefore, we will also refer to Pool as *Stochastic Predictor*. In this terminology, the protocol of interaction between our three players can be rewritten as follows:

```
FOR $t = 1, 2, \ldots$
    Stochastic Predictor chooses random prediction $\xi_t$
    Learner chooses $\gamma_t \in \Gamma$
    Reality chooses $\omega_t \in \Omega$
END FOR.
```

(Therefore, the "random prediction" made by Stochastic Predictor at trial t is the function that maps every elementary predictor θ to his prediction $\xi_t(\theta)$.) This is a "perfect-information" protocol: each player can see the other players' moves; remember that Learner can see the whole mappings $\xi_t : \Theta \to \Gamma$.

The intuition behind this protocol is that Learner knows Stochastic Predictor's strategy; this strategy can depend on some information not mentioned in the protocol, but we assume that Learner can access that information as well. We explicate this assumption by allowing Learner to see the whole random prediction ξ_t (not just its realization).

For the example of the previous subsection (the simple prediction game), the underlying probability space is $(\Theta, \mathcal{F}, \mathbf{P})$, where $\Theta = \{1, \ldots, n\}$ is the set of all elementary predictors, $\mathcal{F}$ is the family of all subsets of Θ, and **P** is the uniform distribution in Θ. The random prediction ξ_t at trial t is the function that maps every elementary predictor $i \in \Theta$ to his prediction $\xi_t(i)$. More complicated examples are given in Section 3.

The name “stochastic predictor” can be easily misinterpreted, because nothing prevents traditional “experts” (as in, e.g., Cesa-Bianchi et al., 1993) from using randomized strategies. The distinctive feature of our “stochastic predictor” is that he discloses how he randomizes and not just the outcome of the randomization.

It is important that, as in probability theory, we will rarely need to explicitly specify the underlying probability space $(\Theta, \mathcal{F}, \mathbf{P})$.

2.4. Properties of the AA

In this subsection it will be useful to consider not only the AA, but also what we call the *Aggregating Pseudo-Algorithm* (APA); the latter makes not allowed predictions but pseudopredictions, in accordance with (2) (it is important that the elementary predictors' weights must be normalized before computing the APA's pseudoprediction). Our notation for the loss suffered by the stochastic predictor, by the AA, and by the APA over the first T trials will be

$$\begin{aligned}
\mathrm{Loss}_T(\mathrm{SP}) &:= \sum_{t=1}^{T} \lambda(\omega_t, \xi_t),\\
\mathrm{Loss}_T(\mathrm{AA}) &:= \sum_{t=1}^{T} \lambda(\omega_t, \gamma_t),\\
\mathrm{Loss}_T(\mathrm{APA}) &:= \sum_{t=1}^{T} g_t(\omega_t),
\end{aligned}$$

respectively (the exponential learning rate and stochastic prediction strategy to which the AA or the APA are applied will be always clear from the context). Notice that $\mathrm{Loss}_T(\mathrm{SP})$ is a random variable, whereas $\mathrm{Loss}_T(\mathrm{AA})$ and $\mathrm{Loss}_T(\mathrm{APA})$ are just numbers.

To clarify the difference between the AA and the APA, in Table 2 we give the losses of these two algorithms in the situation of Table 1 (we do not duplicate the information given in Table 1). For computing the pseudoprediction in Table 2 we use formula (8), which simplifies to

$$\left(\ln \frac{e}{e - ep + p}, \ln \frac{e}{ep + 1 - p}\right).$$

Table 2. AA vs. APA in the situation of Table 1.

Trial no.	Normalized Pool's weights	APA's pseudoprediction	APA's loss	AA's loss
1	(0.33, 0.33, 0.33)	(0.55, 0.24)	0.55	1
2	(0.21, 0.21, 0.58)	(0.69, 0.14)	0.14	0
3	(0.09, 0.24, 0.67)	(0.17, 0.65)	0.65	1

The basic result is that the performance of the APA is the average of Stochastic Predictor's performance in the following sense (**E** stands for the expectation with respect to **P**):

Lemma 1. *For any exponential learning rate* $\beta \in \,]0, 1[$ *and trial* $T = 1, 2, \ldots,$

$$\mathrm{Loss}_T(\mathrm{APA}) = \log_\beta \mathbf{E}\beta^{\mathrm{Loss}_T(\mathrm{SP})}. \tag{10}$$

Remark 4. Rule (2) of making a pseudoprediction looks complicated, but it can be easily deduced (at least for finite Θ) from (10).

Recall that $a(\beta) := c(\beta)/\ln\frac{1}{\beta}$, where $\beta \mapsto c(\beta)$ is the mixability curve. Lemma 1 immediately implies the following results.

Corollary 1. *For any exponential learning rate* $\beta \in \,]0, 1[$ *and* $T = 1, 2, \ldots,$

$$\mathrm{Loss}_T(\mathrm{AA}) \le c(\beta) \log_\beta \mathbf{E}\beta^{\mathrm{Loss}_T(\mathrm{SP})}. \tag{11}$$

Corollary 2. *Let* $T \in \{1, 2, \ldots\}$ *and* $L \ge 0$; *the exponential learning rate is* $\beta \in \,]0, 1[$. *If* $\mathrm{Loss}_T(\mathrm{SP}) \le L$ *with probability at least* $p > 0$, *then*

$$\mathrm{Loss}_T(\mathrm{AA}) \le c(\beta)L + a(\beta)\ln\frac{1}{p}. \tag{12}$$

(Inequality (12), which is analogous to one of the inequalities in Littlestone and Warmuth (1994), follows from (11) and $\mathbf{E}\beta^{\mathrm{Loss}_T(\mathrm{SP})} \ge p\beta^L$.)

3. Examples

3.1. *Basic case*

Let us consider $n \ge 2$ *experts*; at every trial t expert i, $i = 1, \ldots, n$, submits his prediction $\xi_t(i)$ to the learner. Applying the AA to the stochastic predictor who chooses one of the n experts at random and then repeats the predictions made by the chosen expert, we deduce from Corollary 2 the known (see, e.g., Vovk (1998)) inequality

$$\mathrm{Loss}_T(\mathrm{AA}) \le c(\beta)\mathrm{Loss}_T(i) + a(\beta)\ln n, \quad \forall i, T, \tag{13}$$

where $\mathrm{Loss}_T(i) := \sum_{t=1}^T \lambda(\omega_t, \xi_t(i))$ is the loss incurred by the ith expert over the first T trials.

Remark 5. If the pool of experts is countable, we can apply the AA to the stochastic predictor who chooses expert $i = 1, 2, \ldots$ with probability p_i (where $\sum_i p_i = 1$) and then repeats the chosen expert's predictions; instead of (13) we will have

$$\mathrm{Loss}_T(\mathrm{AA}) \le c(\beta)\mathrm{Loss}_T(i) + a(\beta)\ln\frac{1}{p_i}, \quad \forall i, T.$$

3.2. Overconfident experts

To consider a slightly more complicated task, suppose the learner knows that the experts are usually too categorical (like most human experts: see Dawid (1986)). For example, in the case of the log-loss game,

$$\Omega = \{0,1\}, \quad \Gamma = [0,1],$$
$$\lambda(\omega,\gamma) = \begin{cases} -\ln\gamma, & \text{if } \omega = 1, \\ -\ln(1-\gamma), & \text{if } \omega = 0, \end{cases}$$

it might be known that all experts sometimes predict 0 or 1, and sooner or later each suffers infinite loss. The learner's goal is to make more cautious predictions and, therefore, eventually outperform even the best expert. (Analysis of this simple problem will serve as an introduction to the problems considered in the following two subsections.)

By Assumption 3 the value $c_\lambda := \inf_\gamma \sup_\omega \lambda(\omega,\gamma)$ is finite. The compactness of Γ (Assumption 1) implies that there exist $\gamma \in \Gamma$ such that $\sup_\omega \lambda(\omega,\gamma) = c_\lambda$; such γ will be called *minimax*. Let $\alpha \in]0,1[$ be a small constant (we will sometimes call it the *switching rate*). A possible strategy for the stochastic predictor is:

```
Strategy 1.   Stochastic Predictor
  chooses one of the $n$ experts at random
  AT EVERY TRIAL $t = 1, 2, \ldots$ :
     tosses a biased coin with probability of tails $\alpha$
     if heads, replicates the chosen expert's prediction
     if tails, predicts with a minimax prediction.
```

If at trials $1,\ldots,T$ some expert i suffered loss greater than some constant C on k occasions, then the probability that this randomized strategy will choose expert i and obtain "tails" on exactly those k occasions is $\frac{1}{n}\alpha^k(1-\alpha)^{T-k}$. Derandomizing our stochastic strategy with the AA and applying Corollary 2, we obtain that, for every expert i, every trial T, and every constant $C > 0$,

$$\mathrm{Loss}_T(\mathrm{AA}) \le c(\beta)\sum_{t=1}^{T}\lambda^{(C)}(\omega_t,\xi_t(i)) + a(\beta)\left(\ln n + k\ln\frac{1}{\alpha} + (T-k)\ln\frac{1}{1-\alpha}\right),$$

where

$$\lambda^{(C)}(\omega_t,\xi_t(i)) := \begin{cases} \lambda(\omega_t,\xi_t(i)), & \text{if } \lambda(\omega_t,\xi_t(i)) \le C, \\ c_\lambda, & \text{otherwise,} \end{cases}$$

and

$$k := \#\{t \in \{1,\ldots,T\} : \lambda(\omega_t,\xi_t(i)) > C\}.$$

Typically it is difficult to guess a good value for α, so a natural idea is to randomize over α as well (actually randomization over α is a way of learning the right α). The densities $\epsilon\alpha^{\epsilon-1}$ are very convenient analytically, and we will often use them, assuming $0 < \epsilon < 1$.

