How to Use Expert Advice
(Extended Abstract)

Nicolò Cesa-Bianchi[*] Università di Milano (Italy)
Yoav Freund UC Santa Cruz
David P. Helmbold[†] UC Santa Cruz
David Haussler[‡] UC Santa Cruz
Robert E. Schapire[§] AT&T Bell Labs
Manfred K. Warmuth UC Santa Cruz

Abstract

We analyze algorithms that predict a binary value by combining the predictions of several prediction strategies, called *experts*. Our analysis is for worst-case situations, i.e., we make no assumptions about the way the sequence of bits to be predicted is generated. We measure the performance of the algorithm by the difference between the expected number of mistakes it makes on the bit sequence and the expected number of mistakes made by the best expert on this sequence, where the expectation is taken with respect to the randomization in the predictions. We show that the minimum achievable difference is on the order of the square root of the number of mistakes of the best expert, and we give efficient algorithms that achieve this. Our upper and lower bounds have matching leading constants in most cases. We give implications of this result on the performance of batch learning algorithms in a PAC setting which improve on the best results currently known in this context. We also extend our analysis to the case in which log loss is used instead of the expected number of mistakes.

1 Introduction

A central problem in machine learning is the problem of predicting future events based on past observations. In computer science literature in particular, special attention has been given to the case in which the events are simple binary outcomes [16]. For example, in predicting today's weather, we may choose to consider only the possible outcomes 0 and 1, where 1 indicates that it rains today, and 0 indicates that it does not. In this paper we show that some simple prediction algorithms are optimal for this task in a sense that is closely related to the definitions of universal forecasting, prediction, and data compression which have been explored in the information theory literature. We then give applications of these results to the theory of PAC learning [31].

We take the extreme position, as advocated by Dawid and Vovk in the theory of Prequential Probability [4, 3, 5, 35], Rissanen in his theory of stochastic complexity [25, 27, 26, 37] and Cover, Lempel and Ziv, Feder and others in the theory of universal prediction and data compression of individual sequences [7, 24, 1, 2, 12, 36], that no assumptions whatsoever can be made about the actual sequence $\mathbf{y} = y_1, \ldots, y_\ell$ of outcomes that is observed; the analysis is done in the *worst case* over all possible binary outcome sequences. Of course no method of prediction can do better than random guessing in the worst case, so a naive worst-case analysis is fruitless. To illustrate an alternative approach in the vein of universal prediction, consider the following scenario.

Let us suppose that on each morning t you must predict whether or not it will rain that day (i.e., the value of y_t), but before you make your prediction you are allowed to hear the predictions of a (fixed) finite set $\mathcal{E} = \{\mathcal{E}_1, \ldots, \mathcal{E}_N\}$ of *experts*. On the morning of day t, each expert has access to the weather outcomes $y_1, \ldots, y_{t-1}$ of the previous $t-1$ days, and possibly to the values of other weather measurements $x_1, \ldots, x_{t-1}$ made on those days, as well as today's measurements x_t. The measurements $x_1, \ldots, x_\ell$ will be called *instances*. Based on this data, each expert returns a real number p between 0 and 1 that can be interpreted as his/her estimate of the probability that it will rain that day. After hearing the predictions of the experts, you also choose a number $p \in [0,1]$ as your estimate of the probability of rain. Later in the day, nature sets the value of y_t to either 1 or 0 by either raining or not raining. In the evening, you and the experts are scored. A person receives the *loss* $|p-y|$ for making prediction $p \in [0,1]$ when the actual outcome is $y \in \{0,1\}$. To see why this is a reasonable measure of loss,[1] imagine that instead of returning $p \in [0,1]$ you tossed a biased coin and predicted outcome 1 with probability p and outcome 0 with probability $1-p$. Then $|p-y|$ is the probability that your prediction is incorrect when the actual outcome is y.

Let us fix the instance sequence $x_1, \ldots, x_\ell$, since it plays only a minor role here, and vary only the outcome sequence $\mathbf{y} = y_1, \ldots, y_\ell$. Imagine that the above prediction game is played for ℓ days, during which time you accumulate a total loss $L(\mathbf{y}) = \sum_{t=1}^{\ell} |\psi_t - y_t|$, where $\psi_t \in [0,1]$ is your prediction at time t. Each of the experts also accumulates a total loss based on his/her predictions. Your goal is to *try to predict as well as the best expert, no matter what out-*

[*]This research was done while this author was visiting UC Santa Cruz partially supported by the "Progetto finalizzato sistemi informatici e calcolo parallelo" of CNR under grant 91.00884.69.115.09672
Email address cesabian@imiucca.csi unimi it
[†]Email address: dph@cse.ucsc edu
[‡]Haussler, Warmuth and Freund are supported by ONR grant NO0014-91-J-1162 and NSF grant IRI-9123692
Email addresses: {haussler,manfred,yoav}@cse.ucsc.edu
[§]Email address schapire@research att com

25th ACM STOC '93-5/93/CA,USA
© 1993 ACM 0-89791-591-7/93/0005/0382...$1.50

[1] An alternate logarithmic loss function, often considered in the literature, is discussed briefly in Section 8.

come sequence $\mathbf{y}$ *is produced by nature.*[2] Specifically, if we let $L_{\mathcal{E}}(\mathbf{y})$ denote the minimum total loss of any expert on the particular sequence $\mathbf{y}$, then your goal is to minimize the maximum of the difference $L(\mathbf{y}) - L_{\mathcal{E}}(\mathbf{y})$ over all possible binary sequences $\mathbf{y}$ of length ℓ. Since most outcome sequences will look totally random to you, you still won't be able to do better than random guessing on most sequences. However, since most sequences will also look totally random to all the experts (as long as there aren't too many experts), you may still hope to do almost as well as the best expert in most cases. The difficult sequences are the ones that have some structure that is exploited by one of the experts. To do well on these sequences you must quickly zero in on the fact that one of the experts is doing well, and match his/her performance, perhaps by mimicking his/her predictions.

