Prediction with Expert Advice for a Finite Number of Experts: A Practical Introduction

Yuri Kalnishkan

Department of Computer Science and Centre for Reliable Machine Learning, Royal Holloway, University of London, UK and Laboratory of Advanced Combinatorics and Network Applications, Moscow Institute of Physics and Technology, Dolgoprudny, Russia

Abstract

In this paper, prediction with expert advice is surveyed focusing on Vovk's Aggregating Algorithm. The established theory as well as extensions developed in the recent decade are considered. The paper is aimed at practitioners and covers important application scenarios.

Keywords: online learning, prediction, model selection

2020 MSC: 68Q32, 68T05

1. Overview

The problem of prediction with expert advice may be thought of as generalising the classical problem of merging of probabilistic hypotheses. A number of 'experts', which can be formulas, algorithms, or simply black boxes, are offering their predictions. Depending on the context, the predictions can come in the form of bits, probabilities, distributions, or even investment decisions. The problem is to merge them while making as few assumptions as possible about the nature of the data.

The quest for robustness unites the main topics of the special issue, conformal prediction and prediction with expert advice. One can claim that prediction with expert advice goes even further than conformal prediction in minimising the dependency on the laws governing the data: we aim to get worst case performance bounds. On the other hand, the assumptions on the data come in the form of the choice of the pool of experts.

This paper presents a tutorial introduction to key ideas and methods of prediction with expert advice aiming at practitioners in machine learning. While the comprehensive monograph [1] remains an important source on the topic, there have been new interesting and important developments in the area. On the other hand, the methods and results of prediction with expert advice remain

Email address: yuri.kalnishkan@rhul.ac.uk (Yuri Kalnishkan)

largely unknown to machine learning practitioners and an introduction like this may help to raise the awareness of this machine learning framework.

In Section 2, we describe the problem of sequential prediction and the framework of prediction with expert advice. Then, in Section 3, we present a number of specific prediction environments called *games*. Some of them are directly concerned with probabilistic prediction where the learner needs to work out a forecast in the form of a distribution; some generalise this framework.

The approach of this paper centres on Vovk's Aggregating Algorithm. In Section 4 we describe the algorithm taking Bayesian hypothesis merging as a starting point. We formulate the algorithm and its optimality properties. Then, in Sections 6–10, we discuss a number of prediction with expert advice ideas and approaches as extensions of the Aggregating Algorithm. Namely, we cover Fixed Share, specialist experts, discounted cumulative loss, and prediction of packs. All these techniques have clear practical relevance.

The results of prediction with expert advice are formulated for general, not necessarily mixable, loss functions. However, they become most interesting in the mixable case. The concept of mixability is briefly discussed in Section 5 with links for further reading.

Finally, in Section 11, we describe a number of applications where the methods of prediction with expert advice were used for practical prediction.

The ideas we survey are further developed in the papers of this special issue. The concepts of functional forecasts and integral losses from the paper "Mixability of integral losses: a key to efficient online aggregation of functional and probabilistic forecasts" by Alexander Korotin, Vladimir V'yugin, and Evgeny Burnaev are linked to the prediction of packs we describe in Section 10. The forecasts with confidence from "Online aggregation of probability forecasts with confidence" by Vladimir V'yugin and Vladimir Trunov extend the idea of a specialist expert from Section 8. The important applicaiton to the prediction of electricity consumption developed by V'yugin and Trunov features in Section 11.

Due to space considerations we excluded a more detailed discussion of non-mixable loss functions and algorithms for them such as Hedge, Weak Aggregating Algorithm etc. We have also excluded continuous pools of experts and universal algorithms concentrating on finite pools instead. Many results mentioned in this paper can be easily extended to continuous pools.

2. Setup and Goals

Suppose that a learner is tasked with predicting elements of a sequence $\omega_1, \omega_2, \ldots$ called *outcomes*. The outcomes occur in discrete time. Before seeing outcome ω_t, the learner is outputting a prediction γ_t. The quality of the prediction is measured by a loss function $\lambda(\cdot,\cdot)$. The learner aims to suffer low cumulative loss

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

over T steps.

We assume that the set of all possible outcomes (*outcome space*) Ω is known to us in advance and we are allowed to draw predictions from a known *prediction space* Γ, which may or may not be the same as Ω. The function λ is also known and maps $\Gamma \times \Omega$ to a subset of the extended real line, typically $[0, +\infty]$. The inclusion of $+\infty$ is necessary as this permits important special cases and streamlines some statements[1]. The choice of a triple $\mathfrak{G} = \langle \Omega, \Gamma, \lambda \rangle$, sometime referred to as a *game*, makes a lot of difference to prediction with expert advice. We do not normally assume any mechanism generating outcomes and look for worst case results holding for all possible sequences.

Suppose that the learner gets help from *experts*. The experts predict the same sequence and their predictions are made available to the learner before it commits to its own predictions. The experts are treated as black boxes. We are not concerned with their internal mechanics, which may well be inaccessible to us (e.g., the experts may rely on some sources of information unavailable or even unknown to us). The interaction with experts may be described by the following protocol. Here we assume that experts are parametrised by $\theta \in \Theta$.

Protocol 1.

```
FOR $t = 1, 2, \dots$
    experts $\mathcal{E}_\theta$, $\theta \in \Theta$, announce predictions $\gamma_t^\theta \in \Gamma$
    learner outputs $\gamma_t \in \Gamma$
    nature announces $\omega_t \in \Omega$
    each expert $\mathcal{E}_\theta$, $\theta \in \Theta$, suffers loss $\lambda(\gamma_t^\theta, \omega_t)$
    learner suffers loss $\lambda(\gamma_t, \omega_t)$
ENDFOR
```

Expert $\mathcal{E}_\theta$ suffers loss $\mathrm{Loss}_T(\mathcal{E}_\theta) = \sum_{t=1}^T \lambda(\gamma_t^\theta, \omega_t)$. The goal of the learner is to merge experts' predictions γ_t^θ into its own prediction γ_t in such a way that the learner's loss $\mathrm{Loss}_T(L)$ is low as compared to retrospectively best experts. It may use information about past outcomes and predictions. Formally, we are after a *merging strategy*

$$\mathcal{S} : (\Gamma^\Theta \times \Omega)^* \times \Gamma^\Theta \to \Gamma \ .$$

We typically want $\mathcal{S}$ to guarantee an upper bound on $\mathrm{Loss}_T(L)$ in terms of $\inf_{\theta \in \Theta} \mathrm{Loss}_T(\mathcal{E}_\theta)$; we want $\mathrm{Loss}_T(L)$ to be low whenever $\mathrm{Loss}_T(\mathcal{E}_\theta)$ is low for some θ. Ambitious it may sound, this goal is often achievable.

In this paper, we restrict ourselves to finite pools of experts, i.e., $|\Theta| = N < +\infty$.

Prediction with expert advice may be seen as an alternative to model selection. Picking a strategy from a pool Θ can be a difficult task. If one wants to choose an expert on the basis of its performance on some initial training segment, the well known *generalisation problem* occurs. While there are popular

[1] The value $-\infty$ is typically prohibited to avoid $-\infty + \infty$ ambiguity, but a few results will stand if the co-domain of λ is extended to $(-\infty, +\infty]$ or $[-C, +\infty]$ with real positive C.

methods, e.g., picking a strategy that strikes a balance between performance and complexity as expressed by BIC or AIC (see, e.g., Section 8.6 in [2], where the order of an ARIMA model is discussed), obtaining guarantees for their performance beyond the training segment is notoriously hard. Consistency results often require some unrealistic and fragile assumptions. Even the availability of the training segment may be questionable: if we refrain from prediction and use outcomes for training only, we may encounter opportunity costs. This may happen in the context of sequential investment, which we will discuss later.

By contrast, the guarantees of prediction with expert advice are very general, do not depend on statistical assumptions, and are easy to understand and interpret.

3. Some Games

We will now introduce some games to add substance to the abstract framework.

A *binary* game is concerned with the interval $[0,1]$. In a *discrete* binary game, the outcome space is $\{0,1\}$. If predictions are allowed from the interval $[0,1]$, the following loss functions may be considered. The *square* or *Brier* loss is given by $\lambda_{\mathrm{SQ}}(\gamma,\omega)=(\gamma-\omega)^2$, the *absolute* loss is $\lambda_{\mathrm{ABS}}(\gamma,\omega)=|\gamma-\omega|$, and the *logarithmic* loss is

$$\lambda_{\mathrm{LOG}}(\gamma,\omega)=\begin{cases}-\ln(1-\gamma) & \text{if } \omega=0 \ ;\\ -\ln\gamma & \text{if } \omega=1 \ .\end{cases}$$

These three losses define games with the same respective names.

