
Online prediction with expert advice

Jyrki Kivinen
Australian National University

<http://axiom.anu.edu.au/~kivinen>

Contents

1. **Online prediction**: introductory example, basic setting
2. **Classification with Weighted Majority**: introductory example with noise
3. **Regression with averaging**: continuous-valued version of Weighted Majority
4. **The Aggregating Algorithm**: Vovk's algorithm for general loss functions; tight upper bounds
5. **Tracking a sequence of experts**: the nonstationary setting

Introductory example

- Each morning you must predict whether it will rain or not.
- At the end of the day we observe whether rain actually did occur.
- If your prediction was incorrect, you incur one unit of loss.
- Repeat this for T days (say $T = 365$); try to minimise your **total loss**.
- You do not know anything about meteorology, but have lots of other sources, called **experts**, to help you.
- The experts are given to you, you are not responsible for training them etc.
- You expect to do well if at least some experts are good.

Let there be n experts denoted by \mathcal{E}_i , $i = 1, \dots, n$ (say $n = 1000$).

- Experts may or may not have hidden side information.
- Experts may or may not have dependencies between each other.

Example:

\mathcal{E}_1 : Canberra Times weather column
 \mathcal{E}_2 : New York Times weather column
 \mathcal{E}_3 : Australian Bureau of Meteorology web site
...
 \mathcal{E}_i : your own backprop net with 10 hidden nodes
 \mathcal{E}_{i+1} : your own backprop net with 100 hidden nodes
 \mathcal{E}_{i+2} : it rains if you washed your car yesterday
...
 \mathcal{E}_{n-3} : it rains if it rained yesterday
 \mathcal{E}_{n-2} : it never rains
 \mathcal{E}_{n-1} : it always rains
 \mathcal{E}_n : toss a coin

(Almost) realistic examples

Modifications of the basic online expert prediction algorithm have been successful on these problems on realistic simulation data.

Problem: Stock portfolio management

Experts: “Invest all your money in stock s ”

Problem: Equalisation (in signal processing)

Experts: Copies of LMS with different learning rates

Problem: Disk spin-down on laptop

Experts: “Spin down after s seconds of idle time”

Problem: Disk caching

Experts: RAND, FIFO, LIFO, LRU, MRU, ...

Notation

- $\hat{y}_t = 1$ if you predict rain for day t , $\hat{y}_t = 0$ otherwise
- similarly $x_{t,i}$ is prediction of expert \mathcal{E}_i for day t
- $y_t = 1$ if there actually was rain on day t , $y_t = 0$ otherwise
- define the *discrete* or *0-1 loss* as

$$L_{0-1}(y, \hat{y}) = \begin{cases} 0 & \text{if } y = \hat{y} \\ 1 & \text{otherwise} \end{cases}$$

- total losses of the Predictor (*i.e.* you) and expert \mathcal{E}_i are

$$\text{Loss}(P) = \sum_{t=1}^T L_{0-1}(y_t, \hat{y}_t) \quad \text{and} \quad \text{Loss}(\mathcal{E}_i) = \sum_{t=1}^T L_{0-1}(y_t, x_{t,i}),$$

respectively

Basic setting

The following is repeated at each day t :

1. You see the experts' predictions $x_{t,i}$, $i = 1, \dots, n$.
2. You make your prediction \hat{y}_t .
3. You see the actual outcome y_t .

When you make your prediction for day t , you remember what happened at days $1, \dots, n - 1$. In particular, you know the experts' past performances.

- Want to achieve small total loss for the predictor assuming at least *one* expert has small total loss.
- Start with the simplistic case where one expert has loss zero (but of course we do not in advance know which).

Algorithm (Weighted Majority for noise-free case)

(We say “noise-free” to denote that at least one expert is perfectly correct for the whole sequence.)

- Say that expert \mathcal{E}_i is *consistent* up to time t if $x_{\tau,i} = y_\tau$ for $\tau = 1, \dots, t - 1$.
- At time t , predict according to the majority of experts consistent up to that time. (Break ties arbitrarily.)

We are assuming that at least one expert is consistent for the whole sequence so this is well defined.

Proposition If $\text{Loss}(\mathcal{E}_i) = 0$ for at least one expert \mathcal{E}_i , then

$$\text{Loss}(P) \leq \log_2 n.$$

Proof

- Every time the predictor makes a mistake, at least half the remaining consistent experts also made a mistake.
- Hence, every time the predictor incurs a non-zero loss, the number of consistent experts is at least halved.
- Since initially there are n consistent experts and at the end at least one, this can happen at most $\log_2 n$ times.

□

We now move from this example to the more general case.

Basic sequence of the general prediction game

We have a prediction space \mathcal{X} and outcome space $\mathcal{Y} \subseteq \mathcal{X}$. Unless otherwise specified we consider here the case $\mathcal{X} = [0, 1]$ and $\mathcal{Y} = \{0, 1\}$ (continuous-valued predictions, discrete outcomes).

Repeat for $t = 1, \dots, T$:

1. For $i = 1, \dots, n$, expert \mathcal{E}_i makes its prediction $x_{t,i} \in \mathcal{X}$.
2. The predictor makes its prediction $\hat{y}_t \in \mathcal{X}$.
3. An outcome $y_t \in \mathcal{Y}$ is observed.

Can think of this as game where **Environment** picks $x_{t,i}$ and y_t and **Predictor** picks \hat{y}_t . Predictor would be our learning algorithm. No assumptions about how Environment works.

Loss function is a function $L: \mathcal{Y} \times \mathcal{X} \rightarrow [0, \infty]$.

For $\mathcal{X} = \{0, 1\}$ use 0-1 loss.