Through a game-theoretic analysis, we find that for any finite set of experts, there is a strategy that minimizes the maximum of the difference $L(\mathbf{y}) - L_{\mathcal{E}}(\mathbf{y})$ over all possible binary outcome sequences $\mathbf{y}$. While this minimax strategy can be implemented in some cases, it is not practical in general. However, we define an algorithm, called $\mathbf{P}$ for "Predict", that is simple and efficient, and performs essentially as well as the minimax strategy. Actually $\mathbf{P}$ is a family of algorithms that is related to the algorithm studied by Vovk [34] and the Bayesian, Gibbs and "weighted majority" methods studied by a number of authors [23, 22, 15, 29, 28, 13, 18], as well as the method developed by Feder, Merhav and Gutman [7]. We show that $\mathbf{P}$ performs quite well in the sense defined above so that, for example, given any finite set $\mathcal{E}$ of weather forecasting experts, $\mathbf{P}$ is guaranteed not to perform much worse than the best expert in $\mathcal{E}$, no matter what the actual weather turns out to be. The algorithm $\mathbf{P}$ is completely generic in that it makes no use of the side information provided by the instances $x_1, \ldots, x_\ell$. Thus, it would also do almost as well as the Wall Street expert with the best inside information when predicting whether the stock market will rise or fall.

In particular, letting $L_P(\mathbf{y})$ denote the total loss of algorithm $\mathbf{P}$ on the sequence $\mathbf{y}$ and $L_{\mathcal{E}}(\mathbf{y})$ the loss of the best expert on $\mathbf{y}$ as above, we show (Theorem 6) that for all outcome sequences $\mathbf{y}$ of length ℓ, $L_P(\mathbf{y}) - L_{\mathcal{E}}(\mathbf{y}) \leq \sqrt{\frac{\ell \ln(|\mathcal{E}|+1)}{2}} + \frac{\log_2(|\mathcal{E}|+1)}{2}$, and that no algorithm can improve the multiplicative constant of the square-root term.

Previous work has shown how to construct an algorithm A such that the ratio $L_A(\mathbf{y})/L_{\mathcal{E}}(\mathbf{y})$ approaches 1 in the limit [34, 23, 7]. In fact, Vovk [34] described an algorithm with the same bound as the one we give in Theorem 2 for the algorithm $\mathbf{P}$. This theorem leaves a parameter to be tuned. Vovk gives an implicit form of the optimum choice of the parameter. We arrive at an explicit form that allows us to prove nearly optimal bounds on $L_A(\mathbf{y}) - L_{\mathcal{E}}(\mathbf{y})$. To our knowledge, our results give the first precise bounds on this difference.

It turns out that these bounds also give a tight lower bound on the expectation of the minimal distance between a random binary string uniformly chosen from $\{0,1\}^\ell$ and a set of N points in $[0,1]^\ell$, which may be of independent interest from a combinatorial viewpoint.

The remainder of this paper is organized as follows. In Section 2 we introduce some notation. In Section 3, we characterize exactly the performance of the best possible prediction strategy using a minimax analysis. Section 4 describes the algorithm $\mathbf{P}$ and shows that it achieves the optimal bound given above. In Section 4.1 we show that if the loss $L_{\mathcal{E}}(\mathbf{y})$ of the best expert is given to the algorithm *a priori*, then $\mathbf{P}$ can be tuned so that $L_P(\mathbf{y}) - L_{\mathcal{E}}(\mathbf{y}) \leq \sqrt{L_{\mathcal{E}}(\mathbf{y}) \ln |\mathcal{E}|} + \frac{\log_2 |\mathcal{E}|}{2}$. In Section 5 we show that when no knowledge of $L_{\mathcal{E}}(\mathbf{y})$ is available, then using a simple doubling trick we can still obtain a bound that is only a small constant factor larger. This algorithm can nearly match the performance of the best expert on all prefixes of an infinite sequence $\mathbf{y}$.

Finally, we show how the results we have obtained can be applied in other machine learning contexts. In Section 6, we look at the case when one outcome in a sequence of outcomes is covered up at random. We are asked to predict only this covered outcome, based on the values of the other outcomes and other side information. We call this "hold-one-out" prediction, and we show that a variant of $\mathbf{P}$ performs optimally for this problem as well.

Next, in Section 7, we relate the hold-one-out prediction problem to one studied in the PAC learning literature in which examples $(x_1, y_1), \ldots, (x_{t-1}, y_{t-1})$ are drawn independently at random from some arbitrary distribution on the set of all possible labeled examples and the goal is to find a hypothesis that will predict the binary label y_t of the next random example (x_t, y_t) correctly with as high a probability as can be obtained. Using a permutation argument [18, 33], we are able to apply the bounds obtained in Section 6 for the hold-one-out variant of $\mathbf{P}$ to get distribution-independent bounds for the performance on this task as well. These bounds are more robust and improve by constant factors some of the (more general) bounds obtained by Vapnik [33] and Talagrand [30] on the performance of an empirical loss minimization algorithm.

The results presented in this paper contribute to an ongoing program in information theory and statistics to minimize the number of assumptions placed on the actual mechanism generating the observations through the development of robust procedures and strengthened worst-case analysis. In investigating this area, we have been struck by the fact that many of the standard-style statistical results that we have found most useful, such as the bounds given by Vapnik, in fact have worst-case counterparts which are much stronger than we had expected would be possible. We believe that if these results can be extended to more general loss functions and learning/prediction scenarios, with corresponding optimal estimation of constants and rates, this worst-case viewpoint may ultimately provide a fruitful alternative foundation for the statistical theory of learning and prediction.

2 Preliminaries

We denote the set of experts by $\mathcal{E} = \{\mathcal{E}_1, \ldots, \mathcal{E}_N\}$, where N is the number of experts. The binary sequence to be predicted is denoted by $\mathbf{y} = y_1, \ldots, y_t, \ldots y_\ell$, where t is the index of a typical time step (or *trial*) and ℓ is the length of the sequence (if it is finite). The prediction given by the expert $\mathcal{E}_i$ at time t is denoted by $\xi_{i,t} \in [0,1]$, and the prediction of the algorithm by $\psi_t \in [0,1]$. Side information that might be used as input by the experts is denoted by the sequence of instances $\mathbf{x} = x_1, \ldots, x_\ell$. We denote the loss that an algorithm A incurs on the sequence $\mathbf{y}$ by $L_A(\mathbf{y}) = \sum_{t=1}^{\ell} |\psi_t - y_t|$ and the loss of the best expert in $\mathcal{E}$ on the same sequence by $L_{\mathcal{E}}(\mathbf{y}) = \min_{1 \leq i \leq N} L_{\mathcal{E}_i}(\mathbf{y})$.