Strategy 2. Stochastic Predictor

```
generates $\alpha \in ]0, 1[$ from the distribution with density $\epsilon\alpha^{\epsilon-1}$
chooses one of the $n$ experts at random
AT EVERY TRIAL $t = 1, 2, \dots$ :
    tosses a biased coin with probability of tails $\alpha$
    if heads, replicates the chosen expert's prediction
    if tails, predicts with a minimax prediction.
```

Theorem 1. *When applied to Strategy* 2 *with exponential learning rate β, the AA satisfies*

$$\mathrm{Loss}_T(\mathrm{AA}) \le c(\beta)\sum_{t=1}^{T}\lambda^{(C)}(\omega_t, \xi_t(i)) + a(\beta)\left(\ln n + (k+\epsilon)\ln(T+1) + \ln\frac{1}{\epsilon\Gamma(k+\epsilon)}\right). \qquad (14)$$

Remark 6. The term $\ln\frac{1}{\epsilon\Gamma(k+\epsilon)}$ occurs in many of our results. Notice that it never exceeds $\ln\frac{1.13}{\epsilon}$; e.g., it is smaller than 4.73 when $\epsilon = 0.01$.

3.3. *Tracking the best expert*

In this subsection we consider the problem of tracking the best expert. Again we have n experts, but this time we would like to perform at every trial T almost as well as the best sequence $e_1, \dots, e_T$ of experts with a small number of switches $e_t \neq e_{t+1}$. For every $\alpha \in]0, 1[$ we consider the following strategy for the stochastic predictor:

Strategy 3. Stochastic Predictor

```
chooses one of the $n$ experts at random
AT EVERY TRIAL $t = 1, 2, \dots$ :
    replicates the chosen expert's prediction
    with probability $\alpha$ chooses a different expert at random.
```

Applying the AA to derandomize this strategy, we obtain, for any integer $T > 0$ and any sequence $E = e_1, \dots, e_T$ of T experts,

$$\mathrm{Loss}_T(\mathrm{AA}) \le c(\beta)\mathrm{Loss}_T(E) + a(\beta)\left(\ln n + k\ln(n-1) + k\ln\frac{1}{\alpha} + (T-k-1)\ln\frac{1}{1-\alpha}\right),$$

where k is the number of switches in E,

$$k := \#\{t = 1, \dots, T-1 \mid e_t \neq e_{t+1}\},$$

and $\mathrm{Loss}_T(E)$ is the loss of the "compound expert" E,

$$\mathrm{Loss}_T(E) := \sum_{t=1}^{T} \lambda(\omega_t, \xi_t(e_t)).$$

This inequality follows from Corollary 2 and the fact that our randomized strategy suffers loss $\mathrm{Loss}_T(E)$ with probability at least

$$\frac{1}{n}\left(\frac{\alpha}{n-1}\right)^k (1-\alpha)^{T-k-1} \tag{15}$$

(this expression is the probability that Strategy 3 chooses the experts exactly according to the compound expert E and is obtained by multiplying the probabilities of the following three independent events: choosing the first expert in E correctly; switching to the right experts on k occasions; refraining from switching on $T-k-1$ occasions). It was obtained by Herbster and Warmuth (1995) (Theorem 4.3), who used exactly the same algorithm (see Section 4 below), though they proposed their algorithm as an alternative to, rather than a special case of, the AA.

As in the case of Strategy 1, an interesting modification of Strategy 3 is obtained by randomizing over α with density $\epsilon\alpha^{\epsilon-1}$ (for some small $\epsilon > 0$).

```
Strategy 4.  Stochastic Predictor
  generates $\alpha \in ]0,1[$ from the distribution with density $\epsilon\alpha^{\epsilon-1}$
  chooses one of the $n$ experts at random
  AT EVERY TRIAL $t = 1, 2, \dots$ :
      replicates the chosen expert's prediction
      with probability $\alpha$ chooses a different expert at random.
```

Theorem 2. *Fed with exponential learning rate β and Stochastic Predictor's Strategy* 4, *the AA satisfies*

$$\mathrm{Loss}_T(\mathrm{AA}) \leq c(\beta)\mathrm{Loss}_T(E) + a(\beta)\Bigg(\ln n + k\ln(n-1) + (k+\epsilon)\ln T + \ln\frac{1}{\epsilon\Gamma(k+\epsilon)}\Bigg). \tag{16}$$

For example, for $\epsilon = 0.01$ this theorem gives

$$\mathrm{Loss}_T(\mathrm{AA}) \leq c(\beta)\mathrm{Loss}_T(E) + a(\beta)(\ln n + k\ln(n-1) + (k+0.01)\ln T + 4.73).$$

Notice that bound (16) is stronger than the result of Freund et al. (1997), Section 4.3 (Freund et al. consider only the log loss function, for which $c(\beta) = a(\beta) = 1$ when $\beta = e^{-1}$; their k is our k plus 1).

Remark 7. Bound (16) for $k = 0$ is not as good as the usual bound (13); its main disadvantage is the term $\epsilon \ln T$. A straightforward way to improve the algorithm's performance for $k = 0$ is to set the prior to a mixture of the unit mass at $\alpha = 0$ and the distribution with density $\epsilon\alpha^{\epsilon-1}$. Even for $k \neq 0$ one might want to improve the coefficient $(k + \epsilon)$ before $\ln T$. We cannot hope to make it less than k, because a coefficient of k appears even in the situation when T and k are known in advance (see Herbster and Warmuth (1995), Theorem 4.4). We can, however, come very close to making it k by a suitable randomization over ϵ. It can be shown that, for a suitable prior,

$$\begin{aligned}\mathrm{Loss}_T(\mathrm{AA}) \le c(\beta)\mathrm{Loss}_T(E) + a(\beta)\biggl(&\ln n + k\ln(n-1) + k\ln\frac{T}{k}\\ &+ 2.01\ln\ln\frac{T}{k} + 1.01k + C\biggr) \qquad (17)\end{aligned}$$

(C is a universal constant) when $0 < k < \frac{T}{10}$, and

$$\mathrm{Loss}_T(\mathrm{AA}) \le c(\beta)\mathrm{Loss}_T(E) + a(\beta)(\ln n + 1)$$

when $k = 0$ (see Section 5.8).

Herbster and Warmuth also consider the case of a uniformly bounded (say, by constant 1) loss function. In this case the following strategy has some advantages:

```
Strategy 5. Stochastic Predictor
  chooses one of the $n$ experts at random
  AT EVERY TRIAL $t = 1, 2, \dots$
    replicates the chosen expert's prediction suffering some loss $l$
    with probability $1 - (1 - \alpha)^l$ chooses a different expert at random.
```

(Here again $\alpha \in]0, 1[$ is a parameter of the algorithm.) Derandomizing this strategy with the AA, we easily obtain, for any $T > 0$ and $E = e_1, \dots, e_T$,

$$\begin{aligned}\mathrm{Loss}_T(\mathrm{AA}) \le\ & c(\beta)\left(\mathrm{Loss}_T(E) + 2k\right)\\ &+ a(\beta)\left(\ln n + k\ln(n-1) + k\ln\frac{1}{\alpha} + \mathrm{Loss}_T(E)\ln\frac{1}{1-\alpha}\right), \qquad (18)\end{aligned}$$

where k is the number of switches in E. (This inequality is slightly different from Theorem 5.7 of Herbster and Warmuth (1995), but their algorithm is exactly the same.) To see why (18) is true,

- notice that Strategy 5 suffers loss at most $\mathrm{Loss}_T(E) + 2k$ with probability at least

  $$\frac{1}{n}\left(\frac{\alpha}{n-1}\right)^k (1-\alpha)^{\mathrm{Loss}_T(E)}$$

  (this expression is obtained analogously to (15); the only difference is that now we estimate the probability that Strategy 5 will switch to the right expert not simultaneously with E but before it suffers a loss of 2 after E makes a switch; a loss of 2 per switch gives the extra addend $2k$, and the probability that Strategy 5 will not switch to a new expert before its loss exceeds 2 is at most $1-\alpha$, since the bound 1 on the loss function implies that just before the loss of Strategy 5 exceeds 2 it reaches at least 1);
- apply Corollary 2.

The next theorem slightly strengthens both (18) and Herbster and Warmuth's (1995) Theorem 5.7.

Theorem 3. *Let $\lambda(\omega,\gamma) \le 1$, $\forall \omega,\gamma$. Applied to Strategy* 5 *with exponential learning rate β, the AA satisfies, for any $T > 0$ and any $E = e_1, \ldots, e_T$,*

$$\begin{aligned}\mathrm{Loss}_T(\mathrm{AA}) &\le c(\beta)(\mathrm{Loss}_T(E) + k) \\ &\quad + a(\beta)\left(\ln n + k\ln(n-1) + k\ln\frac{1}{\alpha} + \mathrm{Loss}_T(E)\ln\frac{1}{1-\alpha}\right), \qquad (19)\end{aligned}$$

where k is the number of switches in E.

Further randomization over α in Strategy 5 gives

Strategy 6. Stochastic Predictor

```
generates $\alpha \in ]0,1[$ from the distribution with density $\epsilon\alpha^{\epsilon-1}$
chooses one of the $n$ experts at random
AT EVERY TRIAL $t = 1, 2, \ldots$
    replicates the chosen expert's prediction suffering some loss $l$
    with probability $1-(1-\alpha)^l$ chooses a different expert at random.
```

Theorem 4. *Let $\lambda(\omega,\gamma) \le 1$, $\forall \omega,\gamma$. The AA with exponential learning rate β applied to Stochastic Predictor's Strategy 6 satisfies*

$$\begin{aligned}\mathrm{Loss}_T(\mathrm{AA}) &\le c(\beta)(\mathrm{Loss}_T(E) + k) + a(\beta)\biggl(\ln n + k\ln(n-1) \\ &\quad + (k+\epsilon)\ln(\mathrm{Loss}_T(E) + k + 1) + \ln\frac{1}{\epsilon\Gamma(k+\epsilon)}\biggr). \qquad (20)\end{aligned}$$

Notice that, as $\mathrm{Loss}_T(E) \to \infty$, the right-hand side of this inequality asymptotically grows as $c(\beta)\mathrm{Loss}_T(E)$ for fixed n and k. (Herbster and Warmuth's (1995) Theorem 5.8 also has this property, but it requires *a priori* estimates of k and $\mathrm{Loss}_T(E)$.)