The square and absolute loss games have their *continuous* counterparts with the outcome space $[0,1]$ and loss functions given by the same formulas.

In a *simple prediction game* the prediction and the outcome spaces are $\{0,1\}$ and the loss is given by

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

It is easy to see that the cumulative loss $\mathrm{Loss}_T(L)$ of a learner L in the simple prediction game is the number of mistakes it made on the steps $1,2,\ldots,T$.

The square and logarithmic games have important generalisations to the simplex

$$\Delta_{d-1}=\left\{(p_1,p_2,\ldots,p_d)\mid p_i\geq 0 \text{ and } \sum_{i=1}^d p_i=1\right\} \ .$$

Let predictions $\gamma=(\gamma^1,\gamma^2,\ldots,\gamma^d)$ be points from the simplex. In the discrete case, the outcomes are vertices of the simplex e_i ($e_i\in\mathbb{R}^d$ is a vector with one in position i and zeros everywhere else). The square loss can be calculated as the squared Euclidean norm $\lambda(\gamma,\omega)=\|\gamma-\omega\|^2$ and the logarithmic loss by

$\lambda(\gamma, e_i) = -\ln \gamma^i$. One may think of the logarithmic game on the simplex as soft classification assessed by the negative logarithm of the likelihood given to the true class.

In the continuous versions of these games, the outcomes $\omega = (\omega^1, \omega^2, \dots, \omega^d)$ come from the simplex Δ_{d-1} too. The square loss can be calculated by the same formula $\lambda(\gamma, \omega) = \|\gamma - \omega\|^2$ and the logarithmic loss is the KL-divergence $\lambda(\gamma, \omega) = \sum_{i=1}^d \omega^i \ln \frac{\omega^i}{\gamma^i}$.

An important generalisation of the logarithmic game is provided by the Cover's game, where the outcome space is $[0, +\infty)^d$, the prediction game is the simplex Δ_{d-1} and the loss is given by $\lambda(\gamma, \omega) = -\ln\langle\gamma, \omega\rangle$. This game has the following interpretation. Let $\gamma = (\gamma^1, \gamma^2, \dots, \gamma^d)$ represent the way we partition our capital between d assets and $\omega = (\omega^1, \omega^2, \dots, \omega^d)$ be the vector of price ratios showing by how much the asset prices change between this and the next discrete moment in time. Then the scalar product $\langle\gamma, \omega\rangle$ shows by how much our total capital changes. The negative logarithm represents this in the loss framework. Note that the values of the loss can be both positive and negative here; we can naturally suffer loss of $+\infty$ (and go bankrupt forever) but not $-\infty$.

This game is described in [3] (see also [4] for a more recent perspective) and it demonstrates that the protocol we consider is not restricted to prediction as such. Here γ represents a decision we make, ω the turn of events beyond our control, and $\lambda(\gamma, \omega)$ quantifies the consequences we face. The semantics of $\lambda(\gamma, \omega)$ does not have to be the discrepancy between γ and ω.

See [5] for a list of games with more examples including some important monstrosities.

4. The Aggregating Algorithm

In this section, we describe the Aggregating Algorithm (AA) introduced in [6, 5] and presenting a very general solution to the problem of prediction with expert advice.

As a motivation, consider the binary discrete logarithmic game. An expert $\mathcal{E}_i$ may be thought of as a hypothesis $\mathcal{H}_i$ and the prediction γ_t it outputs can be interpreted as a pair of probabilities under the hypothesis

$$\begin{aligned} \gamma_t^i &= \Pr(\omega_t = 1 \mid \mathcal{H}_i, \omega_1, \omega_2, \dots, \omega_{t-1}) \ ; \\ 1 - \gamma_t^i &= \Pr(\omega_t = 0 \mid \mathcal{H}_i, \omega_1, \omega_2, \dots, \omega_{t-1}) \ . \end{aligned}$$

In this framework, $\mathrm{Loss}_T(\mathcal{E}_i) = -\ln \Pr(\omega_1, \omega_2, \dots, \omega_T \mid \mathcal{H}_i)$. If we assign some prior probabilities to the hypotheses, $\Pr(\mathcal{H}_i) = q_i$, $i = 1, 2, \dots, N$, we can work out the probability $\Pr(\omega_t = 1 \mid \omega_1, \omega_2, \dots, \omega_{t-1})$ conditional on past observations $\omega_1, \omega_2, \dots, \omega_{t-1}$ using the law of total probability:

$$\Pr(\omega_t = 1 \mid \omega_1, \omega_2, \dots, \omega_{t-1}) =$$

$$\sum_{i=1}^{N} \Pr(\omega_t = 1 \mid \mathcal{H}_i, \omega_1, \omega_2, \ldots, \omega_{t-1}) \Pr(\mathcal{H}_i \mid \omega_1, \omega_2, \ldots, \omega_{t-1}) \ ,$$

where the posterior probability $\Pr(\mathcal{H}_i \mid \omega_1, \omega_2, \ldots, \omega_{t-1})$ is proportional, by the Bayes theorem, to $\Pr(\omega_1, \omega_2, \ldots, \omega_{t-1} \mid \mathcal{H}_i) \Pr(\mathcal{H}_i)$. Let the learner output $\gamma_t = \Pr(\omega_t = 1 \mid \omega_1, \omega_2, \ldots, \omega_{t-1})$. Then

$$e^{-\operatorname{Loss}_T(L)} = \Pr(\omega_1, \omega_2, \ldots, \omega_T) = \sum_{i=1}^{N} \Pr(\omega_1, \omega_2, \ldots, \omega_T \mid \mathcal{H}_i) \Pr(\mathcal{H}_i) = \sum_{i=1}^{N} q_i e^{-\operatorname{Loss}_T(\mathcal{E}_i)} \ . \quad (1)$$

By dropping from this sum all terms except for one, we get

$$e^{-\operatorname{Loss}_T(L)} \geq q_i e^{-\operatorname{Loss}_T(\mathcal{E}_i)}$$

and

$$\operatorname{Loss}_T(L) \leq \operatorname{Loss}_T(\mathcal{E}_i) + \ln(1/q_i) \ ,$$

which is a worst-case loss bound with no probabilistic terms in it.

This motivates the following approach. Let us maintain experts' weights $w_t^i = q_i e^{-\eta \operatorname{Loss}_t(\mathcal{E}_i)}$, where $\eta > 0$ is a parameter we will refer to as the *learning rate*. By normalising the weights on step t we can obtain $p_{t-1}^i = w_{t-1}^i / \sum_{j=1}^{N} w_{t-1}^j$. However, in a general game, the linear combination $\sum_{i=i}^{N} p_{t-1}^i \gamma_t^i$ does not necessarily have the desired properties and does not have to lead us to an analogue of (1) (and does not even have to exist; Γ does not have to be convex and it is not, for example, in the case of the simple prediction game).

This leads to the following definition. Consider a game $\mathfrak{G} = \langle \Omega, \Gamma, \lambda \rangle$. A constant $C > 0$ is *admissible* for a *learning rate* $\eta > 0$ if for every $N = 1, 2, \ldots$, every set of predictions $\gamma^1, \gamma^2, \ldots, \gamma^n \in \Gamma$, and every distribution $(p_1, p_2, \ldots, p_n) \in \Delta_{n-1}$, there is $\gamma \in \Gamma$ ensuring for all outcomes $\omega \in \Omega$ the inequality

$$\lambda(\gamma, \omega) \leq -\frac{C}{\eta} \ln \sum_{i=1}^{N} p_i e^{-\eta \lambda(\gamma^i, \omega)} \ . \quad (2)$$

The *mixability constant* C_η is the infimum of all $C > 0$ admissible for η. This infimum is usually achieved. For example, it is achieved for all $\eta > 0$ whenever Γ is compact and $e^{-\lambda(\gamma,\omega)}$ is continuous (or $\lambda(\gamma, \omega)$ is continuous w.r.t. the extended topology of $[0, +\infty]$) in γ.

Now we can formulate the Aggregating Algorithm. It takes as parameters a set of prior experts' weights $(q_1, q_2, \ldots, q_N) \in \Delta_{N-1}$, a learning rate $\eta > 0$ and an admissible $C > 0$. The algorithm works according to the following protocol.