[2] This approach is also related to that taken in recent work on the competitive ratio of on-line algorithms, and in particular to work on combining on-line algorithms to obtain the best competitive ratio [9, 8, 10], except that we look at the difference in performance rather than the ratio

3 An optimal prediction strategy

As described in the introduction, our goal is to minimize the worst-case difference between the loss of our prediction strategy and the loss of the best expert. In this section, we derive the optimal value of this difference. That is, we exhibit an algorithm that exactly achieves this optimal value for *every* sequence $\mathbf{y}$, and moreover, we show that every other prediction strategy does worse on some sequence $\mathbf{y}$. To obtain this result, we assume that the learner knows the length ℓ of the sequence $\mathbf{y}$, and is able to foresee the experts' future predictions. We say that a set of experts is *simulatable* if, given the sequence of outcomes and expert predictions up to time $t-1$, the predictions of the experts at time t can be calculated. Implicit in this definition is an assumption that the instances $\mathbf{x}$ are either irrelevant or known ahead of time.

Theorem 1 *Let $\mathcal{E}$ be a set of simulatable experts. Then there exists a prediction strategy A^* such that for every sequence $\mathbf{y}$, we have*

$$L_{A^*}(\mathbf{y}) - L_{\mathcal{E}}(\mathbf{y}) = \frac{\ell}{2} - E_{\mathbf{y}'}(L_{\mathcal{E}}(\mathbf{y}'))$$

where $E_{\mathbf{y}'}$ denotes expectation over a uniformly random choice of $\mathbf{y}'$ from $\{0,1\}^\ell$. Moreover, A^* is optimal in the sense that for every prediction strategy A, there exists a sequence $\mathbf{y}$ such that

$$L_A(\mathbf{y}) - L_{\mathcal{E}}(\mathbf{y}) \geq \frac{\ell}{2} - E_{\mathbf{y}'}(L_{\mathcal{E}}(\mathbf{y}')).$$

In proving this theorem, we find it useful to view the problem in a game-theoretic setting in which the value $V^* = \frac{\ell}{2} - E_{\mathbf{y}'}(L_{\mathcal{E}}(\mathbf{y}'))$ turns out to be the minimax solution of a zero-sum, perfect-information, two-person game. Here is a very brief outline of the proof: For any prediction strategy A, let $V_A(\mathbf{y}) = L_A(\mathbf{y}) - L_{\mathcal{E}}(\mathbf{y})$. Then the expected value of V_A with respect to a uniformly random choice of $\mathbf{y}' \in \{0,1\}^\ell$ is simply $\ell/2 - E_{\mathbf{y}'}(L_{\mathcal{E}}(\mathbf{y}'))$ since we expect any algorithm to make $\ell/2$ prediction errors on an entirely random sequence. Thus, there must be some sequence $\mathbf{y}$ for which $V_A(\mathbf{y})$ is at least as great as this expectation; this proves the second part of the theorem. For the first part of the theorem, it suffices to show that there exists a prediction strategy A^* that yields the same difference in loss $V^* = V_{A^*}(\mathbf{y})$ for every sequence $\mathbf{y}$. We prove this fact in a constructive manner by induction on the number of remaining steps $\ell - t$ (details omitted).

Theorem 1 tells us how to compute the worst-case performance of the best possible algorithm for any set of experts. As an example of its usefulness, suppose that $\mathcal{E}$ consists of only two experts, one that always predicts 0, and the other always predicting 1. In this case Theorem 1 implies that the loss of the optimal algorithm A^* is worse than the loss of the best expert by the following amount :

$$V^* = \frac{\ell}{2} - 2^{-\ell} \sum_{i=0}^{\ell} \binom{\ell}{i} \min\{i, \ell - i\} \sim \sqrt{\frac{\ell}{2\pi}}.$$

This result was previously proved by Cover [1]; we obtain it as a special case.

Strategy A^* makes each prediction in terms of the expected loss of the best expert on the remaining trials (where the expectation is taken over the uniformly random choice of outcomes for these trials). In general, calculating this expectation exactly is intractable. However, the expectation can be estimated by sampling a polynomial number of randomly chosen future outcome sequences.

Note also that A^* relies on the knowledge of how the experts will predict in the future. In the next section, we will describe a simple prediction algorithm that achieves similar performance in the absence of this assumption. This algorithm will also yield general upper bounds on the optimal value V^* given in Theorem 1.

Algorithm $\mathbf{P}(\beta)$

1. All initial weights $\{w_{1,1}, \ldots, w_{N,1}\}$ are set to 1.
2. At each time t, for $t = 1$ to ∞, the algorithm receives the predictions of the N experts, $\xi_{1,t}, \ldots, \xi_{N,t}$, and computes its prediction ψ_t as follows:
   - Compute
   $$r_t := \frac{\sum_{i=1}^{N} w_{i,t}\xi_{i,t}}{\sum_{i=1}^{N} w_{i,t}}$$
   - Output prediction $\psi_t = F_\beta(r_t)$.
3. After the correct outcome y_t is observed, the weight vector is updated in the following way.
   - For each $i = 1$ to N, $w_{i,t+1} = w_{i,t}\, U_\beta(|\xi_{i,t} - y_t|)$.

There is some flexibility in defining the functions $F_\beta(r)$ and $U_\beta(q)$ used in the algorithm. Any functions $F_\beta(r)$ and $U_\beta(q)$ such that

$$1 + \frac{\ln((1-r)\beta + r)}{2\ln(\frac{2}{1+\beta})} \leq F_\beta(r) \leq \frac{-\ln(1-r+r\beta)}{2\ln(\frac{2}{1+\beta})}, \quad (1)$$

for all $0 \leq r \leq 1$, and

$$\beta^q \leq U_\beta(q) \leq 1 - (1-\beta)q, \quad (2)$$

for all $0 \leq q \leq 1$, will achieve the performance bounds established below.

Figure 1: Description of Algorithm $\mathbf{P}(\beta)$, with parameter $0 \leq \beta < 1$.

4 Some simple prediction algorithms

In this section, we present a parameterized prediction algorithm $\mathbf{P}$ that will be used throughout the paper. We will show that this algorithm achieves performance bounds similar to those given in Theorem 1. We assume a basic setup similar to that in the previous section. However, the predictions generated by algorithm $\mathbf{P}$ depend only on the past and present predictions of the experts and on the previously observed outcomes in the sequence $\mathbf{y}$. Thus, there is no restriction on how the experts generate their predictions. In particular, the experts might not be simulatable, e.g. because the experts' predictions depend on some external sources of information that are unavailable to the algorithm.