For $\mathcal{X} = [0, 1]$, possibilities include *square loss*

$$L(y, \hat{y}) = (y - \hat{y})^2,$$

logarithmic loss

$$L(y, \hat{y}) = y \ln \frac{y}{\hat{y}} + (1 - y) \ln \frac{1 - y}{1 - \hat{y}}$$

(where $0 \ln 0 = 0 \ln(1/0) = 0$ and $\ln(1/0) = \infty$ otherwise) and *absolute loss*

$$L(y, \hat{y}) = |y - \hat{y}|.$$

With fairly obvious modifications, square and absolute loss can be extended to other ranges than $[0, 1]$.

What we want: to bound $\text{Loss}(P)$ in terms of $\min_i \text{Loss}(\mathcal{E}_i)$ where

$$\text{Loss}(P) = \sum_{t=1}^T L(y_t, \hat{y}_t) \quad \text{and} \quad \text{Loss}(\mathcal{E}_i) = \sum_{t=1}^T L(y_t, \hat{y}_t).$$

The basic form of bounds we want is

$$\text{Loss}(P) \leq (1 + o(1)) \min_{1 \leq i \leq n} \text{Loss}(\mathcal{E}_i)$$

where $o(1)$ goes to zero as $\min_i \text{Loss}(\mathcal{E}_i)$ goes to infinity.

Suppose $\text{Loss}(\mathcal{E}_i) = T\sigma_i$ where σ_i is a “loss rate” for expert \mathcal{E}_i . Then this would give

$$\frac{\text{Loss}(P)}{T} \leq \min_{1 \leq i \leq n} \sigma_i + o(1)$$

where $o(1)$ goes to zero as T goes to infinity.

What our algorithm(s) will achieve (brief preview)

For square loss, log loss etc. we can guarantee

$$\text{Loss}(P) \leq \min_{1 \leq i \leq n} \text{Loss}(\mathcal{E}_i) + c \ln n.$$

For absolute loss can guarantee for any $0 < \varepsilon < 1$ (given a priori) that

$$\text{Loss}(P) \leq (1 + \varepsilon) \min_{1 \leq i \leq n} \text{Loss}(\mathcal{E}_i) + \frac{c}{\varepsilon} \ln n.$$

For discrete predictions can only guarantee

$$\text{Loss}(P) \leq (2 + \varepsilon) \min_{1 \leq i \leq n} \text{Loss}(\mathcal{E}_i) + \frac{c}{\varepsilon} \ln n.$$

Here $c > 0$ is a constant that depends only on the loss function.

The trade-off parameter ε is determined by the learning rate of the algorithm which (in the basic version of the algorithm) must be **fixed in advance**.

Plug in $\text{Loss}(\mathcal{E}_i) = T\sigma_i$.

For square loss etc. we get

$$\frac{\text{Loss}(P)}{T} \leq \min_{1 \leq i \leq n} \sigma_i + \frac{c \ln n}{T}.$$

For absolute loss, by choosing $\varepsilon = \sqrt{C \ln n / (\min_i \text{Loss}(\mathcal{E}_i))}$ and using $\sigma_i \leq 1$ we get

$$\frac{\text{Loss}(P)}{T} \leq \min_{1 \leq i \leq n} \sigma_i + 2\sqrt{\frac{c \ln n}{T}} + O\left(\frac{1}{T}\right).$$

In these cases, the loss rate of predictor converges to that of the best expert.
(The choice of ε is an issue though.)

Important: The dependence on n is always **only logarithmic**.

Weighted Majority Algorithm [Littlestone & Warmuth 1989]

Assume $y_t, x_{t,i} \in \{0, 1\}$; let $L = L_{0-1}$. Fix a “learning rate” $0 < \beta < 1$.

At time t , each expert \mathcal{E}_i has *weight* $w_{t,i} \geq 0$. Initialise $w_{1,i} = 1$ for $i = 1, \dots, n$.

At time t :

- Compute

$$W_t^+ = \sum_{i: x_{t,i}=1} w_{t,i} \quad \text{and} \quad W_t^- = \sum_{i: x_{t,i}=0} w_{t,i}.$$

Predict $\hat{y}_t = 1$ if $W_t^+ \geq W_t^-$ and $\hat{y}_t = 0$ otherwise.

- Update weights by setting $w_{t+1,i} = \beta w_{t,i}$ if $x_{t,i} \neq y_t$ and $w_{t+1,i} = w_{t,i}$ if $x_{t,i} = y_t$.

Remark: Defining $\eta = \ln(1/\beta) > 0$, we have

$$w_{t,i} = \exp \left(-\eta \sum_{\tau=1}^{t-1} L(y_\tau, x_{\tau,i}) \right).$$

Write $W_t = W_t^+ + W_t^- = \sum_{i=1}^n w_{t,i}$.

Lemma For $c = 1/\ln(2/(1 + \beta))$ we have

$$L(y_t, \hat{y}_t) \leq -c \ln \frac{W_{t+1}}{W_t}.$$

Proof Clear if $L(y_t, \hat{y}_t) = 0$ since $c > 0$ and $W_{t+1} \leq W_t$.