Protocol 2 (AA).

```
1 initialise weights $w_0^i = q_i$, $i = 1, 2, \ldots, N$
```

```
2 FOR $t = 1, 2, \dots$
3     read the experts' predictions $\gamma_t^i$, $i = 1, 2, \dots, N$
4     normalise the weights $p_{t-1}^i = w_{t-1}^i / \sum_{j=1}^N w_{t-1}^j$
5     output $\gamma_t \in \Gamma$ satisfying for all $\omega \in \Omega$ the inequality
        $\lambda(\gamma_t, \omega) \leq -\frac{C}{\eta} \ln \sum_{i=1}^N p_{t-1}^i e^{-\eta\lambda(\gamma_t^i, \omega)}$
6     observe the outcome $\omega_t$
7     update the experts' weights $w_t^i = w_{t-1}^i e^{-\eta\lambda(\gamma_t^i, \omega_t)}$,
   $i = 1, 2, \dots, N$
8 END FOR
```

Since C is admissible, a suitable γ_t can always be found in line 5. For a particular game $\mathfrak{G}$, we do not usually need to solve a system of inequalities numerically and a simple explicit method called a *substitution rule* can usually be used.

By induction on time, one can show that

$$e^{-\eta \operatorname{Loss}_T(L)/C} \geq \sum_{i=1}^N q_i e^{-\eta \operatorname{Loss}_T(\mathcal{E}_i)} \quad . \tag{3}$$

Indeed,

$$\begin{aligned} e^{-\eta \operatorname{Loss}_{T+1}(L)/C} &= e^{-\eta \operatorname{Loss}_T(L)/C} e^{-\eta\lambda(\gamma_{T+1}, \omega_{T+1})/C} & (4) \\ &\geq \left(\sum_{i=1}^N q_i e^{-\eta \operatorname{Loss}_T(\mathcal{E}_i)} \right) \sum_{i=1}^N p_T^i e^{-\eta\lambda(\gamma_{T+1}^i, \omega)} & (5) \end{aligned}$$

using (3) as the inductive hypothesis and the inequality in line 5 of the AA. According to lines 3 and 7 of the AA,

$$p_T^i = \frac{w_T^i}{\sum_{j=1}^N w_T^j} = \frac{e^{-\eta \operatorname{Loss}_T(\mathcal{E}_i)}}{\sum_{j=1}^N q_j e^{-\eta \operatorname{Loss}_T(\mathcal{E}_j)}} \quad ;$$

substituting this into (5) yields the desired inequality.

Since the sum of nonnegative terms is greater or equal to each of the terms, we obtain

$$\operatorname{Loss}_T(L) \leq C \operatorname{Loss}_T(\mathcal{E}_i) + \frac{C}{\eta} \ln \frac{1}{q_i} \quad . \tag{6}$$

This bound holds for all times T no matter what outcomes occurred and what predictions experts made along the way. If equal initial weights $q_1 = q_2 = \ldots = q_N = 1/N$ are used, bound (6) turns into

$$\operatorname{Loss}_T(L) \leq C \operatorname{Loss}_T(\mathcal{E}_i) + \frac{C}{\eta} \ln N \quad . \tag{7}$$

The importance of the Aggregating Algorithm follows from the optimality results of [5]. Under some mild regularity assumptions on the game and assuming the uniform initial distribution, it can be shown that the constants in inequality (7) are optimal. If any merging strategy achieves the guarantee (with some $C, A > 0$)

$$\mathrm{Loss}_T(\mathcal{S}) \leq C\,\mathrm{Loss}_T(\mathcal{E}_i) + A \ln N$$

for all experts $\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_N$, $N = 1, 2, \ldots$, all time horizons T, and all outcomes, then the AA with the uniform prior distribution $q_i = 1/N$ and some $\eta > 0$ provides the guarantee with the same or lower C and A.

In other words, bounds (7) cannot be improved *within their class*. It is important to understand that entirely different bounds are possible. For example, as we will explain soon, the absolute loss game is not mixable, i.e., $C_\eta > 1$ for every $\eta > 1$. One can still achieve $C = 1$ in the bound, but at the cost of an additive term growing in time (for the absolute loss, $O(\sqrt{T})$ is possible) with different algorithms (e.g., the Weak Aggregating Algorithm from [7]). If cumulative loss grows linearly with time, having an additive term of order $o(T)$ would be preferable to having $C > 1$ for large values of T.

5. Mixability for Various Games

Working out mixability constants C_η is an important question. If $C < 1$ is ever admissible, we can iteratively improve the predictions bringing potential losses to 0. For a sensible game this should not be possible and thus $C_\eta \geq 1$. The most interesting case is $C_\eta = 1$. If this holds, the game is called η-mixable. If a game is η-mixable for some η, it is called *mixable*.

5.1. Mixability and Convexity

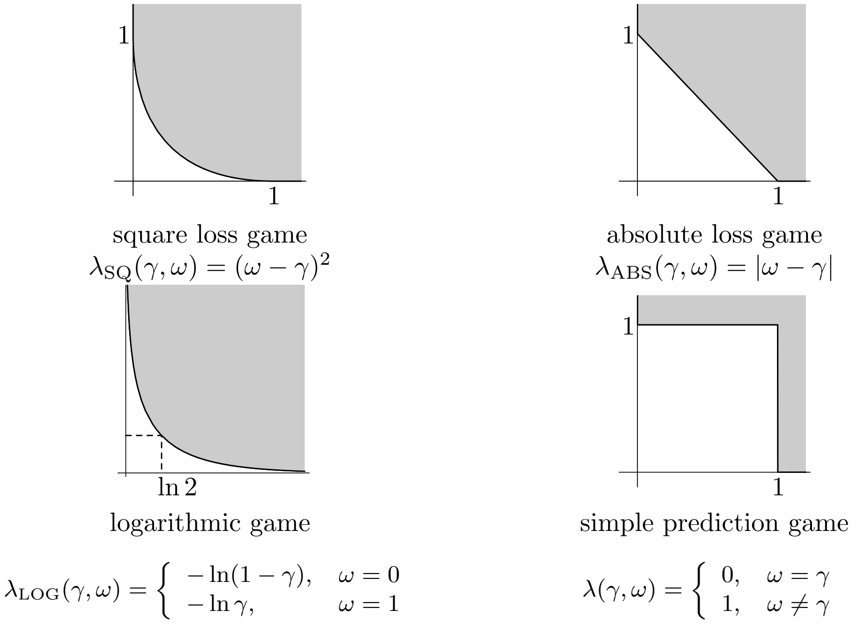
One can formulate a criterion of mixability as follows. Every game $\langle \Omega, \Gamma, \lambda \rangle$ defines $\mathcal{L} = \{\lambda(\gamma, \cdot) \mid \gamma \in \Gamma\}$, a subset of $[-\infty, +\infty]^\Omega$, which is the set of functions $\Omega \to [-\infty, +\infty]$. For a binary game this can be interpreted as a curve on the extended Euclidean plane, $\mathcal{L} = \{(\lambda(\gamma, 0), \lambda(\gamma, 1)) \mid \gamma \in \Gamma\} \subseteq [-\infty, +\infty]^2$.

Inequality (2) we used to define admissible C is equivalent to

$$e^{-\eta\lambda(\gamma,\omega)/C} \geq \sum_{i=1}^{N} p_i e^{-\eta\lambda(\gamma^i,\omega)} \quad . \tag{8}$$

Let us define the transformation $\mathfrak{B}_\eta : [-\infty, +\infty]^\Omega \to [-\infty, +\infty]^\Omega$ as follows: every $f(\cdot)$ is mapped into $e^{-\eta f(\cdot)}$. The system of inequalities (8) with $C = 1$ can always be resolved w.r.t. γ if and only if any convex combination of $\mathfrak{B}_\eta(\ell_1), \mathfrak{B}_\eta(\ell_2), \ldots, \mathfrak{B}_\eta(\ell_N)$, where $\ell_1, \ell_2, \ldots, \ell_N \in \mathcal{L}$, is majorised by $\mathfrak{B}_\eta(\ell)$ for some $\ell \in \mathcal{L}$.