Bound (20) is good when k is small but $\mathrm{Loss}_T(E)$ can be large. If we, on the contrary, bet on achieving very small $\mathrm{Loss}_T(E)$ by increasing k, it is better to use randomization over α with density $\epsilon(1-\alpha)^{\epsilon-1}$; therefore, we consider the following strategy for the stochastic predictor.

```
Strategy 7.  Stochastic Predictor
  generates $\alpha \in ]0,1[$ from the distribution with density $\epsilon(1-\alpha)^{\epsilon-1}$
  chooses one of the $n$ experts at random
  AT EVERY TRIAL $t = 1, 2, \dots$ :
     replicates the chosen expert's prediction suffering some loss $l$
     with probability $1-(1-\alpha)^l$ chooses a different expert at random.
```

Theorem 5. *Let $\lambda(\omega,\gamma) \le 1$, $\forall \omega, \gamma$. The AA with exponential learning rate β applied to Stochastic Predictor's Strategy 7 satisfies*

$$\mathrm{Loss}_T(\mathrm{AA}) \le c(\beta)(\mathrm{Loss}_T(E)+k) + a(\beta)\Bigg(\ln n + k\ln(n-1) + (\mathrm{Loss}_T(E)+\epsilon) \times \ln(\mathrm{Loss}_T(E)+k+1) + \ln\frac{1}{\epsilon\Gamma(\mathrm{Loss}_T(E)+\epsilon)}\Bigg). \tag{21}$$

Remark 8. This subsection clearly shows the difference between the notions of expert and of elementary predictor: the latter is similar to that of compound expert.

3.4. *Fitting polynomials*

To apply the AA to the problem of estimating the right degree of a polynomial to be used for fitting the data set (cf. Vovk, 1998), we consider the following scenario. At each trial $t = 1, 2, \dots$:

- Reality chooses $x_t \in [0,1]$.
- Learner makes a guess $\hat{y}_t \in [0,1]$.
- Reality chooses $y_t \in [0,1]$.
- Learner suffers loss $(\hat{y}_t - y_t)^2$.

So the learner's task is to predict y_t given x_t. Suppose she decided to do so by fitting a polynomial

$$y = a_0 + a_1 x + a_2 x^2 + \cdots + a_i x^i$$

to her data $(x_1, y_1), \dots, (x_{t-1}, y_{t-1})$ by the least-squares method and predicting with

$$\hat{y}_t := \mathrm{trunc}_{[0,1]}\big(a_0 + a_1 x_t + \cdots + a_i x_t^i\big),$$

where

$$\mathrm{trunc}_{[0,1]}u := \begin{cases} 0, & \text{if } u < 0, \\ 1, & \text{if } u > 1, \\ u, & \text{otherwise;} \end{cases}$$

she is unsure, however, what degree i to choose; moreover, we can expect that i should increase with time. In Vovk (1998) we considered experts who always choose the same i; here we consider the following "stochastic expert":

Strategy 8. Stochastic Predictor

```
sets $i := 0$
AT EVERY TRIAL $t = 1, 2, \dots$ :
    makes prediction fitting a polynomial of degree $i$
    with probability $\alpha$ increases $i$ by 1.
```

(As usual, α is a small constant; at the first trial, when no data are available, predict with, say, $\frac{1}{2}$.) Applying the AA to derandomize this strategy, we obtain, for any integer $T > 0$ and any sequence $E = e_1, \dots, e_T$ such that $e_1 = 0$ and $e_{t+1} - e_t \in \{0, 1\}$ for all $t = 1, \dots, T - 1$,

$$\mathrm{Loss}_T(\mathrm{AA}) \le c(\beta)\mathrm{Loss}_T(E) + a(\beta)\left(k \ln \frac{1}{\alpha} + (T - k - 1) \ln \frac{1}{1 - \alpha}\right), \tag{22}$$

where $k = e_T$ is the number of switches in E. In the case of quadratic loss, a good choice for the exponential learning rate is $\beta = e^{-2}$: Haussler, Kivinen, and Warmuth (1994) show that $c(e^{-2}) = 1$; therefore, $a(e^{-2}) = \frac{1}{2}$. With this choice, (22) becomes

$$\mathrm{Loss}_T(\mathrm{AA}) \le \mathrm{Loss}_T(E) + \frac{1}{2}\left(k \ln \frac{1}{\alpha} + (T - k - 1) \ln \frac{1}{1 - \alpha}\right).$$

Further randomizing Strategy 8 over α, we obtain

Strategy 9. Stochastic Predictor

```
generates $\alpha \in ]0, 1[$ from the distribution with density $\epsilon\alpha^{\epsilon-1}$
sets $i := 0$
    AT EVERY TRIAL $t = 1, 2, \dots$ :
        makes prediction fitting a polynomial of degree $i$
        with probability $\alpha$ increases $i$ by 1.
```

Theorem 6. *When applied to Strategy* 9 *with exponential learning rate* e^{-2}*, the AA satisfies*

$$\mathrm{Loss}_T(\mathrm{AA}) \le \mathrm{Loss}_T(E) + \frac{k + \epsilon}{2} \ln T + \frac{1}{2} \ln \frac{1}{\epsilon\Gamma(k + \epsilon)}. \tag{23}$$

4. Explicit algorithms

4.1. Algorithms with fixed α

So far we have ignored the question of computational efficiency of the prediction strategies produced by our algorithms. When understood literally, all prediction strategies of the previous sections, except for the prediction strategy for the basic case, are infeasible: they involve huge pools of elementary predictors (which we did not even care to specify explicitly). Some of these prediction strategies, however, are actually quite efficient in simple games such as the log-loss game, as demonstrated by Herbster and Warmuth (1995). The idea is that instead of the weights for the elementary predictors we can consider "aggregated weights" $w_t(i)$ ($i = 1, \dots, n$ runs over the experts and $t = 1, 2, \dots$ over the trials), where $w_t(i) := \mathbf{P}_t(E_t(i))$ is the total weight of the set $E_t(i)$ of the elementary predictors for which i is the "current expert" at trial t. The weights $w_t(i)$ will always be non-negative but not necessarily normalized: $\sum_{i=1}^{n} w_t(i)$ will typically be different from 1. It will also be convenient for us to drop the assumption that the initial weights of the elementary predictors are specified as a *probability* measure: it is clear that nothing will change if all initial weights are multiplied by the same constant (see (7)).

In the case of overconfident experts (Section 3.2, Strategies 1 and 2), the current expert for an elementary predictor is the expert whose predictions are replicated when the coin lands heads; here $E_t(i)$ does not depend on t. The AA applied to Strategy 1 can be represented as follows (we assume that some fixed minimax prediction $\gamma^* \in \Gamma$ is chosen in the case of tails):

Algorithm 1. Learner

```
sets initial weights
    $w_0(i) := 1, \quad i = 1, \dots, n$
AT EVERY TRIAL $t = 1, 2, \dots$:
    outputs the prediction
        $\gamma_t := \Sigma(g_t)$,
        where $g_t : \Omega \to \mathbb{R}$ is the pseudoprediction
            $g_t(\omega) = \log_\beta \left( \alpha\beta^{\lambda(\omega,\gamma^*)} + (1-\alpha) \sum_{i=1}^{n} \beta^{\lambda(\omega,\xi_t(i))} w_{t-1}(i) \right)$
    updates the weights:
        $w_t(i) := w_{t-1}(i) \big( (1-\alpha)\beta^{\lambda(\omega,\xi_t(i))} + \alpha\beta^{\lambda(\omega,\gamma^*)} \big), \quad i = 1, \dots, n.$
```

(This weight update rule reflects the fact that fraction α of the elementary predictors in $E_t(i)$ suffer loss $\lambda(\omega, \gamma^*)$ and the other elementary predictors in $E_t(i)$ suffer loss $\lambda(\omega, \xi_t(i))$.) It is clear that when Ω is finite and small (e.g., $\Omega = \{0, 1\}$) and the substitution function Σ is efficiently computable, this algorithm is efficient: at every step t all computations can be carried out in time $O(n)$ if our computational model is strong enough (in particular, β^x can be computed in one step from x).