The prediction algorithm $\mathbf{P}$ is given in Figure 1. It works by maintaining a (non-negative) *weight* for each expert. The weight of expert i at time t is denoted $w_{i,t}$. At each time t, the algorithm receives the experts' predictions, $\xi_{1,t}, \ldots, \xi_{N,t}$, and computes their weighted average, r_t. Algorithm $\mathbf{P}$ then makes a prediction that is some function of this weighted average. Then $\mathbf{P}$ receives the correct value y_t and slashes the

weight of each expert i by a multiplicative factor depending on how well that expert predicts, as measured by $|\xi_{i,t} - y_t|$.[3] The worse the prediction of the expert, the more that expert's weight is reduced.

Algorithm **P** takes one parameter, a real number $\beta \in [0, 1)$ which controls how quickly the weights of poorly predicting experts drop. For small β, the algorithm quickly slashes the weights of poorly predicting experts and starts paying attention only to the better predictors. For β closer to 1, the weights will drop slowly, and the algorithm will pay attention to a wider range of predictors for a longer time. The best value for β depends on the circumstances. Later we derive good choices of β for different types of prior knowledge the algorithm may have.

There are two places where the algorithm can choose to use any real value within an allowed range. We have represented these choices by the functions F_β and U_β, with ranges given by (1) and (2), respectively, in Figure 1. In terms of our analysis, the exact choice for these values is not important, as long as they lie in the allowed range. In fact, different choices could be made at different times.

The function $U_\beta(q)$ is called the *update function.* Its lower bound is the exponential β^q and its upper bound is the linear function $1 - (1 - \beta)q$. In related work, Vovk uses the exponential update function [34], and Littlestone and Warmuth [23] use any update between the exponential and the linear update. It turns out that the linear update has a nice Bayesian interpretation.[4]

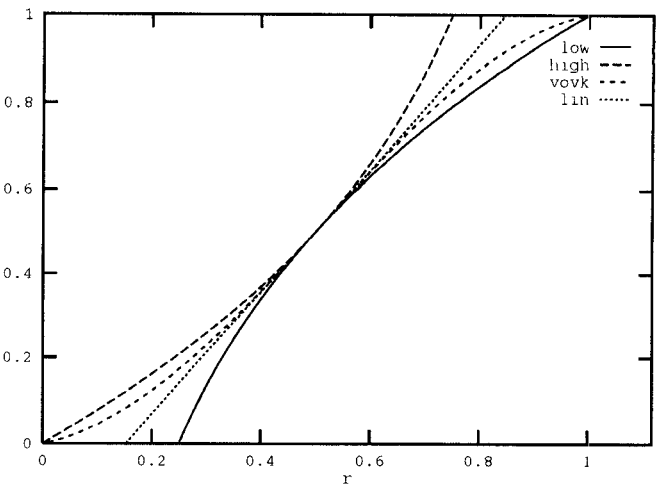

Figure 2: This figure shows the upper (high) and lower (low) bounds on the possible values of the prediction function F_β for $\beta = 0$ (Inequality (1)). Also shown are two possible choices for F_β, a piecewise linear function (lin) given in (4), and the function that has been suggested by Vovk's work (vovk) given in (5).

One function that satisfies the requirements for F_β is the piecewise linear function[5]

$$F_\beta(r) = \begin{cases} 0 & \text{if } r \le \frac{1}{2} - c \\ \frac{1}{2}\left(1 - \frac{(1-2r)(1-\beta)}{(1+\beta)\ln(\frac{2}{1+\beta})}\right) & \text{if } \frac{1}{2} - c \le r \le \frac{1}{2} + c \\ 1 & \text{if } r \ge \frac{1}{2} + c \end{cases} \tag{4}$$

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

Another possible choice for F_β is suggested by Vovk's work[6] [34]

$$F_\beta(r) = \frac{\ln(1 - r + r\beta)}{\ln(1 - r + r\beta) + \ln((1-r)\beta + r)}. \tag{5}$$

These functions, for $\beta = 0$, along with the upper and lower bounds for F_β, given in Inequality (1), are shown in Figure 2. As β goes to one the two bounds merge at r.

Algorithm **P**'s performance is summarized by the following theorem. This parameterized bound was first proved by Vovk [34] for his version of F_β and the exponential update. For the noise-free case ($\beta = 0$), slightly weaker upper bounds have been proved for an algorithm known as the Gibbs algorithm [23, 17]. Also, Littlestone and Warmuth [23] prove a

[3] If the experts' weights are normalized so they sum to one, then the weight of an expert which is predicting well will increase as the weights of other experts are reduced

[4] Here we view each expert as a probability distribution on bit sequences of length ℓ, and pretend that the actual sequence $\mathbf{y} = y_1, \ldots, y_\ell$ is generated by picking an expert uniformly at random and then generating a bit sequence of length ℓ at random according to the distribution defined by that expert The probability distribution for the i^{th} expert is defined as follows For any $y_1, \ldots, y_{t-1}$, if the expert's estimate of the probability that $y_t = 1$ given $y_1, \ldots, y_{t-1}$ is $\xi_{i,t}$, then the actual probability that y_t is 1 given $y_1, \ldots, y_{t-1}$ is defined to be

$$p_{i,t} = \eta + (1 - 2\eta)\xi_{i,t}, \tag{3}$$

where $\eta = \beta/(1+\beta)$ It is easy to see that $p_{i,t}$ is just the probability that y_t is 1 if originally y_t is set to 1 with probability $\xi_{i,t}$ and 0 with probability $1 - \xi_{i,t}$, and then the value of y_t is flipped with independent probability η Hence the value η can be interpreted as a "subjective" noise rate between 0 and 1/2 Under this interpretation, we have the following result.

When the update function U_β of the algorithm $\mathbf{P}(\beta)$ has the form

$$U_\beta(q) = 1 - (1 - \beta)q,$$

then the (normalized) weight $w_{i,t}/(\sum_{j=1}^N w_{j,t})$ is the posterior probability that the outcome sequence is being generated from the distribution defined by the i^{th} expert given the previous outcomes $y_1, \ldots, y_{t-1}$, assuming that all N expert distributions are a priori *equally likely to be generating the sequence.*

Since the weights are posterior probabilities on the experts, the weighted average r_t of the expert's predictions, computed by the algorithm **P**, also has a Bayesian interpretation it is simply the posterior probability that $y_t = 1$ given $y_1, \ldots, y_{t-1}$ The only aspect of the algorithm that does not have a Bayesian interpretation is the *prediction function* $F_\beta(r)$. A Bayes method would predict 1 whenever the posterior probability r_t is greater than 1/2 and predict 0 otherwise, in order to minimize the posterior expectation of the loss $|\psi_t - y_t|$ Thus a Bayes method would use a step function at 1/2 for the prediction function $F_\beta(r)$ However, as is clear from Figure 2, this function lies outside the allowable range for $F_\beta(r)$, and this is no accident. The Bayes method does not perform well in the worst case for this prediction problem, as was shown in [17, 7] Hence we must deviate from the Bayes method at this step