This is easy to connect to a standard notion of convexity. Let us call an element $f \in [-\infty, +\infty]^\Omega$ a *superprediction* if there is $\gamma \in \Gamma$ leading to losses that are uniformly better or the same as those of f, i.e., $f(\omega) \geq \lambda(\gamma, \omega)$ for all $\omega \in \Omega$. In the binary game example, this is the set of points situated north-east of the curve; see Figure 1 for sets of superpredictions for binary games. Let us denote the set of superpredictions $\mathcal{S}$. If $\lambda(\gamma, \omega) > -\infty$, the image of $\mathcal{S}$ under $\mathfrak{B}_\eta$ does not contain infinite points and falls inside $\mathbb{R}^\Omega$. One can see that the



square loss game
$\lambda_{\mathrm{SQ}}(\gamma,\omega) = (\omega-\gamma)^2$

absolute loss game
$\lambda_{\mathrm{ABS}}(\gamma,\omega) = |\omega-\gamma|$

logarithmic game

$$\lambda_{\mathrm{LOG}}(\gamma,\omega) = \begin{cases} -\ln(1-\gamma), & \omega = 0 \\ -\ln\gamma, & \omega = 1 \end{cases}$$

simple prediction game

$$\lambda(\gamma,\omega) = \begin{cases} 0, & \omega = \gamma \\ 1, & \omega \neq \gamma \end{cases}$$

Figure 1: Sets of superpredictions for binary games

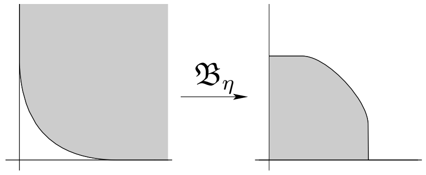


Figure 2: Image of the set of superpredictions under the transformation $\mathfrak{B}_\eta$.

game is η-mixable if and only if $\mathfrak{B}_\eta(\mathcal{S})$ is convex. This is illustrated in Figure 2 (the plot is actually for the square loss game with a large η).

5.2. Mixability of Binary Games

In the binary case, mixability can be investigated using standard calculus tools applied to the boundary of $\mathfrak{B}_\eta(\mathcal{S})$. Suppose that the boundary of $\mathcal{S} \subseteq [0,+\infty]^2$ is parameterised by $(x(u), y(u))$, with u ranging over an open interval $I \subseteq \mathbb{R}$, in the following sense: $\mathcal{S}$ equals the closure (w.r.t. the extended topology) of the points in $\mathbb{R}^2$ situated to the north-east of some $(x(u), y(u))$. If x and y are continuous on I and twice differentiable on the interior of I so that $x'(u) > 0$ and $y'(u) < 0$, then the game is mixable if and only if the fraction

$$\frac{y''(u)x'(u) - x''(u)y'(u)}{x'(u)y'(u)(y'(u) - x'(u))} \tag{9}$$

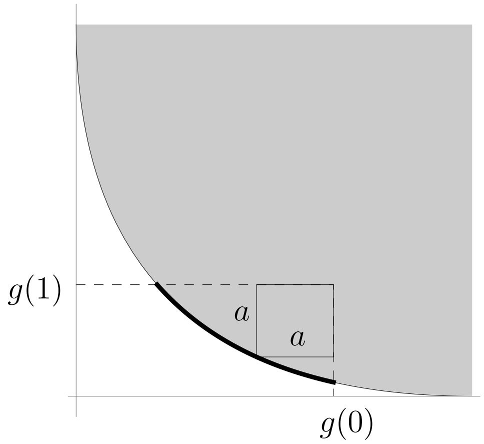


Figure 3: A substitution rule for the square loss game.

is positive and separated from 0 on the interior of I. If Γ is compact and λ is continuous in γ, the infimum of this fraction is the largest η such that $C_\eta = 1$; this η is clearly most practical and should be used in prediction. (This formula was obtained in [8]. In [9, 10] extensions for the non-smooth case are discussed.)

By working out the minimum of (9), one can check that the logarithmic game is mixable with the largest $\eta = 1$ (as to be expected from the probabilistic argument we had) and the binary square loss game with $\eta = 2$.

The concept of mixability is closely related to the curvature of the boundary of $\mathcal{S}$. In [11] a general mixability criterion for finite Ω in terms of the Hessian of the boundary of $\mathcal{S} \subseteq [0, +\infty]^{|\Omega|}$ is proposed.

5.3. Substitution Rules

How can we work out γ solving (2) in a mixable case? For the logarithmic game the question is easy. Indeed, for $\eta = 1$, the expression $e^{-\eta\lambda(\gamma,\omega)}$ amounts to γ or $1-\gamma$ and one can take the convex combination $\gamma = \sum_{i=1}^N p_i\gamma^i$ to satisfy (2).

For the square loss game the situation is a bit trickier. Take $\eta = 2$ and let $g(\omega) = -(1/\eta)\ln\sum_{i=1} p_i e^{-\eta\lambda(\gamma^i,\omega)}$. This is a function from Ω to $[-\infty, +\infty]$ and in [3] it is referred to as a *generalised prediction*. For a binary game, it can be identified with a point $(g(0), g(1))$. The discussion of mixability for the binary square loss game implies that for $\eta = 2$ it always falls north-east of $\mathcal{L}$. Since $(\gamma - \omega)^2 \leq 1$, the point is within $[0, 1]^2$.

We need to replace the generalised prediction by a value $\gamma \in [0, 1]$ such that $\gamma^2 \leq g(0)$ and $(1 - \gamma)^2 \leq g(1)$. Any choice of such γ would be acceptable on step (5) of the Aggregating Algorithm and would guarantee bounds (6) and (7) with $C = 1$; the analysis of the cumulative loss in this paper does not distinguish between them. Figure 3 shows the arc of $\mathcal{L}$ corresponding to all values of $\gamma \in [0, 1]$ suitable to replace $(g(0), g(1))$.

Still one needs to pick up one value for practical purposes. In [12], it is suggested to take the prediction leading to losses equidistant from $g(0)$ and

$g(1)$, respectively. One can find γ from the system

$$\begin{aligned} \gamma^2 + a &= g(0) \\ (1-\gamma)^2 + a &= g(1) \end{aligned}$$

(see Figure 3). Solving this system yields

$$\gamma = \frac{1}{2} - \frac{g(1) - g(0)}{2} \quad . \tag{10}$$

This value is easy to calculate; in [12] it leads to a simple explicit form of an algorithm.

Remark 1. The substitution rule we have worked out for the square loss game is quite different from the simple convex combination $\gamma = \sum_{i=1}^{N} p_i \gamma^i$. If the convex combination always exists (i.e., the prediction space Γ is convex), can it work as a substitution rule? Clearly, the convex combination will satisfy (8) with $C = 1$ if and only if $e^{-\eta\lambda(\gamma,\omega)}$ is concave in γ (this property is called *exponential concavity* in Section 3.3 of [1]). When does it hold for the square loss? We have

$$\frac{\partial^2}{\partial\gamma^2} e^{-\eta\lambda(\gamma,\omega)} = 2\eta(2\eta(\gamma-\omega)^2 - 1)e^{-\eta(\gamma-\omega)^2} \quad .$$

Since the loss $(\gamma-\omega)^2$ can be as high as 1, the inequality $\frac{\partial^2}{\partial\gamma^2} e^{-\eta\lambda(\gamma,\omega)} \le 0$ holds uniformly if and only if $\eta \le 1/2$. Taking $\eta = 1/2$ instead of $\eta = 2$ gives us suboptimal bounds (6) and (7). Still this merging strategy is technically possible; it is called Exponentially Weighted Average Forecaster in [1], Section 3.3.

5.4. Continuous Games

The admissible constants for a continuous game cannot be better than the constants for the corresponding discrete game. Let us show that the continuous square loss game is η-mixable for $\eta = 2$ and, moreover, the same substitution rules (including (10)) can be used for the continuous square loss game. Following [12], we will prove that if $\gamma_0 \in [0,1]$ satisfies

$$(\gamma_0 - \omega)^2 \le -\frac{1}{2} \ln \sum_{i=1}^{N} p_i e^{-2(\gamma^i - \omega)^2}$$

for $\omega = 0$ and $\omega = 1$, then the same γ_0 satisfies the inequality for every $\omega \in [0,1]$ by showing that

$$f(\omega) = (\gamma_0 - \omega)^2 + \frac{1}{2} \ln \sum_{i=1}^{N} p_i e^{-2(\gamma^i - \omega)^2}$$

is convex in ω. Letting

$$\Sigma = \sum_{i=1}^{N} p_i e^{-2(\gamma^i - \omega)^2} \quad ,$$

$$\Sigma' = \sum_{i=1}^{N} p_i (\gamma^i - \omega) e^{-2(\gamma^i - \omega)^2} \quad ,$$

$$\Sigma'' = \sum_{i=1}^{N} p_i (\gamma^i - \omega)^2 e^{-2(\gamma^i - \omega)^2}$$

we can write

$$\frac{d^2 f(\omega)}{d\omega^2} = 2\left(1 + \frac{1}{\Sigma^2}\left[-\Sigma^2 + 4\Sigma''\Sigma - 4\left(\Sigma'\right)^2\right]\right) = \frac{8}{\Sigma^2}\left(\Sigma''\Sigma - \left(\Sigma'\right)^2\right) \quad .$$

One can consider Σ' as a scalar product of the vectors $(\gamma^1 - \omega, \gamma^2 - \omega, \ldots, \gamma^N - \omega)$ and $(1, 1, \ldots, 1)$. Then the Cauchy(-Bunyakovsky-Schwarz) inequality implies $\left(\Sigma'\right)^2 \leq \Sigma''\Sigma$ and $\frac{d^2 f(\omega)}{d\omega^2} \geq 0$.