Consider the case $L(y_t, \hat{y}_t) = 1$, say $y_t = 1$ and $\hat{y}_t = 0$. Then $W_t^+ \leq W_t/2$. We have

$$\begin{aligned} W_{t+1} &= W_t^+ + \beta W_t^- \\ &= (1 - \beta)W_t^+ + \beta(W_t^+ + W_t^-) \\ &\leq (1 - \beta)\frac{W_t}{2} + \beta W_t \end{aligned}$$

from which $c \ln(W_t/W_{t+1}) \geq 1$ follows. □

Theorem For any expert \mathcal{E}_i we have

$$\text{Loss}(P) \leq c\eta \text{Loss}(\mathcal{E}_i) + c \ln n$$

where

$$c = \left(\ln \frac{2}{1 + \beta} \right)^{-1} \quad \text{and} \quad \eta = \ln \frac{1}{\beta}.$$

Proof: Pick any expert \mathcal{E}_i . From Claim 1 we have

$$\begin{aligned} \text{Loss}(P) &\leq -c \sum_{t=1}^T (\ln W_{t+1} - \ln W_t) \\ &= -c \ln W_{T+1} + c \ln W_1 \\ &\leq -c \ln w_{t+1,i} + c \ln W_1 \\ &= c\eta \sum_{t=1}^T L(y_t, x_{t,i}) + c \ln n. \end{aligned}$$

□

Interpretation of the bound (see plot on next slide)

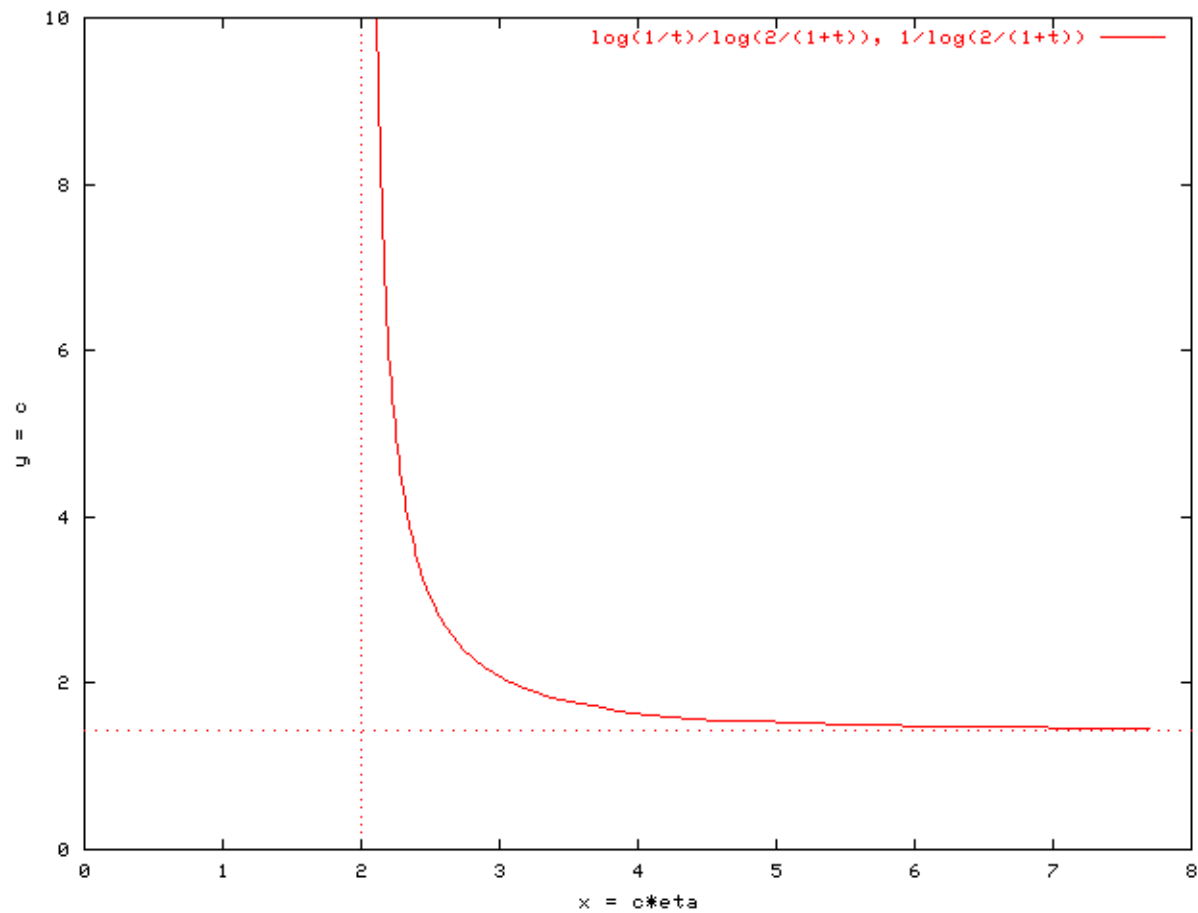
- very low learning rate:

$$\lim_{\beta \rightarrow 1^-} c\eta = 2 \quad \text{and} \quad \lim_{\beta \rightarrow 1^-} c = \infty$$

- very high learning rate:

$$\lim_{\beta \rightarrow 0^+} c\eta = \infty \quad \text{and} \quad \lim_{\beta \rightarrow 0^+} c = \frac{1}{\ln 2} \approx 1.44$$

- tuning β : if even the best expert is fairly bad, the first term will dominate, so want low learning rate to minimise its coefficient
- optimal tuning depends on characteristics of data that are typically not known at the beginning
- there are (fairly complicated) procedures for adjusting β during learning so that “almost optimal” performance is guaranteed without such knowledge



The curve of points $(x, y) = (c\eta, c)$ for $\beta \in (0, 1)$. The asymptotes are $x = 2$ and $y = 1/\ln 2$.

Lower bound: we cannot substantially improve upon this bound.

For simplicity assume $n = 2^{k+1}$ for some k . Fix M , and let $T = k + 2M$.

Fix a predicting algorithm P . Will construct a set of n experts and a sequence of T outcomes such that

1. the best expert makes at most M mistakes and
2. P makes T mistakes.