[5] A similar piecewise linear function was suggested by Feder et al [7], in a related context

[6] Vovk's algorithm generates its prediction according to the prediction function

$$\psi_t = \frac{\ln \sum_{i=1}^N w_{i,t}\beta^{\xi_{i,t}}}{\ln \sum_{i=1}^N w_{i,t}\beta^{\xi_{i,t}} + \ln \sum_{i=1}^N w_{i,t}\beta^{1-\xi_{i,t}}},$$

where the weights are normalized so that they sum to one Note that this function depends on the experts' predictions in a more complicated way than just through the weighted average r_t Hence it does not always satisfy our assumption of Inequality (1) However, when the experts' predictions are all in $\{0, 1\}$, then Vovk's prediction function is equivalent to the one described in Equation (5)

bound for their Algorithm WMC which has the same form as the bound below except the denominator $2\ln\frac{2}{1+\beta}$ is replaced by the smaller function of $1-\beta$. However their algorithm works for the more general setting when the outcome y_t can be in $[0,1]$ as opposed to being binary.

Theorem 2 *For any $0 \le \beta < 1$, for any set $\mathcal{E}$ of N experts, and for any binary sequence $\mathbf{y}$ of length ℓ, the loss of $\mathbf{P}(\beta)$ is bounded by*

$$L_{P(\beta)}(\mathbf{y}) \le \frac{\ln N - L_{\mathcal{E}}(\mathbf{y})\ln\beta}{2\ln\frac{2}{1+\beta}}.$$

The proof of the theorem is based on the following lemma.

Lemma 3

$$L_{P(\beta)}(\mathbf{y}) \le \frac{\ln\left(\frac{\sum_{i=1}^N w_{i,1}}{\sum_{i=1}^N w_{i,\ell+1}}\right)}{2\ln\frac{2}{1+\beta}}$$

Proof: We will show that for $1 \le t \le \ell$,

$$|\psi_t - y_t| \le \frac{\ln\left(\frac{\sum_{i=1}^N w_{i,t}}{\sum_{i=1}^N w_{i,t+1}}\right)}{2\ln\frac{2}{1+\beta}} \qquad (6)$$

The lemma then follows from summing the above inequality for all choices of t. We first lower bound the numerator of the right-hand-side of the above inequality:

$$\ln\left(\frac{\sum_{i=1}^N w_{i,t}}{\sum_{i=1}^N w_{i,t+1}}\right) = -\ln\left(\frac{\sum_{i=1}^N w_{i,t}U_\beta(|\xi_{i,t}-y_t|)}{\sum_{i=1}^N w_{i,t}}\right)$$

$$\ge -\ln\left(\frac{\sum_{i=1}^N w_{i,t}(1-(1-\beta)|\xi_{i,t}-y_t|)}{\sum_{i=1}^N w_{i,t}}\right) \text{ by (2)}$$

$$= -\ln(1-(1-\beta)|r_t-y_t|), \text{ where } r_t = \frac{\sum_{i=1}^N w_{i,t}\xi_{i,t}}{\sum_{i=1}^N w_{i,t}}.$$

(In the last equality we use the fact that $y_t \in \{0,1\}$.) Thus Inequality (6) is implied by

$$|\psi_t - y_t| \le -\frac{\ln(1-(1-\beta)|r_t-y_t|)}{2\ln\frac{2}{1+\beta}}.$$

The above splits into two inequalities since y_t is either 0 or 1. These two inequalities are the same as the two inequalities of (1) which we assumed for the prediction function. ∎

Proof of theorem 2: All initial weights equal 1 and thus $\sum_{i=1}^N w_{i,1} = N$. Let j be an expert with minimum total loss on $\mathbf{y}$, that is, $\sum_{t=1}^\ell |\xi_{j,t}-y_t| = L_{\mathcal{E}}(\mathbf{y})$. We first show that the total final weight, $\sum_{i=1}^N w_{i,\ell+1}$, is lower bounded by $\beta^{L_{\mathcal{E}}(\mathbf{y})}$: Since $U_\beta(q) \ge \beta^q$ (Inequality (2)),

$$\sum_{i=1}^N w_{i,\ell+1} \ge w_{j,\ell+1} = w_{j,1}\prod_{t=1}^\ell U_\beta(|\xi_{j,t}-y_t|)$$

$$\ge \prod_{t=1}^\ell \beta^{|\xi_{j,t}-y_t|} = \beta^{L_{\mathcal{E}}(\mathbf{y})}.$$

Now the theorem follows from Lemma 3. ∎

4.1 Performance for bounded $L_{\mathcal{E}}$

So far we have ignored the issue of how β is chosen. In this section we show how β can be chosen when there is a known bound K on the loss of the best expert.

It turns out that the following function is important for the proper choice of β:

$$g(z) = 1 - 2\frac{\sqrt{1+z}-1}{z}, \qquad (7)$$

and we define $g(0) = 0$. The key property of this function is the following inequality

Lemma 4 *For any real value $z \ge 0$*

$$z + \sqrt{z} + \frac{1}{2\ln(2)} \ge \frac{1 - z\ln(g(z))}{2\ln(2/(1+g(z)))}$$

Using the function g to make our choice of β we get the following bound.

Theorem 5 *Let $\beta = g(K/\ln N)$, for the g defined in Equation (7). Then for any set $\mathcal{E}$ of N experts and for any sequence $\mathbf{y}$ such that $L_{\mathcal{E}}(\mathbf{y}) \le K$, we have*

$$L_{P(\beta)}(\mathbf{y}) - L_{\mathcal{E}}(\mathbf{y}) \le \sqrt{K\ln N} + \frac{\log_2 N}{2}.$$

The proof is a direct application of the inequality given in Lemma 4 to the bound given in Theorem 2.

4.2 Performance for known sequence length

As a corollary of Theorem 5, we can devise a choice for β that will guarantee a bound on the difference between the loss of the algorithm and the loss of the best expert for the case where ℓ, the length of the sequence to be predicted, is given to the algorithm in advance. We will later show that the guaranteed difference is very close to optimal.