The square loss and logarithmic games on a simplex are discussed in [13] and [12], respectively.

The mixability of Cover's game for $\eta = 1$ follows from the linearity of $e^{-\lambda(\gamma,\omega)} = \langle \gamma, \omega \rangle$ in γ. It is not mixable for lower values of η because its special case, logarithmic game, is not. The Aggregating Algorithm for Cover's game has an interesting financial interpretation.

Suppose that the experts are suggesting some investment decisions to us. Let us partition the initial capital of 1 between the experts giving q_i to expert $\mathcal{E}_i$ and let each expert invest its capital and re-invest the proceeds according to its own advice. We do not redistribute the money between the experts; a good expert earns more money and controls a larger sum by design. Our wealth after step T then amounts to $W_T = \sum_{i=1}^{N} q_i W_T^i$, where W_T^i is how much we would have earned entrusting *all* our capital to expert $\mathcal{E}_i$ from the start. Clearly, we have $W_T \geq q_i W_T^i$ for $i = 1, 2, \ldots, N$.

If the investment happens as in Cover's game, $W_T^i = q_i \prod_{t=1}^{T} \langle \gamma_t, \omega_t \rangle = q_i e^{-\operatorname{Loss}_T(\mathcal{E}_i)}$. This equals the weight w_T^i assigned by the Aggregating Algorithm to expert $\mathcal{E}_i$. The normalised weight p_{t-1}^i can be thought of as the fraction of total capital after step $t-1$ controlled by expert $\mathcal{E}_i$. The vector $\gamma_t = \sum_{i=1}^{N} p_{t-1}^i \gamma_t^i$ is just the combined investment decision of the experts. The Aggregating Algorithm is thus equivalent to partitioning the money between the experts and letting them reinvest.

5.5. Non-mixable Games

Since (9) equals zero for the absolute loss game, it is not mixable for any $\eta > 0$. As can be checked directly, the simple prediction game is not mixable either.

The following geometric interpretation of admissible constants can be given when $\lambda(\gamma, \omega) \geq 0$. Let $\mathcal{H}(X)$ denote the convex hull of a set X. If a game is not η-mixable, $\mathcal{H}(\mathfrak{B}_\eta(S)) \neq \mathfrak{B}_\eta(S)$ and the same applies to their inverse images: $\mathfrak{B}_\eta^{-1}(\mathcal{H}(\mathfrak{B}_\eta(S))) \neq S$. The value of C is admissible if and only if $C \cdot \mathfrak{B}_\eta^{-1}(\mathcal{H}(\mathfrak{B}_\eta(S))) \subseteq S$, where the set $C \cdot X$ is obtained from X by multiplying each of its elements by C componentwise.

In [5], the values C_η are given for the absolute loss and simple prediction game. The values C_η for the former game can get arbitrarily close to 1 but never reaches it. The values of C_η for the later game make (7) identical to the bound of Corollary 2.1 from [14]. The Aggregating Algorithm thus reduces to Weighted Majority. The values C_η for the simple prediction game can get arbitrarily close to 2 but never reach 2.

6. Experts and Quantiles

In this section, we will give a few comments on the loss bounds (6) and (7). Obvious these comments may be, they are important for applications.

While the bounds are theoretically optimal in a strong sense, they do not necessarily convert to good practical performance. The rest of this paper is devoted to considerations that may lead to better quality predictions and lower losses. In this section, we discuss some simple aspects of the loss bounds.

The extra terms in (6) and (7) do not depend on time. It is reasonable to expect the cumulative loss to grow and possibly at a linear rate (say, outliers of the same magnitude in the data occurring at a fixed rate will make the loss grow linearly). This implies that in the mixable case for large T the extra term should be negligible and the loss of the learner should be close to the loss of the best expert. The dependency of the extra term on N, the number of experts, is mild. Assuming the uniform initial distribution, the dependency is logarithmic. This implies that one should not normally hesitate to include more experts: with time the algorithm will work out if they are needed at a relatively small cost.

In a practical situation, the extra term may present a problem. For example, in [4] the extra term overwhelms the advantages of the best expert. Thus a closer look may be justified.

Suppose that the absolute best expert outperforms all competitors but the logarithmic extra term eats up its advantages because each expert has a very low initial probability. However, let there be a substantial fraction of pretty good experts. Say, the best 25% of experts perform well. Can they help? Assume $C = 1$ and recall (3):

$$e^{-\eta \operatorname{Loss}_T(L)} \geq \sum_{i=1}^{N} q_i e^{-\eta \operatorname{Loss}_T(\mathcal{E}_i)} \quad .$$

Suppose that over time T experts with the combined initial weight of q or more suffer loss $\operatorname{Loss}_t(\mathcal{E}_i) \leq A$. Then

$$\sum_{i=1}^{N} q_i e^{-\eta \operatorname{Loss}_T(\mathcal{E}_i)} \geq q e^{-\eta A}$$

and

$$\operatorname{Loss}_T(L) \leq A + \frac{1}{\eta} \ln \frac{1}{q} \quad .$$

Thus a large quantile q may help. Note that no modifications of the AA were needed and the behaviour ensued automatically.

A closely related observation can be made following [15]. Suppose that we have two identical experts in the pool. It appears desirable to collate them into one. However, this is again done by the AA automatically. The behaviour of the AA would be the same as if one expert with the combined weight is present in the pool. Assuming the uniform distribution on the initial experts, the weight of the combined expert will be $2/N$ and the loss bound for the duplicated expert $\mathcal{E}_i$ (again assuming the mixable case $C = 1$) turns into

$$\mathrm{Loss}_T(L) \leq \mathrm{Loss}_T(\mathcal{E}_i) + \frac{1}{\eta}(\ln N - \ln 2) \ .$$

If this expert is actually good, this is to our advantage.

However, if duplicate experts are bad, they create a problem: needlessly increasing N worsens the bound for good experts.

7. Fixed Share

In this section we will discuss an important Fixed Share algorithm. It was introduced in [16] for the absolute loss game, but we will describe it in the context of the Aggregating Algorithm after [17].

The Aggregating Algorithm allows one to compete against individual experts. In the mixable case, the learner can perform nearly as well as the best expert, but there are situations when this is just not good enough.

As a motivating example, consider two experts predicting some variables describing the state of the economy, such as the inflation rate, unemployment, or the central bank base interest rate. Suppose that one expert is working well in years of economic growth and the other expert in years of slump and depression. If we consider the aggregate performance of each experts over, say, 50 years, they will look mediocre so predicting as well as either of them is not something worth aspiring to.

Instead we want to switch from one expert to another sometimes (in the example above, whenever the economic climate changes). This problem came to be known as *tracking the best expert* (after [16]). If the right moments for switching are easily predictable, the problem trivialises and we can manually create new experts performing according to the context. However, suppose that these points in time are hard to foresee.

Suppose that we have N experts (we will call them *base experts*). Let us consider *superexperts* that are built from base experts. Each superexpert on every step predicts as one of the base experts according to some switching pattern such as one shown in Table 1.

We can then mix all superexperts using the Aggregating Algorithm.

One cannot reasonably hope to perform as well as every superexpert (consider, for example, one that switches from one base expert to another on every step), but we can try and compete with some of them. We will give higher prior weights to those with fewer switches.

Table 1: A table showing a switching pattern.

time	1	2	3	...	T
expert followed	$\mathcal{E}_{n_1}$	$\mathcal{E}_{n_2}$	$\mathcal{E}_{n_3}$	...	$\mathcal{E}_{n_T}$

Let us introduce a distribution on the switching patterns favouring those with fewer switches. The distribution is controlled by a parameter $\alpha \geq 0$ called the *switching rate*. This parameter controls the frequency of switches, namely, α is the probability that a superexpert "decides" to make a switch to a different base expert at time t. The expert to switch to is chosen uniformly among the other experts. Suppose that a superexpert follows base expert $\mathcal{E}_i$ at time t. The probability it switches to $\mathcal{E}_j$ with $j \neq i$ on step $t+1$ is thus $\frac{\alpha}{N-1}$.