Thus there are cases where we have

$$\text{Loss}(P) = 2M + k \geq 2 \min_{1 \leq i \leq n} \text{Loss}(\mathcal{E}_i) + \frac{\ln n}{\ln 2} - 1.$$

Lower bound construction For $i = 1, \dots, 2^k$, define experts \mathcal{E}_i and \mathcal{E}_{i+2^k} as follows:

- For $t = 1, \dots, k$, both \mathcal{E}_i and \mathcal{E}_{i+2^k} predict 1 if the binary representation of $(i - 1)$ has 1 in position t .
- For $t = k + 1, \dots, k + 2M$, \mathcal{E}_i always predicts 1 and \mathcal{E}_{i+2^k} always predicts 0.

After first k time steps there is exactly one i such that neither \mathcal{E}_i nor \mathcal{E}_{i+2^k} has made any mistakes yet.

During the remaining time steps either \mathcal{E}_i or \mathcal{E}_{i+2^k} must be right at least half the time.

Thus for any sequence of outcomes there is an expert with at most M mistakes. We simply choose the outcomes to be opposite to whatever P predicts. \square

Example $k = 2$, $n = 2^{k+1} = 8$, $M = 3$; $T = k + 2M = 8$.

The experts' predictions are as follows:

t	1	2	3	4	5	6	7	8
\mathcal{E}_1	0	0	1	1	1	1	1	1
\mathcal{E}_2	1	0	1	1	1	1	1	1
\mathcal{E}_3	0	1	1	1	1	1	1	1
\mathcal{E}_4	1	1	1	1	1	1	1	1
\mathcal{E}_5	0	0	0	0	0	0	0	0
\mathcal{E}_6	1	0	0	0	0	0	0	0
\mathcal{E}_7	0	1	0	0	0	0	0	0
\mathcal{E}_8	1	1	0	0	0	0	0	0

Suppose the outcome sequence is

$$(y_1, \dots, y_8) = (1, 0, 1, 0, 0, 0, 0, 1).$$

After $k = 2$ steps \mathcal{E}_2 and \mathcal{E}_6 have loss 0.

After the whole sequence, \mathcal{E}_6 has loss $2 \leq M$ (and \mathcal{E}_2 has loss $4 = 2M - 2$).

Weighted Average Algorithm

Assume $0 \leq x_{t,i} \leq 1$ and $y_t \in \{0, 1\}$. Fix a learning rate $\eta > 0$.

At time t , each expert \mathcal{E}_i has *weight* $w_{t,i} \geq 0$. Initialise $w_{1,i} = 1$ for $i = 1, \dots, n$.

At time t :

- For $i = 1, \dots, n$ let $v_{t,i} = w_{t,i}/W_t$ where $W_t = \sum_{i=1}^n w_{t,i}$. Predict with

$$\hat{y}_t = \mathbf{v}_t \cdot \mathbf{x}_t = \sum_{i=1}^n v_{t,i} x_{t,i}.$$

- Update weights by setting

$$w_{t+1,i} = w_{t,i} \exp(-\eta L(y_t, x_{t,i})).$$

Remark: As with Weighted Majority we have

$$w_{t,i} = w_{1,i} \exp\left(-\eta \sum_{\tau=1}^{t-1} L(y_\tau, x_{\tau,i})\right).$$

The weights can have a [Bayesian interpretation](#) (although we are not doing Bayesian analysis here).

Suppose we have a parametric density $P(y | x)$ and a prior on experts $P(\mathcal{E}_i)$. Expert (*i.e.* model) \mathcal{E}_i would generate y_t according to the distribution $P(\cdot | x_{t,i})$.

Define a loss function $L(y, x) = -\frac{1}{\eta} \ln P(y | x)$. Now

$$\begin{aligned} P(\mathcal{E}_i | y_1, \dots, y_t) &\propto P(y_1, \dots, y_t | \mathcal{E}_i) P(\mathcal{E}_i) \\ &= P(\mathcal{E}_i) \prod_{\tau=1}^t P(y_\tau | x_{\tau,i}) \\ &= P(\mathcal{E}_i) \exp \left(-\eta \sum_{\tau=1}^t L(y_\tau, x_{\tau,i}) \right). \end{aligned}$$

Thus the update of Weighted Average Algorithm (with initial weights $w_{1,i} \propto P(\mathcal{E}_i)$) becomes Bayes' rule.

However not all natural loss functions can be represented as log likelihoods.

Analysis of the algorithm goes, at high level, as follows.

1. Find $\eta > 0$ and $c > 0$ such that we can guarantee

$$L(y_t, \hat{y}_t) \leq -c \ln \frac{W_{t+1}}{W_t}.$$

2. Sum over $t = 1, \dots, T$.

Step 2 is easy and independent of the loss function; we essentially already did it in the analysis of Weighted Majority.

Step 1 is technical and uses properties of the actual loss function. However general results exist that handle large classes of loss functions. The case $c = 1/\eta$ is of special interest.

Remark: In Step 2 we do not need the assumption $\hat{y}_t = \mathbf{v}_t \cdot \mathbf{x}_t$. In many cases Step 1 can actually be performed with better constants using some other prediction. This leads to Vovk's [Aggregating Algorithm](#).

Theorem Fix $\eta > 0$, and let $c > 0$ be such that

$$L(y_t, \hat{y}_t) \leq -c \ln \frac{W_{t+1}}{W_t}$$

holds for all t . Then for any expert \mathcal{E}_i we have

$$\text{Loss}(P) \leq c\eta \text{Loss}(\mathcal{E}_i) + c \ln n.$$

Proof: Pick any expert \mathcal{E}_i . We have

$$\begin{aligned} \sum_{t=1}^T L(y_t, \hat{y}_t) &\leq -c \sum_{t=1}^T (\ln W_{t+1} - \ln W_t) \\ &= -c \ln W_{T+1} + c \ln W_1 \\ &\leq -c \ln w_{T+1,i} + c \ln W_1 \\ &= c\eta \sum_{t=1}^T L(y_t, x_{t,i}) + c \ln n. \end{aligned}$$

□

Denote $L_y(x) = L(y, x)$; in particular $L'_y(x) = \partial L(y, x)/\partial x$. We assume that L is convex and twice differentiable, and that L_0 is increasing and L_1 decreasing.