Theorem 6 *Let $\beta = g(\ell/2\ln N)$. Then for any set $\mathcal{E}$ of N experts, and for any sequence $\mathbf{y}$ of length ℓ*

$$L_{P(\beta)}(\mathbf{y}) - L_{\mathcal{E}}(\mathbf{y}) \le \sqrt{\frac{\ell\ln(N+1)}{2}} + \frac{\log_2(N+1)}{2}.$$

Proof: Add an $N+1^{\text{st}}$ expert that predicts the opposite of the first expert, i.e. $\xi_{N+1,t} = 1 - \xi_{1,t}$. Then $L_{\mathcal{E}}(\mathbf{y}) \le \ell/2$ for all $\mathbf{y}$. The result follows from Th. 5 with $K = \ell/2$. ∎

We remark that while the bound stated in Theorem 6 holds for all ℓ, there is a slightly better bound on $\mathbf{P}(\beta)$ when $\ell \to \infty$:

$$L_{P(\beta)}(\mathcal{E}) - L_{\mathcal{E}}(\mathbf{y}) \le \sqrt{\frac{\ell\ln(N+1)}{2}} + (\frac{1}{2} + o(1))\ln N.$$

This is (asymptotically) the best bound that can be proved for $\mathbf{P}$ using Theorem 2.

We now use Theorem 1 to give a matching asymptotic lower bound on the performance of any prediction algorithm.

Theorem 7 *There exists a set $\mathcal{E}$ of N experts such that for every prediction strategy A, there is a sequence $\mathbf{y}$ of length ℓ such that*

$$L_A(\mathbf{y}) - L_{\mathcal{E}}(\mathbf{y}) \ge (1-o(1))\sqrt{\frac{\ell\ln N}{2}}$$

where $o(1)$ is a quantity that tends to zero as $\ell, N \to \infty$.

The proof of this theorem is based on the following well-known lemma about order statistics.

Lemma 8 *For each $\ell, N \geq 1$ let $S_{\ell,1}, \ldots, S_{\ell,N}$ be N independent random variables, where $S_{\ell,i}$ is the number of heads in ℓ independent tosses of a fair coin.*

Let $A_{\ell,N} = \min_{1 \leq i \leq N}\{S_{\ell,i}\}$. Then

$$\lim_{N \to \infty} \lim_{\ell \to \infty} \frac{\frac{\ell}{2} - E(A_{\ell,N})}{\sqrt{\ell \ln N/2}} = \sqrt{2}$$

The proof of the lemma is based on the central limit theorem and a theorem about the expected value of the minimum of a set of independent normally distributed random variables (see e.g. Galambos [11]).

Proof of Theorem 7: We use the following random construction. Assume that each bit in the sequence $\mathbf{y}$ of length ℓ is chosen independently at random using a fair coin flip. Assume that each prediction of each expert is similarly chosen independently at random to be either 0 or 1. From Lemma 8 it follows that the expected value of $L_\mathcal{E}$, over the random choice of $\mathbf{y}$ and of the expert predictions, is $\ell/2 - (1 - o(1))\sqrt{\ell \ln N/2}$. Thus there exist specific choices of expert predictions for which the expected value of $L_\mathcal{E}$ over the random choice of $\mathbf{y}$ is at most $\ell/2 - (1 - o(1))\sqrt{\ell \ln N/2}$. Assume we fix such a choice.

Assume now that we make predictions using the min-max algorithm from Theorem (1). Clearly, this prediction algorithm, as well as any other prediction algorithm, will suffer a loss of exactly $\ell/2$ on average over uniformly distributed binary sequences. On the other hand, Theorem 1 guarantees that the difference between its loss and the loss of the best expert is *the same for every sequence*, and is thus at least $(1-o(1))\sqrt{\ell \ln N/2}$, for the fixed experts' predictions and for every sequence $\mathbf{y}$. Finally, as this is the min-max algorithm, for every other algorithm A there exists a sequence $\mathbf{y}_A$ on which the difference between the loss of the algorithm and the loss of the best expert is at least $(1 - o(1))\sqrt{\ell \ln N/2}$. ∎

As a final note, from Theorem 6 we get an interesting corollary concerning the uniform distribution on binary strings.

Corollary 9 *Let A be a finite set of points in $[0,1]^\ell$, let $\mathbf{y}$ be in $\{0,1\}^\ell$ and let $d(\mathbf{y}, A)$ denote the l_1 distance between $\mathbf{y}$ and the closest point in A. Then*

$$\frac{\ell}{2} - \sqrt{\frac{\ell \ln(|A|+1)}{2}} - \frac{\log_2(|A|+1)}{2} \leq E(d(\mathbf{y}, A)) \leq \frac{\ell}{2}$$

where the expectation is over the uniform distribution on points $\mathbf{y}$ in $\{0,1\}^\ell$.

The proof of this is based on the case in which the predictions of the experts depend only on the time t. In this case $\mathcal{E}$ is isomorphic to a set of points in $[0,1]^\ell$. The left inequality follows from Theorem 1 and the loss bound proven for $\mathbf{P}$ in Theorem 6. The right inequality is elementary.

5 Prediction without prior knowledge

In the previous section we showed how to tune β so that $\mathbf{P}(\beta)$ has optimal performance when either a bound on the loss of the best expert or the length of the sequence, ℓ, is known to the algorithm. Here we present a version of the algorithm, Algorithm $\mathbf{P}^*$, that achieves similar performance when neither the length of the sequence nor the loss of the best expert is known. Algorithm $\mathbf{P}^*$ repeatedly guesses different loss bounds until it guesses a bound greater than the remaining loss of the best expert.

Algorithm $\mathbf{P}^*(a, c)$:
Parameters a and c are constants. We show later that good values for these parameters are $a = 1$ and $c = 2.618$
for $l := 1$ to ∞ do
  $k_l := a^2 c^l \ln N$;
  $b_l := k_l + \sqrt{k_l \ln N} + \frac{\lg N}{2}$
  Reset all weights of the experts to 1.
  repeat
    run $\mathbf{P}(g(k_l/\ln N))$ to generate predictions
  until the total loss in this loop exceeds b_l.

Figure 3: Description of Algorithm $\mathbf{P}^*$.

Algorithm $\mathbf{P}^*$ (see Figure 3) takes two parameters, a and c, which control how it guesses loss bounds. We show later that a good choice for these parameters is $a = 1$ and $c = 2.618$. At the start of each iteration l of the outer loop, a bound on the best expert's remaining loss, k_l, is guessed. Algorithm $\mathbf{P}^*$ resets the experts' weights and uses Algorithm $\mathbf{P}(g(k_l/\ln N))$ (for the function g defined in Equation (7)) to generate predictions. If the bound k_l is correct then the remaining loss will be no greater than b_l. If the total loss incurred by Algorithm $\mathbf{P}$ during the iteration exceeds b_l, then the guessed bound on the loss of the best expert is incorrect, and Algorithm $\mathbf{P}^*$ proceeds to the next iteration of the outer loop.