Assuming the uniform distribution of the initial expert $\mathcal{E}_{n_1}$, we thus assign to a switching pattern probability

$$\frac{1}{N}\left(\frac{\alpha}{N-1}\right)^k (1-\alpha)^{T-1-k} \quad , \tag{11}$$

where k is the number of switches. If we merge all superexperts S with these prior weights using the AA, inequality (6) yields the loss bound

$$\mathrm{Loss}_T(L) \leq \\ C\,\mathrm{Loss}_T(S) + \frac{C}{\eta}\left(\ln N + k \ln\frac{N-1}{\alpha} + (T-1-k)\ln\frac{1}{1-\alpha}\right) \tag{12}$$

for every superexpert S, where k is the number of switches made by the superexpert.

Here α is a parameter we can choose. If we have reasons to believe that superexperts with k switches should perform particularly well, the extra term in (12) can be optimised for them by taking $\alpha = \frac{k}{T-1}$ (this motivates the name *switching rate*).

Now let us discuss practical implementation. Applying the AA to the superexperts directly is very inefficient: for N^T superexperts we need to maintain N^T weights (and also need to know T in advance). There is, however, a trick that makes the algorithm nearly as simple as the AA. Each superexpert on every step follows some base expert. Instead of the weights of the superexperts, let us maintain the weights of base experts. In other words, for each base expert we will maintain the combined weight of all superexperts that follow it now.

Let w_{t-1}^n be the weight of base expert $\mathcal{E}_n$ on step t, i.e., the sum of weights of all superexperts that follow $\mathcal{E}_n$ on step t. How does it change after step t? First note that all superexperts following E_n make the same prediction γ_t^n and then have their weight multiplied by the same coefficient $e^{-\eta\lambda(\gamma_t^n,\omega_t)}$. So we multiply w_t^n by $e^{-\eta\lambda(\gamma_t^n,\omega_t)}$:

$$\tilde{w}_t^n = w_{t-1}^n e^{-\eta\lambda(\gamma_t^n,\omega_t)} \quad .$$

Then we need to take switching into account. Of the superexperts following $\mathcal{E}_n$ on step t, a fraction α leave and switch away from $\mathcal{E}_n$ to other base experts. At the same time, of the experts that follow $\mathcal{E}_m$ with $m \neq n$ share $\alpha/(N-1)$ switch to the base expert $\mathcal{E}_n$. Hence we can write

$$w_t^n = (1-\alpha)\tilde{w}_t^n + \frac{\alpha}{N-1}\sum_{m\neq n}\tilde{w}_t^m \quad .$$

Protocol 2 can be adjusted by replacing line 7 with

Protocol 3 (FS).

```
7'    update the experts' weights, part 1:  $\tilde{w}_t^i = w_{t-1}^i e^{-\eta\lambda(\gamma_t^i,\omega_t)}$,
      $i = 1, 2, \ldots, N$
7''   update the experts' weights, part 2:
      $w_t^n = (1-\alpha)\tilde{w}_t^n + \frac{\alpha}{N-1}\sum_{m\neq n}\tilde{w}_t^m$,  $n = 1, 2, \ldots, N$
```

Fixed Share turns out to be a fundamental algorithm in many respects. In [18] it is shown that it optimises *adaptive regret*, i.e., the difference between the cumulative loss of a learner and the best (base) expert on an arbitrary time interval $[t_1, t_2]$.

In [19] an important generalisation of Fixed Share has been put forward covering the case where we have both unpredictable and known changes of the context: on every step the learner is receiving side information from a finite set (a *task*), which it can utilize. Also refer to [19] for an extensive up-to-date literature list.

8. Specialist Experts

In this section, we discuss specialist experts. Specialist experts were introduced in [20] and we will follow the approach of [21] and [22]

A *specialist expert* is an expert that may refrain from making a prediction. If a specialist expert is not making a prediction, we say that it *sleeps*. Otherwise we say that it is *awake*. This may cover a range of real-life situations. A prediction algorithm may see that its internal confidence is low and decide to skip a turn in order to re-train. Alternatively, an algorithm may simply break down (e.g., a regression algorithm may have its matrix very close to singular).

When we dealt with normal experts, we wanted to be as good as the best expert in terms of the cumulative loss, where the cumulative loss was the sum of losses over a period of time. For a specialist expert we can calculate the sum of losses over the steps when the expert was awake. It is natural to judge the learner by its cumulative loss over the same steps, i.e., the steps when that expert was awake.

A natural idea for handling sleeping experts is to assume that a sleeping expert "joins the crowd". Let us imagine that the sleeping expert sides with the learner and outputs the learner's prediction for that turn.

The Aggregating Algorithm guarantees (in the mixable case) that

$$\sum_{t=1}^{T} \lambda(\gamma_t, \omega_t) \leq \sum_{t=1}^{T} \lambda(\gamma_t^i, \omega_t) + \frac{1}{\eta} \ln \frac{1}{q_i} \quad .$$

If on some steps t the learner made the same predictions as the expert $\mathcal{E}_i$, i.e., $\gamma_t = \gamma_t^i$, then the corresponding terms in the sums on the left and on the right cancel out and we get sums over the times when the expert $\mathcal{E}_i$ was awake.

Note that our argument has been circular so far: we say that the sleeping expert outputs learner's prediction γ_t but then the learner works out γ_t on the basis of experts' predictions. Fortunately, this is easy to resolve. On line 5 in Protocol 2 one finds γ_t by solving a system of inequalities equivalent to

$$e^{-\eta\lambda(\gamma_t,\omega)/C} \geq \sum_{n=1}^{N} p_{t-1}^n e^{-\eta\lambda(\gamma_t^n,\omega)} \quad ,$$

where ω ranges over Ω. Let us equate the loss of the experts $\mathcal{E}_n$ that sleep on step T to $\lambda(\gamma_t, \omega_t)/C$:

$$e^{-\eta\lambda(\gamma_t,\omega)/C} \geq \sum_{n:E_n \text{ is awake}} p_{t-1}^n e^{-\eta\lambda(\gamma_t^n,\omega)} + \sum_{n:E_n \text{ sleeps}} p_{t-1}^n e^{-\eta\lambda(\gamma_t,\omega)/C} \quad .$$

We can then subtract the last sum from both the sides and get

$$e^{-\eta\lambda(\gamma_t,\omega)/C} \geq \frac{1}{Z_t} \sum_{n:E_n \text{ is awake}} p_{t-1}^n e^{-\eta\lambda(\gamma_t^n,\omega)} \quad ,$$

where

$$Z_t = \sum_{n:E_n \text{ is awake}} p_{t-1}^n \quad .$$

This is a system of inequalities on γ_t with no signs of a vicious circle.

Here is the pseudocode for the Aggregating Algorithm handling sleeping experts. The parameters are $\eta > 0$, an admissible $C > 0$, and an initial distribution $q_1, q_2, \ldots, q_N$.

Protocol 4 (AA for Specialist Experts).

```
initialise weights $w_0^n = q_n$, $n = 1, 2, \ldots, N$
FOR $t = 1, 2, \ldots$
    read the predictions $\gamma_t^n$ of awake experts
    normalise the weights of awake experts
    $p_{t-1}^n = w_{t-1}^n / \sum_{i:E_i \text{ is awake}} w_{t-1}^i$
    solve the system ($\omega \in \Omega$):
    $\lambda(\gamma, \omega) \leq -\frac{C}{\eta} \ln \sum_{n:E_n \text{ is awake}} p_t^n e^{-\eta\lambda(\gamma_t^n,\omega)}$
    w.r.t. $\gamma$ and output a solution $\gamma_t$
    observe the outcome $\omega_t$
    update the awake experts' weights $w_t^n = w_{t-1}^n e^{-\eta\lambda(\gamma_t^n,\omega)}$,
    update the sleeping experts' weights $w_t^n = w_{t-1}^n e^{-\eta\lambda(\gamma_t,\omega)/C(\eta)}$
END FOR
```

The learner following this algorithm achieves loss satisfying

$$\sum_{\substack{t=1,2,\ldots,T:\\ E_n \text{ is awake}\\ \text{on step } t}}^{T} \lambda(\gamma_t,\omega_t) \le C\cdot \sum_{\substack{t=1,2,\ldots,T:\\ E_n \text{ is awake}\\ \text{on step } t}}^{T} \lambda(\gamma_t^n,\omega_t) + \frac{C}{\eta}\ln\frac{1}{q_n} \enspace .$$

In [21], the concept of a sleeping expert is generalised to a *second-guessing expert*, which, instead of a prediction $\gamma_t^n \in \Gamma$, outputs a mapping $\gamma_t^n : \Gamma \to \Gamma$ and suffers loss $\lambda(\gamma_t^n(\gamma_t),\omega_t)$ depending on the learner's prediction γ_t (in this section we treated a sleeping expert as an identity mapping). Under mild regularity assumptions, we can handle second-guessing expert and get bounds similar to (6).