Lemma Assume $c \geq \tilde{c}_L$ where

$$\tilde{c}_L = \max_{y \in \{0,1\}} \sup_{0 < x < 1} \frac{L'_y(x)^2}{L''_y(x)}.$$

Then for all $\eta \leq 1/c$ the Weighted Average Algorithm satisfies

$$L(y_t, \hat{y}_t) \leq -c \ln \frac{W_{t+1}}{W_t}.$$

Remarks

- By the previous remarks, for $\eta = 1/\tilde{c}_L$ we thus have

$$\text{Loss}(P) \leq \min_{1 \leq i \leq n} \text{Loss}(\mathcal{E}_i) + \tilde{c}_L \ln n.$$

- If we replace $\max_{y \in \{0,1\}}$ by $\sup_{0 < y < 1}$, the bound holds for $0 \leq y_t \leq 1$ (and not just $y_t \in \{0,1\}$); the value of \tilde{c}_L remains the same in most cases.

Examples (The last column gives the optimal constant that can be achieved with the “real” Aggregating Algorithm; more of this later.)

loss function	$L(y, x)$	\tilde{c}_L	c_L
square	$(y - x)^2$	2	1/2
logarithmic	$(1 - y) \ln((1 - y)/(1 - x)) + y \ln(y/x)$	1	1
Hellinger	$\frac{1}{2} \left((\sqrt{1 - y} - \sqrt{1 - x})^2 + (\sqrt{y} - \sqrt{x})^2 \right)$	1	$2^{-1/2} \approx 0.71$
absolute	$ y - x $	(∞)	(∞)

Remark The constant c_L can be shown to be optimal, *i.e.* for any predictor P there are sequences such that

$$\text{Loss}(P) \geq \min_{1 \leq i \leq n} \text{Loss}(\mathcal{E}_i) + c_L \ln n - o(1).$$

Proof of Lemma: Without loss of generality take $c = 1/\eta$. The claim is

$$L(y_t, \mathbf{v}_t \cdot \mathbf{x}_t) \leq -c \ln \frac{\sum_{i=1}^t w_{t,i} \exp(-L(y_t, x_{t,i})/c)}{W_t} = -c \ln \sum_{i=1}^n v_{t,i} \exp(-L(y_t, x_{t,i})/c),$$

or equivalently

$$\exp(-L(y_t, \mathbf{v}_t \cdot \mathbf{x}_t)/c) \geq \sum_{i=1}^n v_{t,i} \exp(-L(y_t, x_{t,i})/c).$$

Write this as

$$f\left(\sum_{i=1}^n v_{t,i} x_{t,i}\right) \geq \sum_{i=1}^n v_{t,i} f(x_{t,i})$$

where $f(x) = \exp(-L(y, x)/c)$. The assumption implies that f'' is negative so the claim follows by Jensen's Inequality. □

We have nothing yet for **absolute loss** $L(y, \hat{y}) = |y - \hat{y}|$.

- For $y \in \{0, 1\}$ and $0 \leq \hat{y} \leq 1$, can interpret $|y - \hat{y}|$ as probability of mistake when prediction \hat{y} taken as bias in predicting by a coin toss.

- For 0-1 loss (*i.e.* number of mistakes) can only get

$$\text{Loss}(P) \leq (2 + o(1)) \min_{1 \leq i \leq n} \text{Loss}(\mathcal{E}_i).$$

- For absolute loss we soon show bounds of the type

$$\text{Loss}(P) \leq (1 + o(1)) \min_{1 \leq i \leq n} \text{Loss}(\mathcal{E}_i).$$

- Thus can get the leading coefficient to 1 if we allow **randomised predictions**.

To get this type of bound we need Vovk's Aggregating Algorithm with its nontrivial prediction method.

Aggregating Algorithm

[Vovk 1989]

Assume $0 \leq x_{t,i} \leq 1$ and $y_t \in \{0, 1\}$. Fix a learning rate $\eta > 0$ and “fudge factor” $c > 0$. Initialise $w_{1,i} = 1$ for $i = 1, \dots, n$.

At time t :

- Define $W_t = \sum_{i=1}^n w_{t,i}$ and, for $y \in \{0, 1\}$, $W_{t+1}(y) = \sum_{i=1}^n w_{t,i} \exp(-\eta L(y, x_{t,i}))$.
- If there is $0 \leq z \leq 1$ such that for both $y = 0$ and $y = 1$ we have

$$L(y, z) \leq -c \ln \frac{W_{t+1}(y)}{W_t},$$

pick $\hat{y}_t = z$ for any such z . Otherwise the algorithm fails.

- Update weights by $w_{t+1,i} = w_{t,i} \exp(-\eta L(y_t, x_{t,i}))$.

This generalises our previous algorithms by allowing more freedom in choice of \hat{y}_t .

- As before we have

$$w_{t,i} = w_{1,i} \exp \left(-\eta \sum_{\tau=1}^{t-1} L(y_{\tau}, x_{\tau,i}) \right).$$

- Since $W_{t+1} = W_{t+1}(y_t)$, it now follows **by definition** that

$$L(y_t, \hat{y}_t) \leq -c \ln \frac{W_{t+1}}{W_t}.$$

The problem is choosing (c, η) so that the algorithm never fails.

- If this can be achieved then as before we get

$$\text{Loss}(P) \leq c\eta \min_{1 \leq i \leq n} \text{Loss}(\mathcal{E}_i) + c \ln n.$$

- Although the prediction rule looks rather abstract as written it usually becomes rather simple when c and η are given. Also given η there is a unique best value for c , depending only on which loss function we use.