In the case of tracking the best expert (Section 3.3) or fitting polynomials (Section 3.4), "current expert" means the expert whose prediction is going to be replicated at this trial. The AA applied to Strategies 3 and 5 is also easy to represent in the "aggregated" form (cf. figure 1 of Herbster and Warmuth (1995)). For Strategy 3 such an aggregated form is

Algorithm 2. Learner

```
sets initial weights
    $w_0(i) := 1, \quad i = 1, \dots, n$
AT EVERY TRIAL $t = 1, 2, \dots$ :
    outputs the prediction
        $\gamma_t := \Sigma(g_t),$
        where $g_t : \Omega \to \mathbb{R}$ is the pseudoprediction
            $g_t(\omega) = \log_\beta \sum_{i=1}^{n} \beta^{\lambda(\omega,\xi_t(i))} w_{t-1}(i)$
    updates the weights in two steps:
        $w_t^*(i) := w_{t-1}(i)\beta^{\lambda(\omega,\xi_t(i))}, \quad i = 1, \dots, n,$
        $w_t(i) := (1-\alpha)w_t^*(i) + \frac{\alpha}{n-1}\sum_{j \neq i} w_t^*(j), \quad i = 1, \dots, n.$
```

(The first step of the weight update reflects the performance of elementary predictors, and the second step says that fraction $\frac{\alpha}{n-1}$ of elementary predictors in $E_{t-1}(j)$, $j \neq i$, will be in $E_t(i)$.) Analogously, for Strategy 5 we obtain

Algorithm 3. Learner

```
sets initial weights
    $w_0(i) := 1, \quad i = 1, \dots, n$
AT EVERY TRIAL $t = 1, 2, \dots$ :
    outputs the prediction
        $\gamma_t := \Sigma(g_t),$
        where $g_t : \Omega \to \mathbb{R}$ is the pseudoprediction
            $g_t(\omega) = \log_\beta \sum_{i=1}^{n} \beta^{\lambda(\omega,\xi_t(i))} w_{t-1}(i)$
    updates the weights in two steps:
        $w_t^*(i) := w_{t-1}(i)\beta^{\lambda(\omega,\xi_t(i))}, \quad i = 1, \dots, n,$
        $w_t(i) := (1-\alpha)^{\lambda(\omega,\xi_t(i))} w_t^*(i)$
            $+ \frac{1}{n-1}\sum_{j \neq i}\left(1 - (1-\alpha)^{\lambda(\omega,\xi_t(j))}\right) w_t^*(j), \quad i = 1, \dots, n.$
```

Notice that Algorithms 2 and 3 have $O(n)$ implementations as well: say, in the case of Algorithm 2, we can replace $\sum_{j \neq i} w_t^*(j)$ by $\sum_j w_t^*(j) - w_t^*(i)$ and compute $\sum_j w_t^*(j)$ only once.

When the AA is applied to Strategy 8, the pool of "experts" becomes infinite (an "expert" corresponds to any possible degree i of the fitted polynomial, $i = 0, 1, \ldots$) but at every trial only finitely many experts (at most $t + 1$ for trial t) have non-zero weights. In this case an explicit representation of the algorithm is:

Algorithm 4. Learner

```
  sets initial weights
      $w_0(0) := 1$ and $w_0(i) := 0, \quad i = 1, 2, \ldots$
  AT EVERY TRIAL $t = 1, 2, \ldots$ :
      outputs the prediction
          $\gamma_t := \Sigma(g_t) = \frac{1 - g_t(1) + g_t(0)}{2}$,
        where $g_t : \Omega \to \mathbb{R}$ is the pseudoprediction
          $g_t(\omega) = \log_\beta \sum_{i=0}^{\infty} \beta^{\lambda(\omega, \xi_t(i))} w_{t-1}(i) = \log_\beta \sum_{i=0}^{t-1} \beta^{\lambda(\omega, \xi_t(i))} w_{t-1}(i)$
      updates the weights in two steps:
          $w_t^*(i) := w_{t-1}(i) \beta^{\lambda(\omega, \xi_t(i))}, \quad i = 0, 1, \ldots, t-1,$
          $w_t(i) := (1 - \alpha) w_t^*(i) + \alpha w_t^*(i-1), \quad i = 0, 1, \ldots, t,$
        where $w_t^*(-1) := 0$ and $w_t^*(t) := 0$.
```

Since we consider only the quadratic loss function in the problem of fitting polynomials, it was possible to write down an explicit expression for the substitution function: it can be shown that it suffices to consider only the outcomes $\omega \in \{0, 1\}$, and so $p := \Sigma(g)$ should satisfy

$$(1 - p)^2 - g(1) = p^2 - g(0),$$

which gives

$$p = \frac{1 - g(1) + g(0)}{2}. \tag{24}$$

Remark 9. Using assumptions (6) and (7) rather than (5) enabled us to give an explicit expression for a substitution function, (24). Figure 2 illustrates the difference between $q = \Sigma_\beta(g)$ with Σ_β satisfying (5) and $p = \Sigma_\beta(g)$ with Σ_β satisfying (6) and (7); that figure assumes the quadratic loss function $\lambda(\omega, \gamma) = (\omega - \gamma)^2$, the binary outcome space $\Omega = \{0, 1\}$, and the continuous prediction space $\Gamma = [0, 1]$.

4.2. *Algorithms with random α*

In this subsection we consider a more complicated situation where the value of α needs to be learnt as well. Now we group our elementary predictors into groups $E_t(i, \alpha)$: for all

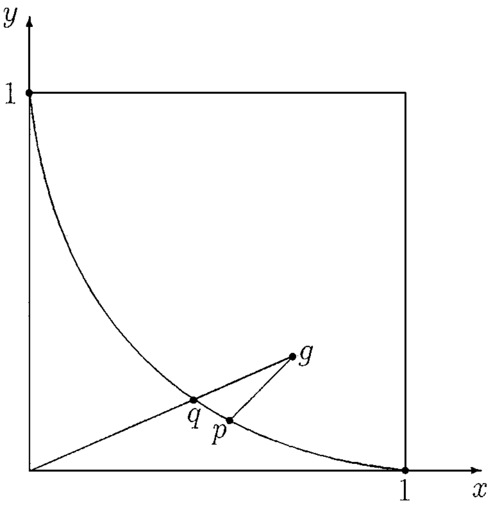


Figure 2. A pseudoprediction $g = (g(0), g(1))$ and the corresponding allowed predictions: p, computed using (6) and (7) (with $c(\beta) = 1$), and q, computed using (5) (the slope of the line connecting p and g is 1, the line connecting g and q passes through the origin of the coordinate system, and the curve connecting $(1, 0)$ and $(0, 1)$ is the set of all allowed predictions).

$t \in \{1, 2, \dots\}$, $i \in \{1, \dots, n\}$, and $\alpha \in]0, 1[$, $E_t(i, \alpha)$ are the elementary predictors whose current expert at trial t is i and whose switching rate is α; our notation for the cumulative weight of the group $E_t(i, \alpha)$ will be $w_t(i, \alpha)\,\mathrm{d}\alpha$.

In the case of Strategy 2, $E_t(i, \alpha)$ do not depend on t but $w_t(i, \alpha)$ do; the AA can be represented as follows:

```
Algorithm 5. Learner
  sets initial weights
    $w_0(i, \alpha) := \alpha^{\epsilon-1}, \quad i = 1, \dots, n, \alpha \in ]0, 1[$
  AT EVERY TRIAL $t = 1, 2, \dots$ :
    outputs the prediction
      $\gamma_t := \Sigma(g_t),$
      where $g_t : \Omega \to \mathbb{R}$ is the pseudoprediction
        $g_t(\omega) = \log_\beta \int_0^1 \sum_{i=1}^n \left(\alpha\beta^{\lambda(\omega,\gamma^*)} + (1-\alpha)\beta^{\lambda(\omega,\xi_t(i))}\right) w_{t-1}(i, \alpha)\,\mathrm{d}\alpha$
    updates the weights:
      $w_t(i, \alpha) := w_{t-1}(i, \alpha)\left((1-\alpha)\beta^{\lambda(\omega,\xi_t(i))} + \alpha\beta^{\lambda(\omega,\gamma^*)}\right),$
        $i = 1, \dots, n, \quad \alpha \in ]0, 1[.$
```

The main computational problem with this algorithm is that α ranges over the continuum $]0, 1[$; in practice, the continuous range $]0, 1[$ can be replaced by a discrete subset of $]0, 1[$.

(For further discussion, see Section 6.3.) Another solution will be discussed at the end of this subsection.

Analogously, the AA applied to Strategy 4 is

Algorithm 6. Learner
sets initial weights

$$w_0(i,\alpha) := \alpha^{\epsilon-1}, \quad i = 1,\dots,n, \quad \alpha \in]0,1[$$

AT EVERY TRIAL $t = 1, 2, \dots$:
outputs the prediction

$$\gamma_t := \Sigma(g_t),$$

where $g_t : \Omega \to \mathbb{R}$ is the pseudoprediction

$$g_t(\omega) = \log_\beta \int_0^1 \sum_{i=1}^{n} \beta^{\lambda(\omega,\xi_t(i))} w_{t-1}(i,\alpha)\,\mathrm{d}\alpha$$

updates the weights in two steps:

$$w_t^*(i,\alpha) := w_{t-1}(i,\alpha)\beta^{\lambda(\omega,\xi_t(i))}, \quad i = 1,\dots,n, \alpha \in]0,1[,$$

$$w_t(i,\alpha) := (1-\alpha)w_t^*(i,\alpha) + \frac{\alpha}{n-1}\sum_{j\neq i} w_t^*(j,\alpha),$$

$$i = 1,\dots,n, \quad \alpha \in]0,1[.$$

The description of the AA applied to Strategies 6 and 7 is almost the same and we do not spell it out, and for Strategy 9 it is:

Algorithm 7. Learner
sets initial weights

$$w_0(i,\alpha) := \alpha^{\epsilon-1}, \quad i = 1,\dots,n, \ \alpha \in]0,1[$$

AT EVERY TRIAL $t = 1, 2, \dots$:
outputs the prediction

$$\gamma_t := \Sigma(g_t) = \frac{1 - g_t(1) + g_t(0)}{2},$$

where $g_t : \Omega \to \mathbb{R}$ is the pseudoprediction

$$g_t(\omega) = \log_\beta \int_0^1 \sum_{i=0}^{t-1} \beta^{\lambda(\omega,\xi_t(i))} w_{t-1}(i,\alpha)\,\mathrm{d}\alpha$$

updates the weights in two steps:

$$w_t^*(i,\alpha) := w_{t-1}(i,\alpha)\beta^{\lambda(\omega,\xi_t(i))}, \quad i = 0,1,\dots,t-1, \ \alpha \in]0,1[$$

$$w_t(i,\alpha) := (1-\alpha)w_t^*(i,\alpha) + \alpha w_t^*(i-1,\alpha), \quad i = 0,1,\dots,t, \ \alpha \in]0,1[,$$

where $w_t^*(-1,\alpha) := 0$ and $w_t^*(t,\alpha) := 0$.