Before analyzing Algorithm $\mathbf{P}^*$, we state a few simple facts that will be needed. First, from the description of the algorithm,

$$b_l = k_l + \sqrt{k_l \ln N} + \frac{\lg N}{2} = k_l + (ac^{l/2} + \frac{1}{2 \ln 2}) \ln N. \quad (8)$$

Also, since at most one unit of loss is incurred by any prediction, the loss incurred by Algorithm $\mathbf{P}^*$ during any iteration l of the outer loop is at most $b_l + 1$.

Lemma 10 *If Algorithm $\mathbf{P}^*$ exits iteration l of the outer loop then, for all $\mathcal{E}_i \in \mathcal{E}$, the loss incurred by $\mathcal{E}_i$ while Algorithm $\mathbf{P}^*$ is executing iteration l of the outer loop is greater than k_l.*

Proof: If some expert incurs loss at most k_l during the outer loop indexed by l, then Algorithm $\mathbf{P}$ has loss at most b_l during this iteration (by Theorem 5), and iteration l is not exited. ∎

Corollary 11 *If iteration l of the outer loop is exited when Algorithm $\mathbf{P}^*$ is running on sequence $\mathbf{y}$ then*

$$L_\mathcal{E}(\mathbf{y}) \geq L_\mathcal{E}(\mathbf{y}_1 \mathbf{y}_2 \ldots \mathbf{y}_l) > \sum_{j=1}^{l} k_j = a^2 \ln N \frac{c^{l+1} - 1}{c - 1}$$

where $\mathbf{y}_j$ is the sequence of instances seen during iteration j of the outer loop. Thus if last is the index of the last loop

entered, then loop last − 1 is exited and

$$L_{\mathcal{E}}(\mathbf{y}) \geq a^2 \ln N \frac{c^{last} - 1}{c - 1}.$$

Solving for last yields

$$last \leq \log_c \left(1 + \frac{L_{\mathcal{E}}(\mathbf{y})(c-1)}{a^2 \ln N}\right).$$

The above corollary shows that Algorithm $\mathbf{P}^*$ executes the outer loop a finite number of times whenever the loss of the best expert is bounded. Thus our bounds on Algorithm $\mathbf{P}^*$ hold even for infinite sequences, as long as the loss of the best expert is finite over the infinite sequence.

Theorem 12 *Let $\mathcal{E}$ be a set of N experts, and $\mathbf{y}$ be any sequence. If $L_{\mathcal{E}}(\mathbf{y})$ is finite then for all $a \geq 0$, the difference $L_{P^*}(\mathbf{y}) - L_{\mathcal{E}}(\mathbf{y})$ is at most*

$$\left(3.3302 + \frac{0.7675}{a} + \frac{1.064}{a \ln N}\right)\sqrt{L_{\mathcal{E}}(\mathbf{y}) \ln N} + a \ln N$$

when Algorithm $\mathbf{P}^*$ is given parameters $c = 2.618$ and a.

Note that the constant in front of the $\sqrt{L_{\mathcal{E}}(\mathbf{y}) \ln N}$ term can be made arbitrarily close to 3.3302 by choosing the constant a large enough.

Since the Algorithm $\mathbf{P}^*$ is not given the length of the sequence $\mathbf{y}$, the bound of Theorem 12 holds for all prefixes $\mathbf{y}$ of any infinite sequence $\mathbf{y}'$. Different experts might have minimum loss for different prefixes of $\mathbf{y}'$, but the loss of $\mathbf{P}^*$ is always close to the best expert on each prefix.

6 The hold-one-out model of prediction

We now turn to a slightly different prediction problem. As above, let $\mathbf{x} = x_1, \ldots, x_\ell$ be a sequence of instances chosen from an arbitrary set X, $\mathbf{y} = y_1, \ldots, y_\ell$ be a sequence of binary outcomes, and $\mathcal{E} = \{\mathcal{E}_1, \ldots, \mathcal{E}_N\}$ be a set of experts. In this section we will assume that each expert $\mathcal{E}_i$ is a function from X into $[0,1]$, i.e., the i^{th} expert's prediction at time t, denoted $\xi_{i,t}$, depends only on the instance x_t, and not on previous outcomes or instances. We call such experts *time independent*. As above, the total loss of the i^{th} expert is $L_{\mathcal{E}_i}(\mathbf{y}) = \sum_{t=1}^{\ell} |\xi_{i,t} - y_t|$, and the total loss of the best expert is $L_{\mathcal{E}}(\mathbf{y}) = \min_{1 \leq i \leq N} L_{\mathcal{E}_i}(\mathbf{y})$.

In *hold-one-out prediction*, the goal is still to predict almost as well as the best expert, but the prediction algorithm is allowed more information to help it make its predictions. In particular, when asked to predict the outcome y_t, the prediction algorithm is provided with all the instances $\mathbf{x} = x_1, \ldots, x_\ell$, the entire matrix $\xi_{i,t}$, $1 \leq i \leq N$, $1 \leq t \leq \ell$, giving the advice of each expert on each instance, and the outcomes $y_1, \ldots, y_{t-1}, y_{t+1}, \ldots, y_\ell$, i.e., all outcomes except y_t. Given this input, a hold-one-out prediction algorithm produces a prediction $\psi_t \in [0,1]$. The total loss of the hold-one-out prediction algorithm A on outcome sequence $\mathbf{y}$ is defined in analogy with the on-line prediction loss by $L_{\mathrm{A}}^H(\mathbf{y}) = \sum_{t=1}^{\ell} |\psi_t - y_t|$. This total loss can be viewed as the sum of the losses of ℓ *separate* runs of the algorithm, where in each run the algorithm is asked to predict a different outcome y_t.

It is clear that any on-line prediction strategy can also be used as a hold-one-out prediction strategy: the hold-one-out version of the strategy simply ignores the additional information available to it and makes its prediction of y_t based solely on the instances $x_1, \ldots, x_t$, the predictions of the experts on these instances, and the outcomes $y_1, \ldots, y_{t-1}$. In this case the total hold-one-out loss is the same as the total on-line loss. One might suppose, however, that significantly smaller hold-one-out losses could be obtained by employing more sophisticated strategies which take into account all the information that is available. Curiously, this is not true, at least in the worst case: the minimax value of the hold-one-out prediction problem is the same as that of the on-line problem given in Theorem 1.