Example of the prediction rule: square loss.

- First compute

$$\Delta(y) = -c \ln \frac{W_{t+1}(y)}{W_t} = -c \sum_{i=1}^N \frac{w_{t,i}}{W_t} \exp(-\eta(y - x_{t,i})^2)$$

for $y = 0$ and $y = 1$.

- Since $L(0, \hat{y}) = \hat{y}^2$ and $L(1, \hat{y}) = (1 - \hat{y})^2$, we now require $\hat{y}_t^2 \leq \Delta(0)$ and $(1 - \hat{y}_t)^2 \leq \Delta(1)$. which becomes

$$1 - \sqrt{\Delta(1)} \leq \hat{y}_t \leq \sqrt{\Delta(0)}.$$

- Thus we can take e.g.

$$\hat{y}_t = \frac{1}{2} \left(\sqrt{\Delta(0)} - \sqrt{\Delta(1)} + 1 \right)$$

assuming $\sqrt{\Delta(0)} + \sqrt{\Delta(1)} \geq 1$. As it turns out this is the case in particular when $\eta = 1/c \leq 1/2$.

Denote $L_y(x) = L(y, x)$; in particular $L'_y(x) = \partial L(y, x)/\partial x$. We assume that L is convex and twice differentiable, and that L_0 is increasing and L_1 decreasing.

Lemma Assume $c \geq c_L$ where

$$c_L = \sup_{0 < x < 1} \frac{L'_0(x)L'_1(x)^2 - L'_1(x)L'_0(x)^2}{L'_0(x)L''_1(x) - L'_1(x)L''_0(x)}.$$

Then the Aggregating Algorithm for any $\eta \leq 1/c$ will not fail.

Remarks:

- For examples of c_L and respective values of the constant \tilde{c}_L for the Weighted Average Algorithm see the earlier table.
- We get $c_L \leq \tilde{c}_L$ directly from the fact that $(a + b)/(a' + b') \leq \max\{a/a', b/b'\}$ for positive a, a', b, b' .

Absolute loss Pick any $\eta > 0$; write $\beta = e^{-\eta}$.

- Can show that the algorithm never fails if $c \geq 1/(2 \ln(2/(1 + \beta)))$ so

$$\text{Loss}(P) \leq \frac{\ln \frac{1}{\beta}}{2 \ln \frac{2}{1+\beta}} \min_{1 \leq i \leq n} \text{Loss}(\mathcal{E}_i) + \frac{1}{2 \ln \frac{2}{1+\beta}} \ln n,$$

i.e. factor 2 better than Weighed Majority (**but** this is for a different loss).

- Write $\sigma_i = \text{Loss}(\mathcal{E}_i)/T$. **Optimal choice of β** guarantees

$$\frac{\text{Loss}(P)}{T} \leq \min_{1 \leq i \leq n} \left(\sigma_i + \sqrt{\frac{\sigma_i \ln n}{T}} + \frac{\ln n}{2T \ln 2} \right)$$

but choosing optimal β requires knowing $\min_i \text{Loss}(\mathcal{E}_i)$.

- Bounds of the form $\text{Loss}(P)/T \leq \min_i \sigma_i + O(T^{-1/2})$ can also be achieved **without additional knowledge** by more sophisticated algorithms.

Lower bound for the absolute loss

We show that the **square root term** is unavoidable. Thus we can only get

$$\frac{\text{Loss}(P)}{T} \leq \min_{1 \leq i \leq n} \sigma_i + O(T^{-1/2})$$

where again σ_i is the loss rate for expert \mathcal{E}_i . This is weaker than for square loss, log loss etc. which have

$$\frac{\text{Loss}(P)}{T} \leq \min_{1 \leq i \leq n} \sigma_i + O(T^{-1})$$

The bound is based on a simple **probabilistic argument**. We show a distribution for the experts predictions $x_{t,i}$ and outcomes y_t such that

$$\mathbb{E}[\text{Loss}(P)] \geq \mathbb{E}[\min_{1 \leq i \leq n} \text{Loss}(\mathcal{E}_i)] + \Omega(\sqrt{T})$$

where \mathbb{E} denotes expectation with respect to the choice of $x_{t,i}$ and y_t .

Since this holds on **average**, it of course also holds **in worst case**.

The distribution is simply that each y_t and $x_{t,i}$ is 1 with probability 1/2 and 0 with probability 1/2 (and they are all independent).

Clearly $E[\text{Loss}(P)] = E[\text{Loss}(\mathcal{E}_i)] = T/2$ for any i . More specifically, $\text{Loss}(\mathcal{E}_i)$ is a binomial random variable with mean $T/2$ and variance $\sqrt{T}/2$.

Define

$$F_i = \frac{\text{Loss}(\mathcal{E}_i) - T/2}{\sqrt{T}/2}.$$

Then F_i are independent, and for large T have approximately standard normal distribution (by central limit theorem). It is known that asymptotically,