9. Discounted Loss

Discounting losses (and gains) with time is a common practice in on-line learning. In finance-related applications this can be motivated by inflation[2]. In reinforcement learning, discounting comes as standard and is essential to ensure convergence to an optimal policy. In this section, we will discuss (after [23]) how discounting can be introduced into the Aggregating Algorithm.

Suppose that we are given coefficients $\alpha_1, \alpha_2, \ldots \in (0,1]$. Let the cumulative discounted loss for a learner be given by

$$\widetilde{\mathrm{Loss}}_T(L) = \sum_{t=1}^{T} \lambda(\gamma_t,\omega_t)\left(\prod_{s=t}^{T-1}\alpha_s\right) = \alpha_{T-1}\widetilde{\mathrm{Loss}}_{T-1}(L) + \lambda(\gamma_T,\omega_T) \enspace ;$$

the discounted loss of an expert $\mathcal{E}_n$ is defined in the same way. If all α_i are equal, $\alpha_1 = \alpha_2 = \ldots = \alpha$, then $\lambda(\gamma_t,\omega_t)$ comes into the formula with the discounting coefficient α^{T-t}.

Let us change line 4 in the AA to work out weights according to $p_{t-1}^i \propto q_i e^{-\eta\alpha_{t-1}\widetilde{\mathrm{Loss}}_{t-1}(\mathcal{E}_i)}$. Then one can show by induction on time that

$$e^{-\eta\widetilde{\mathrm{Loss}}_T(L)/C} \ge \sum_{i=1}^{N} q_i e^{-\eta\widetilde{\mathrm{Loss}}_T(\mathcal{E}_i)} \enspace . \tag{13}$$

Indeed, raising (13) to the power $\alpha_T \in (0,1]$ and applying Jensen's inequality yields

$$e^{-\eta\alpha_T\widetilde{\mathrm{Loss}}_T(L)/C} \ge \left(\sum_{i=1}^{N} q_i e^{-\eta\widetilde{\mathrm{Loss}}_T(\mathcal{E}_i)}\right)^{\alpha_T} \ge \sum_{i=1}^{N} q_i e^{-\eta\alpha_T\widetilde{\mathrm{Loss}}_T(\mathcal{E}_i)} \enspace .$$

[2] The discounting we consider does not directly translate into inflation because of time direction.

On step $T+1$, the learner suffers loss satisfying

$$e^{-\eta\lambda(\gamma_{T+1},\omega_{T+1})/C} \geq \sum_{n=1}^{N} p_T^n e^{-\eta\lambda(\gamma_{T+1}^n,\omega_{T+1})} = \sum_{n=1}^{N} \frac{q_n e^{-\eta\alpha_T \widetilde{\mathrm{Loss}_{\mathcal{E}_n}}(T)}}{\sum_{j=1}^{N} q_j e^{-\eta\alpha_T \widetilde{\mathrm{Loss}_{\mathcal{E}_j}}(T)}} e^{-\eta\lambda(\gamma_{T+1}^n,\omega_{T+1})} .$$

Multiplying these inequalities proves (13) for time $T+1$.

By dropping from (13) all terms except for one, we get the bound

$$\widetilde{\mathrm{Loss}_T}(L) \leq C\widetilde{\mathrm{Loss}_T}(\mathcal{E}_i) + \frac{C}{\eta}\ln\frac{1}{q_i} . \tag{14}$$

The following modification of the AA achieves the required probabilities. It takes as parameters $\eta > 0$, an admissible $C > 0$, an initial distribution $q_1, q_2, \ldots, q_N$, and discounting factors $\alpha_1, \alpha_2, \ldots$

Protocol 5 (AA with discounting).

```
initialise weights $w_0^i = 1$, $i = 1, 2, \ldots, N$
FOR $t = 1, 2, \ldots$
    read the experts' predictions $\gamma_t^i$, $i = 1, 2, \ldots, N$
    normalise the weights $p_{t-1}^i = q_i (w_{t-1}^i)^{\alpha_{t-1}} / \sum_{j=1}^N q_i (w_{t-1}^j)^{\alpha_{t-1}}$,
      $i = 1, 2, \ldots, N$
    output $\gamma_t \in \Gamma$ satisfying for all $\omega \in \Omega$ the inequality
      $\lambda(\gamma_t, \omega) \leq -\frac{C}{\eta} \ln \sum_{i=1}^N p_{t-1}^i e^{-\eta\lambda(\gamma_t^i, \omega)}$
    observe the outcome $\omega_t$
    update the experts' weights $w_t^i = (w_{t-1}^i)^{\alpha_{t-1}} e^{-\eta\lambda(\gamma_t^i, \omega_t)}$,
  $i = 1, 2, \ldots, N$
END FOR
```

10. Prediction of Packs

Suppose that on step t we need to make more than one prediction. The learner makes K_t predictions $\gamma_{t,1}, \gamma_{t,2}, \ldots, \gamma_{t,K_t}$ on the basis of experts' predictions $\gamma_{t,1}^n, \gamma_{t,2}^n, \ldots, \gamma_{t,K_t}^n$ ($n = 1, 2, \ldots, N$) and then outcomes $\omega_{t,1}, \omega_{t,2}, \ldots, \omega_{t,K_t}$ occur. For example, the learner may need to predict results of K_t football matches happening on day t. The number K_t may vary with time. We will be speaking of K_t outcomes as of a pack of outcomes and of K_t predictions as of a pack of predictions.

This situation may be considered as a special case of the delayed feedback protocol of [24]. However, we will describe the approach of [25] here instead.

For a game $\mathfrak{G} = \langle \Omega, \Gamma, \lambda \rangle$ and a positive integer K consider the game $\mathfrak{G}^K$ with the outcome space Ω^K, prediction space Γ^K, and the loss function $\lambda^K((\gamma_1, \gamma_2, \ldots, \gamma_K), (\omega_1, \omega_2, \ldots, \omega_K)) = \sum_{k=1}^K \lambda(\gamma_k, \omega_k)/K$.

Let $C > 0$ be admissible for $\mathfrak{G}$ with a learning rate $\eta > 0$. Suppose that we have experts' predictions γ_k^n, $k = 1, 2, \ldots, K$, $n = 1, 2, \ldots, N$ and a distribution $p_1, p_2, \ldots, p_K$. For every $k = 1, 2, \ldots, K$ there is $\gamma_k \in \Gamma$ such that

$$e^{-\eta\lambda(\gamma_k,\omega_k)/C} \geq \sum_{n=1}^{N} p_n e^{-\eta\lambda(\gamma_k^n,\omega_k)}$$

for all $\omega_k \in \Omega$. By multiplying these inequalities over k and applying Hölder's inequality one can show that

$$e^{-\eta\sum_{k=1}^{K}\lambda(\gamma_k,\omega_k)/(KC)} \geq \sum_{n=1}^{N} p_n e^{-\eta\sum_{k=1}^{K}\lambda(\gamma_k^n,\omega_k)/K} \quad ,$$

i.e., C remains admissible for $\mathfrak{G}^K$ with the same learning rate η. Moreover, experts predictions may be merged for $\mathfrak{G}^K$ using the same substitution rule as for $\mathfrak{G}$; it is just applied componentwise.

It is shown in [25] that under some general assumptions the inverse is true and the admissible constants for $\mathfrak{G}^K$ are not better than for $\mathfrak{G}$.

This motivates the following extension of the Aggregating Algorithm to the case of packs. It takes as parameters prior experts' weights $q_1, q_2, \ldots, q_N$, a learning rate $\eta > 0$ and an admissible $C > 0$.