One typically wants to compute prediction γ_t for trial t in constant time (independent of t). It is possible that our algorithms with random α cannot be implemented, even on a computer able to perform operations (such as multiplication or exponentiation) on real numbers, to satisfy this requirement (and so one needs to use approximations such as replacing the continuous range of α by a discrete subset). If, however, one is willing to spend time $O(t)$ at trial t, such an implementation becomes easy for Algorithms 5 and 6. (And it is easy to implement Algorithm 7 so that the time spent for computation at trial t is $O(t^2)$.)

For simplicity we will only consider Algorithm 5. Notice that $w_t(i,\alpha)$ equals the initial weight $\alpha^{\epsilon-1}$ times a degree t polynomial of α; in other words, $w_t(i,\alpha)$ can be represented in the form

$$w_t(i,\alpha) = \alpha^{\epsilon-1}\sum_{j=0}^{t} W_t(i,j)\alpha^j.$$

In terms of the coefficients $W_t(i,j)$, Algorithm 5 can be rewritten as follows:

```
Algorithm 8.  Learner
  sets initial coefficients
      $W_0(i,0) := 1, \quad i = 1,\ldots,n$
  AT EVERY TRIAL $t = 1,2,\ldots$ :
      outputs the prediction
          $\gamma_t := \Sigma(g_t)$,
          where $g_t : \Omega \to \mathbb{R}$ is the pseudoprediction
              $g_t(\omega) = \log_\beta \sum_{i=1}^{n}\sum_{j=0}^{t-1} W_{t-1}(i,j)$
                  $\times \left(\frac{1}{j+\epsilon+1}\beta^{\lambda(\omega,\gamma^*)} + \frac{1}{(j+\epsilon)(j+\epsilon+1)}\beta^{\lambda(\omega,\xi_t(i))}\right)$
      updates the coefficients:
          $W_t(i,j) := W_{t-1}(i,j)\beta^{\lambda(\omega,\xi_t(i))} + W_{t-1}(i,j-1)\bigl(\beta^{\lambda(\omega,\gamma^*)} - \beta^{\lambda(\omega,\xi_t(i))}\bigr)$,
              $i = 1,\ldots,n, \quad j = 0,1,\ldots,t$,
              where $W_{t-1}(i,-1) := 0$ and $W_{t-1}(i,t) := 0$.
```

The only non-obvious point in this transformation is the formula for the pseudoprediction: making use of the fact that

$$\int_0^1 \alpha^c \, \mathrm{d}\alpha = \frac{1}{c+1},$$

we found:

$$\begin{aligned}
\beta^{g_t(\omega)} &= \int_0^1 \sum_{i=1}^{n} \left(\alpha\left(\beta^{\lambda(\omega,\gamma^*)} - \beta^{\lambda(\omega,\xi_t(i))}\right) + \beta^{\lambda(\omega,\xi_t(i))}\right)\alpha^{\epsilon-1} \sum_{j=0}^{t-1} W_{t-1}(i, j)\alpha^j \, \mathrm{d}\alpha \\
&= \sum_{i=1}^{n} \sum_{j=0}^{t-1} W_{t-1}(i, j)\left(\frac{1}{j+\epsilon+1}\left(\beta^{\lambda(\omega,\gamma^*)} - \beta^{\lambda(\omega,\xi_t(i))}\right) + \frac{1}{j+\epsilon}\beta^{\lambda(\omega,\xi_t(i))}\right) \\
&= \sum_{i=1}^{n} \sum_{j=0}^{t-1} W_{t-1}(i, j)\left(\frac{1}{j+\epsilon+1}\beta^{\lambda(\omega,\gamma^*)} + \frac{1}{(j+\epsilon)(j+\epsilon+1)}\beta^{\lambda(\omega,\xi_t(i))}\right).
\end{aligned}$$

Another, very ingenious, way of implementing Algorithms 5 and 6 so that every trial requires computation time $O(t)$ is described in Herbster and Warmuth (1997) (they only consider Algorithm 6, but their idea is general).

4.3. Algorithms with random ϵ

It is also possible to move one more level up and randomize the parameter ϵ of the density $\epsilon\alpha^{\epsilon-1}$ for $\alpha \in]0, 1[$. In Remark 7 we mentioned that there exists a distribution over ϵ such that the AA, when fed with this distribution, will satisfy inequality (17) at every trial T (provided $k < \frac{T}{10}$). In Section 5.8 we will see that we can take the distribution concentrated on the points $\epsilon = \frac{1}{\ln N}$, $N = 2, 3, \ldots$, with the probability of $\frac{1}{\ln N}$ being $\frac{1}{cN \ln^{1.01} N}$ (c is the normalizing constant). It might seem that, in this case, we need to consider the groups $E_t(i, \alpha, \epsilon)$ of elementary predictors, where for all $t \in \{1, 2, \ldots\}$, $i \in \{1, \ldots, n\}$, $\alpha \in]0, 1[$, and $\epsilon > 0$, $E_t(i, \alpha, \epsilon)$ consists of the elementary predictors whose current expert at trial t is i, whose switching rate is α, and whose switching rate was generated from the distribution with density $\epsilon\alpha^{\epsilon-1}$. In fact, the members of the groups $E_t(i, \alpha, \epsilon_1)$ and $E_t(i, \alpha, \epsilon_2)$, $\epsilon_1 \neq \epsilon_2$, behave identically and we do not need to distinguish between them. So we, as before, will consider the groups $E_t(i, \alpha)$ and their cumulative weights $w_t(i, \alpha)\, \mathrm{d}\alpha$.

Now we can give a description of the AA applied to the stochastic prediction strategy with random ϵ:

Algorithm 9. Learner

sets initial weights

$$w_0(i, \alpha) := \sum_{N=2}^{\infty} \epsilon_N \alpha^{\epsilon_N - 1} \frac{1}{N \ln^{1.01} N}, \quad i = 1, \ldots, n, \ \alpha \in]0, 1[,$$

where $\epsilon_N := \dfrac{1}{\ln N}$

AT EVERY TRIAL $t = 1, 2, \ldots$:

outputs the prediction

$$\gamma_t := \Sigma(g_t),$$

where $g_t : \Omega \to \mathbb{R}$ is the pseudoprediction

$$g_t(\omega) = \log_\beta \int_0^1 \sum_{i=1}^n \beta^{\lambda(\omega,\xi_t(i))} w_{t-1}(i,\alpha)\,\mathrm{d}\alpha$$

updates the weights in two steps:

$$w_t^*(i,\alpha) := w_{t-1}(i,\alpha)\beta^{\lambda(\omega,\xi_t(i))}, \quad i = 1,\ldots,n,\ \alpha \in]0,1[,$$

$$w_t(i,\alpha) := (1-\alpha)w_t^*(i,\alpha) + \frac{\alpha}{n-1}\sum_{j\neq i} w_t^*(j,\alpha),$$

$$i = 1,\ldots,n, \alpha \in]0,1[.$$

We can see that this algorithm is very similar to the algorithm for Strategy 4; the only difference is in the prior weights for the groups $E_t(i,\alpha)$. A similar effect could be achieved by choosing a density over α which is more concentrated around 0 as compared with any of the densities $\epsilon\alpha^{\epsilon-1}$. An example of such a density is

$$\frac{1}{c\alpha \ln\frac{1}{\alpha} \ln\ln\frac{1}{\alpha}\ldots},$$

where $c > 0$ is the normalizing constant and the product in the denominator contains all terms of the form $\ln\ldots\ln\frac{1}{\alpha} \geq 1$ (cf. Rissanen, 1983).

5. Proofs

5.1. Preliminary results

We will often use the following lemma.

Lemma 2. *For all $a > 0$ and $A > 0$,*

$$B(a,A) \geq \Gamma(a)(A + (a-1)^+)^{-a},$$

where we use the notation $t^+ := \max(t,0)$.

To prove it, we need the following simple result.

Lemma 3. *For any convex function f, any interval $[a,b] \subseteq \mathbb{R}$, and any $x, y \in [a,b]$,*

$$\frac{f(x)-f(a)}{x-a} \leq \frac{f(b)-f(y)}{b-y},$$

provided both denominators do not vanish.

Proof: The desired inequality immediately follows from

$$\frac{f(x)-f(a)}{x-a} \le \frac{f(b)-f(a)}{b-a} \le \frac{f(b)-f(y)}{b-y}. \qquad \square$$

Proof of Lemma 2: Recalling the representation

$$B(a,b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$$

of the beta function, we can see that we are required to prove

$$\frac{\Gamma(a)\Gamma(A)}{\Gamma(A+a)} \ge \Gamma(a)(A+(a-1)^+)^{-a},$$

i.e.,

$$\frac{\Gamma(A+a)}{\Gamma(A)} \le B^a,$$

where $B := A + (a-1)^+$. The last inequality is equivalent to

$$\frac{\ln\Gamma(A+a) - \ln\Gamma(A)}{a} \le \frac{\ln\Gamma(B+1) - \ln\Gamma(B)}{1}.$$

By Lemma 3, the last inequality is true when $B \ge A$ and $B+1 \ge A+a$, i.e., when $B \ge A+(a-1)^+$. (Recall that the function Γ is log-convex; (Carlson, 1977), Section 3.5.) □

Notice that, as $A \to \infty$,

$$\frac{B(a,A)}{\Gamma(a)(A+(a-1)^+)^{-a}} \sim \frac{\Gamma(A)}{\Gamma(A+a)} A^a \to 1$$

(see, e.g., Carlson (1977), Theorem 3.4-1); therefore, Lemma 2 is asymptotically tight.