$$E[\min_{1 \leq i \leq n} F_i] \approx -\sqrt{2 \ln n}.$$

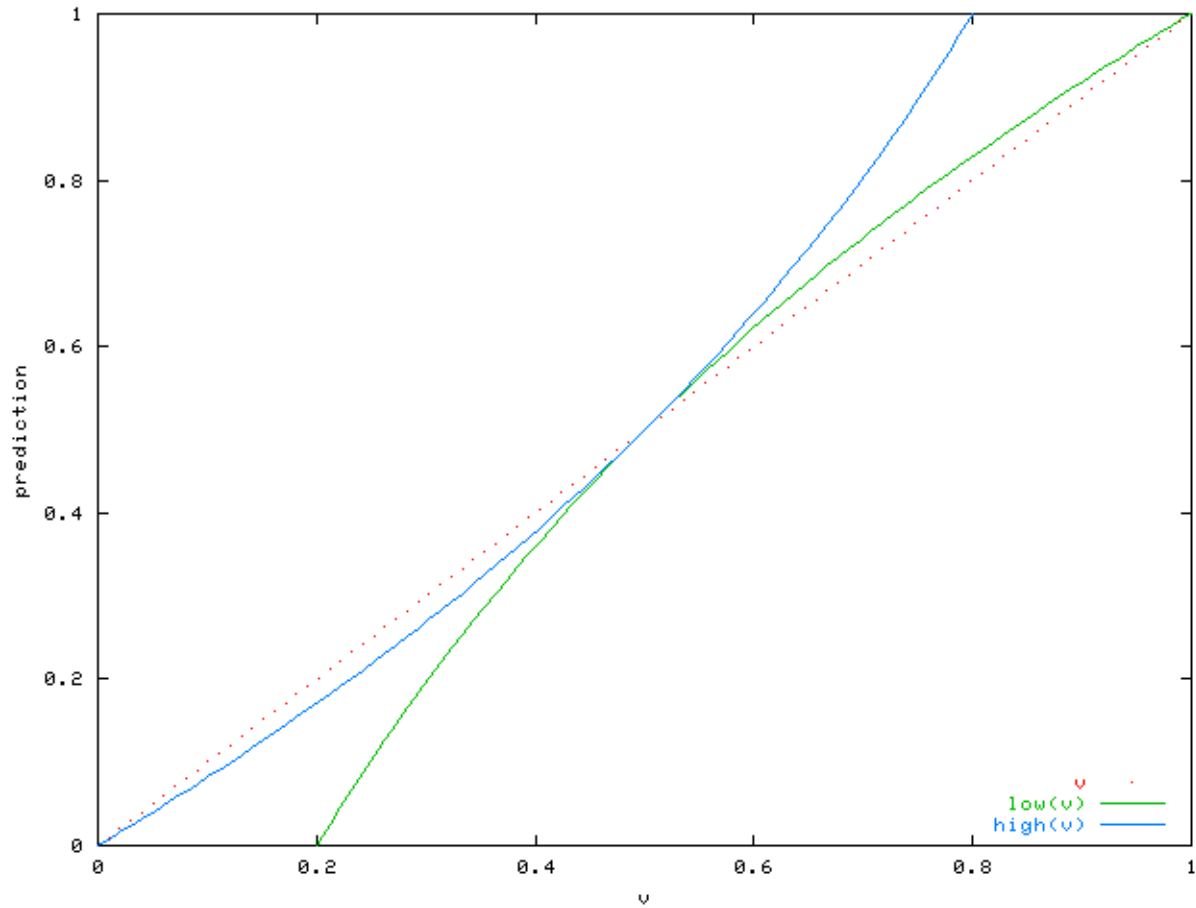
Thus

$$E[\min_{1 \leq i \leq n} \text{Loss}(\mathcal{E}_i)] \approx T/2 - \sqrt{\frac{T \ln n}{2}}$$

which gives the claim.

Further remarks on the prediction rule

- Weighted average $\hat{y}_t = \mathbf{v}_t \cdot \mathbf{x}_t$ will not in general work.
- Special case: for **log loss** $\hat{y}_t = \mathbf{v}_t \cdot \mathbf{x}_t$ is the **only** thing that works.
- There may not even be any function $\phi: [0, 1] \rightarrow [0, 1]$ such that $\hat{y}_t = \phi(\mathbf{v}_t \cdot \mathbf{x}_t)$ would work; however for absolute loss such a function ϕ does exist.
- The following plot shows the range of legal predictions for absolute loss in the special case $n = 2$, $(x_1, x_2) = (1, 0)$.



Upper and lower bounds for prediction \hat{y} when $(x_1, x_2) = (1, 0)$ as function of $v = w_1/(w_1 + w_2)$. (Here $\beta = 0.2$, *i.e.* fairly high learning rate.)

Tracking a sequence of experts

Previous methods give good results if a fixed single expert is good over the whole sequence of predictions.

Suppose the process we are predicting is non-stationary. Then perhaps different experts are good at different times.

Example Disk caching

- a number of different caching strategies
- some strategies may not be good at all in a given environment
- which one is really the best depends on what activity is going on right now

(There are various other problems in applying prediction algorithms in a caching setting.)

Formal setting Consider a sequence of T experts $U = (U_1, \dots, U_T)$, $U_t \in \{\mathcal{E}_i \mid 1 \leq i \leq n\}$ for all T .

- U has k *shifts* if there are exactly k indices $t \in \{1, \dots, T-1\}$ such that $U_t \neq U_{t+1}$.
- Expert \mathcal{E}_i is *active* in U if there is at least one index $t \in \{1, \dots, T\}$ such that $U_t = \mathcal{E}_i$.
- Let $\mathcal{U}(k, m)$ be the set of sequences with k shifts and m active experts, and $\mathcal{U}(k) = \cup_{m=1}^n \mathcal{U}(k, m)$
- Define $\mathbf{u} \in [0, 1]^T$ by $u_t = x_{t,i}$ where i is the index such that $U_t = \mathcal{E}_i$, and let

$$\text{Loss}(U) = \sum_{t=1}^T L(y_t, u_t).$$

Assume log loss for now. The basic expert bound

$$\text{Loss}(P) \leq \min_{1 \leq i \leq n} \text{Loss}(\mathcal{E}_i) + \ln n$$

has a (loose) **coding theoretic interpretation**:

- $\ln n$ is the number of bits (using natural logs for convenience) to encode the best expert
- $\min_{1 \leq i \leq n} \text{Loss}(\mathcal{E}_i)$ is the number of bits to encode the outcomes given the best expert.