Protocol 6 (AA for packs).

```
initialise weights $w_0^i = q_i$, $i = 1, 2, \ldots, N$
FOR $t = 1, 2, \ldots$
    read the experts' predictions $\gamma_{t,k}^i$,
        $i = 1, 2, \ldots, N$, $k = 1, 2, \ldots, K_t$
    normalise the weights $p_{t-1}^i = w_{t-1}^i / \sum_{i=1}^N w_{t-1}^i$
    output $\gamma_{t,k} \in \Gamma$ satisfying for all $\omega \in \Omega$ the inequalities
        $\lambda(\gamma_{t,k}, \omega) \leq -\frac{C}{\eta} \ln \sum_{i=1}^N p_{t-1}^i e^{-\eta\lambda(\gamma_{t,k}^i,\omega)}$
        $k = 1, 2, \ldots, K_t$
    observe the outcomes $\omega_{t,1}, \omega_{t,2}, \ldots, \omega_{t,K_t}$
    update the experts' weights $w_t^i = w_{t-1}^i e^{-\eta\sum_{k=1}^{K_t}\lambda(\gamma_t^i,\omega_t)/K_t}$,
  $i = 1, 2, \ldots, N$
END FOR
```

The learner following this algorithm suffers loss satisfying

$$\sum_{t=1}^{T} \frac{\sum_{k=1}^{K_t}\lambda(\gamma_{t,k},\omega_{t,k})}{K_t} \leq C\sum_{t=1}^{T} \frac{\sum_{k=1}^{K_t}\lambda(\gamma_{t,k}^n,\omega_{t,k})}{K_t} + \frac{C}{\eta}\ln\frac{1}{q_n}$$

for every $n = 1, 2, \ldots, N$.

11. Experimental Results

In this section we survey some computational results for prediction with expert advice. One should note that prediction with expert advice has mainly developed as a theoretical area motivated by theoretical questions. The area has not been known to practitioners to a sufficient extent and hence practical studies are few.

In [13], important experiments were carried out with sports data. One can interpret the odds quoted by a bookmaker as a probabilistic prediction of the result of a sports match. Different bookmakers can be treated as experts and the availability of odds data through betting websites makes this a convenient playground for prediction with expert advice methods.

The experiments of [13] are followed upon in [25], where the matches happening on the same day are treated as packs rather than processed sequentially.

In [26], methods of prediction with expert advice are applied to electricity consumption. This introduced the methods of prediction with expert advice to the large area of demand forecasting. The algorithm used in the paper is the Exponentially Weighted Average Forecaster from Section 3.3 of [1]. It is discussed in Remark 1 in this paper. Its guarantee for the square loss is not as good as that of the Aggregating Algorithm, but it is not worse off by a lot.

In [27], an important application of specialist experts to prediction of electricity consumption is discussed. The paper generalises the notion of a specialist expert to an expert that can be partially awake. Apart from a prediction γ_t such an expert produces a confidence value $p_t \in [0, 1]$, which quantifies its confidence (a fully sleeping expert would output confidence of 0 and a fully awake expert confidence of 1). A result similar to that described in Section 8 can be proven for them and they turn out to be very helpful for predicting electricity consumption under changing conditions.

In [25], prediction with expert advice is applied to house pricing. Working out the price of a house by its description has long been a benchmark problem in statistics and machine learning. Now with the availability of records of house sales (such as the Ames housing dataset and official datasets of the London area) this problem can be formulated in the on-line framework. As house sales are often dated to a month, prediction of packs naturally applies here.

In [28], a new direction of applications is proposed. Methods of prediction with expert advice can be used for selecting the right scope of past information. In machine learning, a practitioner is often presented with the problem of selecting the right training data. The problem has a temporal and a spacial aspect. What time horizon do we take so as not to include outdated information? What range of similar examples do we include in the training set to make it relevant? The approach of prediction with expert advice is to train models on different regions of data and then merge them in on-line settings. The resulting algorithm should perform little worse than the one knowing the right scope. In [28] this method is used for the prediction of implied volatility of options and prediction of students' performance at tests on the GrockIt dataset.

Acknowledgements

I would like to thank Algolab for support through the *Online Machine Learning for Effective Hedging Strategies* project.

I am grateful to Elena Kareva, Raisa Dzhamtyrova, Alexey Chernov, and anonymous reviewers for useful suggestions.

Bibliography

[1] N. Cesa-Bianchi, G. Lugosi, Prediction, Learning, and Games, Cambridge University Press, 2006.

[2] R. J. Hyndman, G. Athanasopoulos, Forecasting: principles and practice, 2nd Edition, OTexts: Melbourne, Australia, 2018.

[3] V. Vovk, C. J. H. C. Watkins, Universal portfolio selection, in: Proceedings of the 11th Annual Conference on Computational Learning Theory, ACM Press, 1998, pp. 12–23.

[4] N. Al-Baghdadi, D. Lindsay, Y. Kalnishkan, S. Lindsay, Practical investment with the long-short game, in: Conformal and Probabilistic Prediction and Applications 2020, PMLR, 2020, pp. 209–228.

[5] V. Vovk, A game of prediction with expert advice, Journal of Computer and System Sciences 56 (1998) 153–173.

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

[7] Y. Kalnishkan, M. V. Vyugin, The weak aggregating algorithm and weak mixability, Journal of Computer and System Sciences 74 (8) (2008) 1228–1244.

[8] D. Haussler, J. Kivinen, M. K. Warmuth, Sequential prediction of individual sequences under general loss functions, IEEE Transactions on Information Theory 44 (5) (1998) 1906–1925.

[9] Y. Kalnishkan, M. V. Vyugin, Mixability and the existence of weak complexities, in: Computational Learning Theory, 15th Annual Conference, Proceedings, Vol. 2375 of Lecture Notes in Artificial Intelligence, Springer, 2002, pp. 105–120.

[10] Y. Kalnishkan, V. Vovk, M. V. Vyugin, A criterion for the existence of predictive complexity for binary games, in: Algorithmic Learning Theory, 15th International Conference, ALT 2004, Proceedings, Vol. 3244 of Lecture Notes in Computer Science, Springer, 2004, pp. 249–263.

[11] T. van Erven, M. D. Reid, R. C. Williamson, Mixability is Bayes risk curvature relative to log loss, Journal of Machine Learning Research 13 (2012) 1639–1663.

[12] V. Vovk, Competitive on-line statistics, International Statistical Review 69 (2) (2001) 213–248.

[13] V. Vovk, F. Zhdanov, Prediction with expert advice for the Brier game, Journal of Machine Learning Research 10 (2009) 2445–2471.

[14] N. Littlestone, M. K. Warmuth, The Weighted Majority Algorithm, Information and Computation 108 (1994) 212–261.

[15] Y. Freund, R. 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.

[16] M. Herbster, M. K. Warmuth, Tracking the best expert, Machine Learning 32 (1998) 151–178.

[17] V. Vovk, Derandomizing stochastic prediction strategies, Machine Learning 35 (3) (1999) 247–282.

[18] D. Adamskiy, W. M. Koolen, A. Chernov, V. Vovk, A closer look at adaptive regret, in: Algorithmic Learning Theory, Springer, 2012, pp. 290–304.

[19] M. Herbster, S. Pasteris, L. Tse, Online multitask learning with long-term memory, Advances in Neural Information Processing Systems 33 (2020) 17779–17791.

[20] Y. Freund, R. E. Schapire, Y. Singer, M. K. Warmuth, Using and combining predictors that specialize, in: Proceedings of STOC'97, ACM, 1997, pp. 334–343.

[21] A. Chernov, Y. Kalnishkan, F. Zhdanov, V. Vovk, Supermartingales in prediction with expert advice, Theoretical Computer Science 411 (29-30) (2010) 2647–2669.

[22] A. Chernov, V. Vovk, Prediction with expert evaluators' advice, in: Algorithmic Learning Theory, ALT 2009, Proceedings, Vol. 5809 of LNCS, Springer, 2009, pp. 8–22.

[23] A. Chernov, F. Zhdanov, Prediction with expert advice under discounted loss, in: Proccedings of ALT 2010, Vol. LNAI 6331, Springer, 2010, pp. 255–269, see also arXiv:1005.1918 [cs.LG].

[24] P. Joulani, A. Gyorgy, C. Szepesvári, Online learning under delayed feedback, in: Proceedings of the 30th International Conference on Machine Learning (ICML-13), 2013, pp. 1453–1461.

[25] D. Adamskiy, A. Bellotti, R. Dzhamtyrova, Y. Kalnishkan, Aggregating algorithm for prediction of packs, Machine Learning 108 (8) (2019) 1231–1260.

[26] M. Devaine, P. Gaillard, Y. Goude, G. Stoltz, Forecasting electricity consumption by aggregating specialized experts, Machine Learning 90 (2) (2013) 231–260.

[27] V. V'yugin, V. Trunov, Online aggregation of probability forecasts with confidence, Pattern Recognition 121 (2022).

[28] Y. Kalnishkan, D. Adamskiy, A. Chernov, T. Scarfe, Specialist experts for prediction with side information, in: 2015 IEEE International Conference on Data Mining Workshop (ICDMW), IEEE, 2015, pp. 1470–1477.