5.2. *Proof of Theorem 1*

The probability that Stochastic Predictor who follows Strategy 2 will choose expert i and will choose a minimax action at the trials when the loss of expert i exceeds C is at least

$$\begin{aligned}\int_0^1 \frac{1}{n}\alpha^k(1-\alpha)^{T-k}\epsilon\alpha^{\epsilon-1}\mathrm{d}\alpha &= \frac{\epsilon}{n}\int_0^1 \alpha^{k+\epsilon-1}(1-\alpha)^{T-k}\mathrm{d}\alpha \\ &= \frac{\epsilon}{n}B(k+\epsilon, T-k+1) \ge \frac{\epsilon}{n}\Gamma(k+\epsilon)(T+1)^{-k-\epsilon}\end{aligned}$$

(the last inequality follows from Lemma 2). Substituting this value into Corollary 2, we obtain inequality (14).

5.3. *Proof of Theorem 2*

Let Stochastic Predictor follow Strategy 4. The probability that he will choose experts exactly following E is

$$\begin{aligned}\int_0^1 \frac{1}{n}(1-\alpha)^{T-1-k}\left(\frac{\alpha}{n-1}\right)^k \epsilon\alpha^{\epsilon-1}\,\mathrm{d}\alpha &= \frac{\epsilon}{n(n-1)^k}\int_0^1 \alpha^{k+\epsilon-1}(1-\alpha)^{T-1-k}\,\mathrm{d}\alpha \\ &= \frac{\epsilon}{n(n-1)^k}B(k+\epsilon, T-k) \\ &\geq \frac{\epsilon}{n(n-1)^k}\Gamma(k+\epsilon)T^{-k-\epsilon}, \qquad (25)\end{aligned}$$

the last inequality again following from Lemma 2. Applying Corollary 2, we obtain inequality (16).

5.4. *Proof of Theorem 3*

We will actually prove a stronger form of inequality (19): we will show that the term $\ln\frac{1}{\alpha}$ can be replaced with $\ln\frac{1-\beta+\alpha\beta}{\alpha}$. This proof will require a subtler analysis than the proof of Eq. (18): instead of Corollary 2 we will use directly Corollary 1. Remembering that $a(\beta) = c(\beta)/\ln\frac{1}{\beta}$, we can see that (19) follows from

$$\mathbf{E}\beta^{\mathrm{Loss}_T(\mathrm{SP})} \geq \beta^{\mathrm{Loss}_T(E)+k}\frac{1}{n}\left(\frac{\alpha}{n-1}\right)^k(1-\alpha)^{\mathrm{Loss}_T(E)}; \qquad (26)$$

we will actually prove

$$\mathbf{E}\beta^{\mathrm{Loss}_T(\mathrm{SP})} \geq c^k\beta^{\mathrm{Loss}_T(E)}\frac{1}{n}(1-\alpha)^{\mathrm{Loss}_T(E)}, \qquad (27)$$

where

$$c = \frac{\beta\alpha}{(n-1)(1-\beta+\alpha\beta)} \qquad (28)$$

(rather than $c = \frac{\beta\alpha}{n-1}$).

Recall that $E = e_1, \ldots, e_T$ and E contains k switches. We will consider the following two modifications of Strategy 5. In both modifications Stochastic Predictor chooses deterministically the first expert whose predictions he is going to repeat; under Modification R Stochastic Predictor chooses expert e_1 (the "right" expert), and under Modification W he chooses expert i, where $i \neq e_1$ (a "wrong" expert; because of the symmetry, the actual value of i does not matter). It is easy to see that (27) will follow from (the first of) the following two inequalities: under Modification R,

$$\mathbf{E}\beta^{\mathrm{Loss}_T(\mathrm{SPR})} \geq c^k\beta^{\mathrm{Loss}_T(E)}(1-\alpha)^{\mathrm{Loss}_T(E)} \qquad (29)$$

and, under Modification W,

$$\mathbf{E}\beta^{\mathrm{Loss}_T(\mathrm{SPW})} \geq c^{k+1}\beta^{\mathrm{Loss}_T(E)}(1-\alpha)^{\mathrm{Loss}_T(E)} \tag{30}$$

(we use the notation SPR and SPW to mean SP in the case where Modification R or Modification W is followed, respectively). We will prove these two inequalities by induction in T. For $T = 1$ they are true (in the case of (30), this follows from $\beta \geq c$), so we are only required to prove these inequalities assuming that $T \geq 2$ and that they hold with T replaced by $T - 1$.

Inequality (29) is easy to prove: if $e_1 = e_2$ it immediately follows from

$$\mathbf{E}\beta^{\mathrm{Loss}_{T-1}(\mathrm{SPR})} \geq c^k\beta^{\mathrm{Loss}_{T-1}(E)}(1-\alpha)^{\mathrm{Loss}_{T-1}(E)}$$

(where $\mathrm{Loss}_{T-1}(E)$ refers to the cumulative loss suffered by the sequence $e_2, \ldots, e_T$ of experts over trials $2, \ldots, T$ and $\mathrm{Loss}_{T-1}(\mathrm{SPR})$ refers to the cumulative loss suffered over trials $2, \ldots, T$ by Stochastic Predictor who follows Modification R *starting from trial* 2; analogous notation will be used for SPW), and if $e_1 \neq e_2$ it immediately follows from

$$\mathbf{E}\beta^{\mathrm{Loss}_{T-1}(\mathrm{SPW})} \geq c^k\beta^{\mathrm{Loss}_{T-1}(E)}(1-\alpha)^{\mathrm{Loss}_{T-1}(E)}$$

(notice that the number of switches in the sequence $e_2, \ldots, e_T$ is $k - 1$).

It remains to prove inequality (30). Let $i \neq e_1$ be the initial expert of Modification W followed by Stochastic Predictor. First we assume that $e_2 \neq i$ (this is the difficult case). In this case it suffices to prove

$$\begin{aligned}\beta^l\frac{1-(1-\alpha)^l}{n-1}\mathbf{E}\beta^{\mathrm{Loss}_{T-1}(\mathrm{SPR})} + \beta^l(1-\alpha)^l\mathbf{E}\beta^{\mathrm{Loss}_{T-1}(\mathrm{SPW})}\\ \geq c^{k+1}\beta^{\mathrm{Loss}_T(E)}(1-\alpha)^{\mathrm{Loss}_T(E)},\end{aligned} \tag{31}$$

where l is the loss of expert i at trial 1 (indeed, with probability $\frac{1-(1-\alpha)^l}{n-1}$ Stochastic Predictor will have e_2 as his current expert at trial 2 and $\frac{1-(1-\alpha)^l}{n-1} + (1-\alpha)^l \leq 1$). By the inductive assumption, (31) will follow from

$$\begin{aligned}\beta^l\frac{1-(1-\alpha)^l}{n-1}c^k\beta^{\mathrm{Loss}_{T-1}(E)}(1-\alpha)^{\mathrm{Loss}_{T-1}(E)}\\ +\beta^l(1-\alpha)^l c^{k+1}\beta^{\mathrm{Loss}_{T-1}(E)}(1-\alpha)^{\mathrm{Loss}_{T-1}(E)} \geq c^{k+1}\beta^{\mathrm{Loss}_T(E)}(1-\alpha)^{\mathrm{Loss}_T(E)},\end{aligned}$$

so it is enough to prove

$$\beta^l\frac{1-(1-\alpha)^l}{n-1} + \beta^l(1-\alpha)^l c \geq c,$$

i.e.,

$$c \leq \beta^l\frac{1-(1-\alpha)^l}{(n-1)(1-\beta^l(1-\alpha)^l)}.$$

Value (28) corresponds to $l = 1$, so it is sufficient to prove that the function

$$\beta^l \frac{1-(1-\alpha)^l}{1-\beta^l(1-\alpha)^l}$$

is decreasing in l, $0 < l < 1$. This is equivalent to proving that the function $\frac{a^l-1}{A^l-1}$ is decreasing in l, where a and A are constants satisfying $A > a > 1$. Differentiating in l, we can see that it is enough to prove that the function $\frac{x \ln x}{x-1}$ is increasing in x, $x > 1$; the latter fact can again be checked by differentiation.

In the case $e_2 = i$ inequality (30) will follow from

$$\begin{aligned}&\beta^l(1-\alpha)^l \mathbf{E}\beta^{\mathrm{Loss}_{T-1}(\mathrm{SPR})} + \beta^l\left(1-(1-\alpha)^l\right)\mathbf{E}\beta^{\mathrm{Loss}_{T-1}(\mathrm{SPW})}\\ &\quad\geq c^{k+1}\beta^{\mathrm{Loss}_T(E)}(1-\alpha)^{\mathrm{Loss}_T(E)},\end{aligned} \tag{32}$$

where again l is the loss of expert i at trial 1. By the inductive assumption, (32) will follow from

$$\begin{aligned}&\beta^l(1-\alpha)^l c^{k-1}\beta^{\mathrm{Loss}_{T-1}(E)}(1-\alpha)^{\mathrm{Loss}_{T-1}(E)} + \beta^l\left(1-(1-\alpha)^l\right)c^k\beta^{\mathrm{Loss}_{T-1}(E)}\\ &\quad\times(1-\alpha)^{\mathrm{Loss}_{T-1}(E)} \geq c^{k+1}\beta^{\mathrm{Loss}_T(E)}(1-\alpha)^{\mathrm{Loss}_T(E)}\end{aligned}$$

(remember that $e_2, \ldots, e_T$ has one switch less than $e_1, \ldots, e_T$), so it is enough to prove

$$\beta^l(1-\alpha)^l + c\beta^l\left(1-(1-\alpha)^l\right) \geq c^2.$$

We can assume $(1-\alpha)^l = 0$ (if $(1-\alpha)^l > 0$, it will only help us), obtaining

$$\beta^l \geq c;$$

this inequality follows from $\beta \geq c$.