Can we achieve the same here, *i.e.* have

$$\text{Loss}(P) \leq \min_{U \in \mathcal{U}(k)} \text{Loss}(U) + f(k)$$

where $f(k)$ is the coding length for a sequence $U \in \mathcal{U}(k)$? (And similarly for $\mathcal{U}(k, m)$.)

In principle this is possible by considering each $U \in \mathcal{U}(k)$ as a new expert and running Aggregating Algorithm on top of those. However, this would be computationally prohibitive since we would need $|\mathcal{U}(k)|$ weights.

To encode a sequence $\mathbf{U} \in \mathcal{U}(k)$ we need roughly

- $\log \binom{T-1}{k} \approx k \log(T/k)$ bits to encode the shift times and
- $(k + 1) \log n$ bits to encode the expert active at each shift.

To encode a sequence $\mathbf{U} \in \mathcal{U}(k, m)$ we need roughly

- $\log \binom{T-1}{k} \approx k \log(T/k)$ bits to encode the shift times,
- $\log \binom{n}{m} \approx m \log(n/m)$ bits to encode the *pool* of m active experts and
- $(k + 1) \log m$ bits to encode one expert from the pool for each shift.

Problem 1: Can we get

$$\text{Loss}(P) \leq \min_{U \in \mathcal{U}(k)} \text{Loss}(U) + O(k \log \frac{T}{k}) + O(k \log n)$$

with an efficient algorithm?

Problem 2: Can we get

$$\text{Loss}(P) \leq \min_{U \in \mathcal{U}(k,m)} \text{Loss}(U) + O(k \log \frac{T}{k}) + O(m \log \frac{n}{m}) + O(k \log m)$$

with an efficient algorithm?

- Here “efficient” means roughly $O(n)$ time per prediction and update.
- Going from 1 to 2 we replace $k \log n$ by $k \log m + m \log n$. This is interesting when $m \ll n$ and $m \ll k$.

The basic Aggregating Algorithm will **not** work here. The exponential update eradicates the weights of experts that initially perform poorly, and they never recover even if they later start performing better.

Thus we need to make sure each expert retains the chance of recovery if its performance improves.

Simple method: Divide a fixed portion of the total weight uniformly over all the experts. Thus everyone has a chance to recover. This is enough to solve Problem 1.

Fancy method(s): Divide a fixed portion of the total weight favouring the experts that performed well some time in the past. Thus the experts that are in the “pool” have faster recovery. This solves Problem 2.

We next sketch the simple method, called **Fixed Share Algorithm**.

Fixed Share Algorithm

Assume $0 \leq x_{t,i} \leq 1$ and $y_t \in \{0, 1\}$. Fix η and c as in the Aggregating Algorithm. Fix a share factor $0 \leq \gamma < 1$.

Prediction as in Aggregating Algorithm, using weights w_t .

Update at time t consist of two steps with an intermediate weight vector w'_t :

Loss update step: For $i = 1, \dots, n$ let

$$w'_{t+1,i} = w_{t,i} \exp(-\eta L(y_t, x_{t,i})).$$

Share update step: For $i = 1, \dots, n$ let

$$w_{t+1,i} = (1 - \gamma)w'_{t+1,i} + \frac{\gamma}{n} \sum_{j=1}^n w'_{t+1,j}.$$

Thus we take a fraction γ of the intermediate weight and redistribute it uniformly. (Notice that $\sum_i w_{t+1,i} = \sum_i w'_{t+1,i}$.)

For simplicity we state the result only for log loss and omit the finer issues of tuning γ .

Theorem For log loss the Fixed Share Algorithm with

$$\gamma = \frac{n}{n-1} \frac{k}{T-1}$$

satisfies

$$\text{Loss}(P) \leq \min_{U \in \mathcal{U}(k)} \text{Loss}(U) + (k+1) \ln n + k \ln \frac{T-1}{k} + k.$$

Thus the bound is as stipulated in Problem 1.

References and further reading

The basic algorithms

Vladimir Vovk: A game of prediction with expert advice. *Journal of Computer and System Sciences* 56:153–173 (1998).

Nick Littlestone and Manfred Warmuth: The weighted majority algorithm. *Information and Computation* 108:212–261 (1994).

Dealing with the absolute loss

Nicolò Cesa-Bianchi, Yoav Freund, David Haussler, David Helmbold and Robert Schapire: How to use expert advice. *Journal of the ACM* 44:427–485 (1997).

Fine-tuning for “nice” loss functions

Jyrki Kivinen, David Haussler and Manfred Warmuth: Sequential prediction of individual sequences under general loss functions. *IEEE Transactions on Information Theory* 44:1906–1925 (1998).

Jyrki Kivinen and Manfred Warmuth: Averaging expert predictions. In *EuroCOLT '99*, pp. 153–167 (1999).

Tracking expert sequences

Mark Herbster and Manfred Warmuth: Tracking the best expert. *Machine Learning* 32:151–178 (1998).

Olivier Bousquet and Manfred Warmuth: Tracking a small set of experts by mixing past posteriors. *Journal of Machine Learning Research* 3:363–396 (2002).

Robert Gramacy, Manfred Warmuth, Scott Brandt and Ismail Ari: Adaptive caching by refetching. In *NIPS 2002*.

Adaptive tuning of learning rate

Peter Auer, Nicolò Cesa-Bianchi and Claudio Gentile: Adaptive and self-confident on-line learning algorithms. *Journal of Computer and System Sciences* 64:48–75 (2002).

A parametric infinite set of experts

Vladimir Vovk: Competitive on-line statistics. *International Statistical Review* 69:213–248 (2001).

Related model: Universal Prediction

Marcus Hutter: General loss bounds for universal sequence prediction. In *18th International Conference on Machine Learning*, pp. 210–217 (2001).