5.5. *Proof of Theorem 4*

Recall that the density for α is $\epsilon\alpha^{\epsilon-1}$. Therefore, inequality (26) implies that when Stochastic Predictor follows Strategy 6,

$$\begin{aligned}\mathbf{E}\beta^{\mathrm{Loss}_T(\mathrm{SP})} &\geq \int_0^1 \beta^{\mathrm{Loss}_T(E)+k}\frac{1}{n}\left(\frac{\alpha}{n-1}\right)^k(1-\alpha)^{\mathrm{Loss}_T(E)}\epsilon\alpha^{\epsilon-1}\,\mathrm{d}\alpha\\ &= \beta^{\mathrm{Loss}_T(E)+k}\frac{\epsilon}{n(n-1)^k}\int_0^1 \alpha^{k+\epsilon-1}(1-\alpha)^{\mathrm{Loss}_T(E)}\,\mathrm{d}\alpha\\ &= \beta^{\mathrm{Loss}_T(E)+k}\frac{\epsilon}{n(n-1)^k}B(k+\epsilon, \mathrm{Loss}_T(E)+1)\\ &\geq \beta^{\mathrm{Loss}_T(E)+k}\frac{\epsilon}{n(n-1)^k}\Gamma(k+\epsilon)(\mathrm{Loss}_T(E)+k+1)^{-k-\epsilon}\end{aligned}$$

(the last inequality follows from Lemma 2). Application of Corollary 1 completes the proof of inequality (20).

5.6. *Proof of Theorem 5*

Now the density for α is $\epsilon(1-\alpha)^{\epsilon-1}$. Therefore, when Stochastic Predictor follows Strategy 7,

$$\begin{aligned}\mathbf{E}\beta^{\mathrm{Loss}_T(\mathrm{SP})} &\geq \int_0^1 \beta^{\mathrm{Loss}_T(E)+k}\frac{1}{n}\left(\frac{\alpha}{n-1}\right)^k (1-\alpha)^{\mathrm{Loss}_T(E)}\epsilon(1-\alpha)^{\epsilon-1}\,\mathrm{d}\alpha\\ &= \beta^{\mathrm{Loss}_T(E)+k}\frac{\epsilon}{n(n-1)^k}\int_0^1 \alpha^k(1-\alpha)^{\mathrm{Loss}_T(E)+\epsilon-1}\,\mathrm{d}\alpha\\ &= \beta^{\mathrm{Loss}_T(E)+k}\frac{\epsilon}{n(n-1)^k}B(k+1,\mathrm{Loss}_T(E)+\epsilon)\\ &\geq \beta^{\mathrm{Loss}_T(E)+k}\frac{\epsilon}{n(n-1)^k}\Gamma(\mathrm{Loss}_T(E)+\epsilon)(\mathrm{Loss}_T(E)+k+1)^{-\mathrm{Loss}_T(E)-\epsilon}\end{aligned}$$

(the last inequality follows from Lemma 2). Substituting this value into Corollary 1, we obtain inequality (21).

5.7. *Proof of Theorem 6*

The probability that Stochastic Predictor following Strategy 9 will choose the right moments to increase the degree of the polynomial is

$$\begin{aligned}\int_0^1 \alpha^k(1-\alpha)^{T-k-1}\epsilon\alpha^{\epsilon-1}\,\mathrm{d}\alpha &= \epsilon\int_0^1 \alpha^{k+\epsilon-1}(1-\alpha)^{T-k-1}\,\mathrm{d}\alpha\\ &= \epsilon B(k+\epsilon,T-k) \geq \epsilon\Gamma(k+\epsilon)T^{-k-\epsilon}.\end{aligned}$$

It remains to substitute this value into Corollary 2 and recall that $c(e^{-2})=1$ and $a(e^{-2})=\frac{1}{2}$.

5.8. *Proof of Remark 7*

We are only required to prove that, for a suitable distribution P over ϵ,

$$\int \frac{\epsilon}{n(n-1)^k}\Gamma(k+\epsilon)T^{-k-\epsilon}P(\mathrm{d}\epsilon) \geq^{\times} \frac{1}{n(n-1)^k}\left(\frac{T}{k}\right)^{-k}\left(\ln\frac{T}{k}\right)^{-2.01} e^{-1.01k}$$

(see (25) and (17)), where $R^{\times}$ means that the relation R (such as $\geq$, $\leq$, $=$) holds to within a constant factor; i.e., we are required to prove

$$\int \epsilon\Gamma(k+\epsilon)T^{-\epsilon}P(\mathrm{d}\epsilon) \geq^{\times} k^k \left(\ln \frac{T}{k}\right)^{-2.01} e^{-1.01k}.$$

By Stirling's formula, it is sufficient to show that

$$\int \epsilon \frac{\Gamma(k+\epsilon)}{\Gamma(k)} T^{-\epsilon}P(\mathrm{d}\epsilon) \geq^{\times} \left(\ln \frac{T}{k}\right)^{-2.01}.$$

By the log-convexity of the gamma function,

$$\frac{\Gamma(k+\epsilon)}{\Gamma(k)} \geq \left(\frac{\Gamma(k)}{\Gamma(k-1)}\right)^{\epsilon} = (k-1)^{\epsilon} \geq^{\times} k^{\epsilon}$$

(the last inequality is wrong for $k = 1$, but the inequality between the extreme terms of the whole chain is still true), so our task reduces to proving

$$\int \epsilon \left(\frac{T}{k}\right)^{-\epsilon} P(\mathrm{d}\epsilon) \geq^{\times} \left(\ln \frac{T}{k}\right)^{-2.01}. \tag{33}$$

Taking as P the distribution concentrated on the points $\epsilon = \frac{1}{\ln N}$, N ranging over the integers $2, 3, \ldots$, with

$$P\left\{\frac{1}{\ln N}\right\} = \frac{1}{cN \ln^{1.01} N}$$

(where $c := \sum_{N=2}^{\infty} \frac{1}{N \ln^{1.01} N}$ is the normalizing constant), we transform the left-hand side of (33) to

$$\sum_{N=2}^{\infty} \frac{1}{\ln N} \left(\frac{T}{k}\right)^{-\frac{1}{\ln N}} \frac{1}{cN \ln^{1.01} N} = \sum_{N=2}^{\infty} e^{-\frac{\ln(T/k)}{\ln N}} \frac{1}{cN \ln^{2.01} N}.$$

For the Ns of the order of magnitude T/k (their number has the order of magnitude T/k) the common term of this series has the order of magnitude

$$\frac{1}{(T/k)\ln^{2.01}(T/k)},$$

so the order of magnitude of the series is at least $\ln^{-2.01}(T/k)$. The proof is complete.

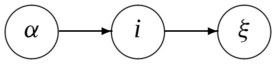
6. Further research

6.1. Other stochastic predictors

There are stochastic prediction strategies for the examples in Section 3 that are even more adaptive than those we considered. For example, an interesting possibility is to track the best value of the parameter α in inequality (22) (and analogous inequalities in other subsections): we described algorithms that learn the best constant value for α, but in some applications we might want to allow α to change slowly from trial to trial.

Of course, besides the examples of Section 3 there are many other examples where our methods might be useful, e.g.: tracking the best Bernoulli distribution (Freund, 1996); tracking the best portfolio of securities (Cover & Ordentlich, 1996; Vovk & Watkins, 1998); tracking the best pruning of a decision tree (Helmbold & Schapire, 1997; Takimoto, Maruoka, & Vovk, 1998).

Another possible direction of development of the ideas of this paper would be to consider "structured" pools of elementary predictors more systematically. (Recall that our "elementary predictors" correspond to the "experts" of the earlier papers; pools of experts have not been usually given any structure.) In this paper we considered only very simple, linear structures on the pool Θ of elementary predictors. For example, the structure corresponding to Strategy 2 looks like



(first Stochastic Predictor chooses α, then he chooses the expert i to emulate, after which he can easily compute his predictions); the trivial structure corresponding to Section 3.1 can be represented as $\textcircled{i}\to\textcircled{\xi}$. Much more sophisticated structures have been considered in the theory of Bayesian networks (see, e.g., Pearl, 1986; Lauritzen & Spiegelhalter, 1988). In the case of tree-like structures very efficient algorithms have been developed for weight updating, and it seems plausible that these ideas can also be used for efficient implementation of the AA.

6.2. Optimality

A natural question is whether the inequalities that we proved for the performance of the AA are optimal. So far this question has been studied only in the simplest situation of Section 3.1. In Vovk (1998) the following result is proven.

Theorem 7 (Vovk, 1998). *Let $\beta \in]0, 1[$. Suppose that $c \le c(\beta)$, $a \le a(\beta)$, and at least one of these two inequalities is strict. There does not exist Learner's strategy that would guarantee*

$$\mathrm{Loss}_T(\mathrm{Learner}) \le c\mathrm{Loss}_T(i) + a \ln n, \quad \forall i, T \tag{34}$$

(*cf.* (13)), *where*

$$\mathrm{Loss}_T(\mathrm{Learner}) := \sum_{t=1}^{T} \lambda(\omega_t, \gamma_t)$$

is Learner's total loss over the first T trials.

For some common loss functions (such as the quadratic or the Kullback—Leibler loss functions) there exist $\beta \in]0, 1[$ such that $c(\beta) = 1$ (see Haussler, Kivinen, & Warmuth (1994)); we will call such loss functions *perfectly mixable*. For a perfectly mixable loss function, an important parameter is

$$A := \inf_{\beta}\{a(\beta) \mid c(\beta) = 1\} \tag{35}$$

(this infimum is attained under our assumptions). Inequality (13) implies

$$\mathrm{Loss}_T(\mathrm{AA}) \le \mathrm{Loss}_T(i) + A \ln n, \quad \forall i, T. \tag{36}$$

Before Theorem 7 was proven, Haussler, Kivinen, & Warmuth (1994) had obtained the following important special case of it:

Corollary 3 (Haussler, Kivinen, & Warmuth, 1994). *Let λ be perfectly mixable, A be defined by* (35), *and $a < A$. There does not exist Learner's strategy that would guarantee*

$$\mathrm{Loss}_T(\mathrm{Learner}) \le \mathrm{Loss}_T(i) + a \ln n, \quad \forall i, T$$

(*cf.* (36)).

Unfortunately, Theorem 7 does not assert anything in the case where the number of experts is fixed. That theorem presupposes the following protocol: first an adversary chooses the number n of experts, and after that Learner must ensure that (34) holds; in the proof of Theorem 7 the number of experts is taken very large. Actually, Cesa-Bianchi et al. (1996) (see also (Vovk, 1998), Section 8) give a simple example with $n = 3$ where bound (13) can be improved. Chris Watkins (1997) has recently noticed that in the case of a perfectly mixable loss function the situation is different: inequality (36) is optimal for any fixed number of experts n.

Theorem 8 [Watkins, 1997]. *Let λ be perfectly mixable, A be defined by* (35), *and $a < A$. Let the pool size $n > 1$ be known to Learner in advance. There does not exist Learner's strategy that would guarantee*

$$\mathrm{Loss}_T(\mathrm{Learner}) \le \mathrm{Loss}_T(i) + a \ln n, \quad \forall i, T. \tag{37}$$

Proof (sketch): If there exists Learner's strategy $\mathcal{L}_1$ that guarantees (37) for $n = k$, then: there exists Learner's strategy $\mathcal{L}_2$ that guarantees (37) for $n = k^2$ (we can split the k^2 experts into k groups of k, merge the experts' predictions in every group with $\mathcal{L}_1$, and finally merge the groups' predictions with $\mathcal{L}_1$); there exists Learner's strategy $\mathcal{L}_3$ that guarantees (37) for $n = k^3$ (we can split the k^3 experts into k groups of k^2, merge the experts' predictions in every group with $\mathcal{L}_2$, and finally merge the groups' predictions with $\mathcal{L}_1$); and so on. Therefore, we can make the number of experts arbitrarily large, after which we can apply the proof given in (Vovk, 1998). □

This theorem comes very close to proving Haussler, Kivinen, and Warmuth's (1994) conjecture (see Eq. (3.29) of that paper).

It would be very interesting to obtain analogues of these optimality results for the theorems of Sections 3.2–3.4 (in various situations: loss functions can be assumed perfectly mixable or not, n large or fixed, etc.).

6.3. Computational efficiency

Finding computationally efficient implementations is perhaps the most important problem for the AA. One possible approach was discussed in Section 6.1: we can consider "structured" pools of elementary predictors. In this subsection we will briefly discuss a different approach.

The AA belongs to the "Bayesian" family of learning algorithms: instead of picking the best model in the light of the empirical data, it merges the predictions output by all possible models (i.e., elementary predictors), and learning consists in recomputing the weights for different models. The best-model algorithms (such as the maximum likelihood method, Vapnik's Structural Risk Minimization principle, Rissanen's Minimum Description Length principle, Wallace's Minimum Message Length principle, etc.) are often computationally more efficient. It seems that compromises between these two approaches (best-model and Bayesian) might be useful in practice. For example, in the situation of Section 3.4 and fixed α (see Strategy 8) a Bayesian algorithm would maintain the weights for all possible values $i = 0, 1, \dots$ of the degree of polynomial and a best-model algorithm would consider, at every trial, only one ("best", in accordance with some criterion) value of i; a possible middle ground is to consider, say, the 5 "best" values of i at every trial (e.g., the value of i that is assigned the biggest weight by the AA and the values $i - 2$, $i - 1$, $i + 1$, and $i + 2$). Analogously, for the algorithms with random α one could consider not all possible values $\alpha \in]0, 1[$ of the switching rate but a discrete subset of a small neighborhood of the α assigned the largest weight.

Acknowledgments

The main idea of this paper (comparing the learner's performance to the performance that the stochastic predictor can achieve with a significant probability) was suggested to me by Philip Dawid. I am grateful to the referees of the conference version of this paper (Vovk, 1997a) for their very helpful comments and to Chris Watkins for telling me about Theorem 8.

The comments of the referees of this journal version of the paper have greatly improved the presentation.

References

Auer, P., & Long, P. (1994). Simulating access to hidden information while learning. *Proceedings of the 26th Annual ACM Symposium on Theory of Computing* (pp. 263–272). New York: Assoc. Comput. Mach.

Carlson, B.C. (1977). *Special functions of applied mathematics*. New York: Academic Press.

Cesa-Bianchi, N., Freund, Y., Helmbold, D.P., Haussler, D., Schapire, R.E., & Warmuth, M.K. (1993). How to use expert advice. *Proceedings of the 25th Annual ACM Symposium on Theory of Computing* (pp. 382–391). New York: Assoc. Comput. Mach.

Cesa-Bianchi, N., Freund, Y., Helmbold, D.P., & Warmuth, M.K. (1996). On-line prediction and conversion strategies. *Machine Learning, 25*, 71–110.

Cesa-Bianchi, N., Helmbold, D.P., & Panizza, S. (1996). On Bayes methods for on-line Boolean prediction. *Proceedings of the 9th Annual ACM Conference on Computational Learning Theory* (pp. 314–324). New York: Assoc. Comput. Mach.

Cover, T., & Ordentlich, E. (1996). Universal portfolios with side information. *IEEE Trans. Inform. Theory, 42*, 348–363.

Dawid, A.P. (1986). Probability forecasting. In S. Kotz & N.L. Johnson (Eds.), *Encyclopedia of Statistical Sciences* (Vol. 7). New York: Wiley.

DeSantis, A., Markowsky, G., & Wegman, M.N. (1988). Learning probabilistic prediction functions. *Proceedings of the 29th Annual IEEE Symposium on Foundations of Computer Science* (pp. 110–119). Los Alamitos, CA: IEEE Comput. Soc.

Feder, M., Merhav, N., & Gutman, M. (1992). Universal prediction of individual sequences. *IEEE Trans. Inform. Theory, 38*, 1258–1270.

Freund, Y. (1996). Predicting a binary sequence almost as well as the optimal biased coin. *Proceedings of the 9th Annual ACM Conference on Computational Learning Theory* (pp. 89–98). New York: Assoc. Comput. Mach.

Freund, Y., Schapire, R., Singer, Y., & Warmuth, M. (1997). Using and combining predictors that specialize. *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*. New York: Assoc. Comput. Mach.

Haussler, D., Kivinen, J., & Warmuth, M.K. (1994). Tight worst-case loss bounds for predicting with expert advice. (Technical Report UCSC-CRL-94-36). University of California, Santa Cruz, CA, revised December 1994. Short version in P. Vitányi (Ed.), *Computational Learning Theory*. Lecture Notes in Computer Science (Vol. 904). Berlin: Springer (1995).

Helmbold, D., & Schapire, R. (1997). Predicting nearly as well as the best pruning of a decision tree. *Machine Learning, 27*, 51–68.

Herbster, M., & Warmuth, M. (1995). Tracking the best expert. *Proceedings of the 12th International Conference on Machine Learning* (pp. 286–294). Morgan Kaufmann. To appear in *Machine Learning*.

Herbster, M., & Warmuth, M. (1997). Tracking the best expert, II. Manuscript.

Lauritzen, S.L., & Spiegelhalter, D.J. (1988). Local computations with probabilities on graphical structures and their application to expert systems (with discussion). *J. R. Statist. Soc. B, 50*, 157–224. Also in (Shafer and Pearl, 1990).

Littlestone, N., & Warmuth, M.K. (1994). The weighted majority algorithm. *Inform. Computation, 108*, 212–261.

Pearl, J. (1986). Fusion, propagation, and structuring in belief networks. *Artificial Intelligence, 29*, 241–288. Also in (Shafer and Pearl, 1990).

Rissanen, J. (1983). A universal prior for integers and estimation by minimum description length. *Ann. Statist., 11*, 416–431.

Shafer, G., & Pearl, J. (Eds.) (1990). *Uncertain reasoning*. San Mateo, CA: Morgan Kauffman.

Takimoto, E., Maruoka, A., & Vovk, V. (1998). Predicting nearly as well as the best pruning of a decision tree through dynamic programming scheme. Submitted for publication.

Vovk, V. (1990). Aggregating strategies. *Proceedings of the 3rd Annual Workshop on Computational Learning Theory* (pp. 371–383). San Mateo, CA: Morgan Kaufmann.

Vovk, V. (1992). Universal forecasting algorithms. *Inform. Computation, 96*, 245–277.

Vovk, V. (1997a). Derandomizing stochastic prediction strategies. *Proceedings of the 9th Annual ACM Conference on Computational Learning Theory* (pp. 32–44). New York: Assoc. Comput. Mach.

Vovk, V. (1997b). On-line competitive linear regression. M.I. Jordan, M.J. Kearns, & S.A. Solla (Eds.), *Advances in Neural Information Processing Systems 10* (pp. 364–370). Cambridge, MA: MIT Press.

Vovk, V. (1998). A game of prediction with expert advice. *J. Comput. Inform. Syst., 56*, 153–173.

Vovk, V., & Watkins, C.J.H.C. (1998). Universal portfolio selection. *Proceedings of the 11th Annual ACM Conference on Computational Learning Theory* (pp. 12–23). New York: Assoc. Comput. Mach.

Watkins, C.J.H.C. (1997). How to use advice from small numbers of experts. (Technical Report CSD-TR-97-16) Department of Computer Science, Royal Holloway, University of London.

Yamanishi, K. (1995). Randomized approximate aggregating strategies and their applications to prediction and discrimination. *Proceedings of the 8th Annual ACM Conference on Computational Learning Theory* (pp. 83–90). New York: Assoc. Comput. Mach.

Received August 7, 1997
Accepted July 16, 1998
Final manuscript July 7, 1998