Theorem 13 *For any length ℓ, any sequence $\mathbf{x} = x_1, \ldots, x_\ell$, and any set $\mathcal{E}$ of N time independent experts, the minimum over all hold-one-out prediction strategies A of the maximum over all binary outcome sequences $\mathbf{y} = y_1, \ldots, y_\ell$ of the difference $L_A^H(\mathbf{y}) - L_{\mathcal{E}}(\mathbf{y})$ is $\ell/2 - E_{\mathbf{y}'}(L_{\mathcal{E}}(\mathbf{y}'))$, where $E_{\mathbf{y}'}$ denotes expectation with respect to the uniform distribution over binary strings $\mathbf{y}'$ in $\{0,1\}^\ell$.*

Proof: The minimax value of the hold-one-out prediction problem is at most that of the on-line problem, since every on-line strategy is also a hold-one-out strategy with the same loss. Thus Theorem 1 shows that $L_{\mathrm{A}}^H(\mathbf{y}) - L_{\mathcal{E}}(\mathbf{y}) \leq \ell/2 - E_{\mathbf{y}'}(L_{\mathcal{E}}(\mathbf{y}'))$. To see that this bound is tight, consider the case when $\mathbf{y}$ is chosen at random, and note that the expectation of $L_{\mathrm{A}}^H(\mathbf{y}) - L_{\mathcal{E}}(\mathbf{y})$ is equal to the right-hand-side for any hold-one-out strategy. ∎

Theorem 14 *Let $\mathbf{P}$ be the on-line prediction algorithm defined in Section 4. For all ℓ and N, if β is chosen to be $g(\ell/2 \ln N)$, where g is as defined in (7), then for any $\mathbf{x}$, any set $\mathcal{E}$ of N time-independent experts, and any sequence $\mathbf{y}$ of length ℓ, the total hold-one-out loss of $\mathbf{P}$ is bounded by*

$$L_{P(\beta)}^H(\mathbf{y}) - L_{\mathcal{E}}(\mathbf{y}) \leq \sqrt{\frac{\ell \ln(|\mathcal{E}| + 1)}{2}} + \frac{\log_2(|\mathcal{E}| + 1)}{2}.$$

This is optimal in the sense that for all ℓ and N there exists a set of time-independent experts $\mathcal{E}$ such that for every prediction strategy A there is a sequence $\mathbf{y}$ where

$$L_A^H(\mathbf{y}) - L_{\mathcal{E}}(\mathbf{y}) \geq (1 - o(1))\sqrt{\frac{\ell \ln |\mathcal{E}|}{2}}$$

where $o(1) \to 0$ as $\ell, N \to \infty$.

When the value $L_{\mathcal{E}}(\mathbf{y})$ is given, we can use Algorithm $\mathbf{P}$ with an appropriately tuned β (as in Theorem 5) to get a good hold-one-out prediction algorithm. When neither this value nor the length of the sequence is available, Algorithm $\mathbf{P}^*$, which iteratively guesses the loss of the best expert, can be used. However, Algorithm $\mathbf{P}^*$ ignores the extra information provided and its bound has a factor greater than one multiplying the $\sqrt{L_{\mathcal{E}} \ln N}$ term. It is better to use the observed losses of the experts on the $\ell - 1$ outcomes provided to estimate $L_{\mathcal{E}}(\mathbf{y})$. Unfortunately, we are unable to show that when these estimates are plugged directly into Algorithm $\mathbf{P}$, a small total loss results. The problem is that different runs of the algorithm could use different values of β resulting in different predictions. Conceivably, the worst prediction in each run could be the one used to predict the held out label.

Our solution is to discretize the estimated total loss. A little randomization is used to ensure that the estimate is

Algorithm $\mathbf{P}^H(t)$:
{ The algorithm receives a sequence of instances, $\mathbf{x} = x_1, \ldots, x_\ell$, a sequence of binary outcomes, $\mathbf{y}_t = y_1, \cdots, y_{t-1}, y_{t+1}, \cdots, y_\ell$, and the predictions $\mathcal{E}_{i,j}$ of each expert $\mathcal{E}_i$ for $1 \le i \le N$ on each instance x_j for $1 \le j \le \ell$. The algorithm produces a prediction ψ_t for the held out outcome y_t. }

1. Pick $r \in [0,1]$ uniformly at random;
2. Compute $L_{\text{obs}} = \min_i \left(\sum_{j=1}^{t-1} |\mathcal{E}_{i,j} - y_j| + \sum_{j=t+1}^{\ell} |\mathcal{E}_{i,j} - y_j| \right)$;
3. Compute $L_{\text{est}} = (\lceil \sqrt{L_{\text{obs}} + 1} - r \rceil + r)^2$;
4. Run Algorithm $\mathbf{P}(g(L_{\text{est}}/\ln N))$ on the sequence of instances $x_1, \ldots, x_t$ and observations $y_1, \ldots, y_{t-1}$, and predict with the ψ_t (for y_t) generated by $\mathbf{P}$.

Figure 4: Description of Algorithm $\mathbf{P}^H$ for hold-one-out prediction.

likely to be the same regardless of which label is held out. The resulting algorithm is Algorithm $\mathbf{P}^H$, described in Figure 4. The square and square-root functions together with the ceiling increase the probability that all of the estimates are the same when the best expert's loss is large.

Theorem 15 *For algorithm $\mathbf{P}^H$, any $\mathbf{x}$, any set $\mathcal{E}$ of N time independent experts, and any sequence $\mathbf{y}$,*

$$L^H_{\mathbf{P}^H}(\mathbf{y}) - L_{\mathcal{E}}(\mathbf{y}) \le \sqrt{L_{\mathcal{E}}(\mathbf{y})}(\sqrt{\ln N}+1) + 3\sqrt{\ln N} + \frac{1}{\ln 2}\ln(N).$$

7 Relation to the PAC model for learning

We now give an application of these results to a PAC learning framework [31]. We look at a special variant of the PAC model in which nothing is assumed about the "target concept" that generates the examples other than independence between examples (sometimes referred to as *agnostic learning* [20]), and in which the learning algorithm is not required to return a hypothesis in any specific form.

Let X be any set and D any probability distribution on $X \times \{0,1\}$. When X is uncountable, appropriate assumptions are made to insure measurability. In our version of the PAC model, a sequence $\mathbf{s} = (x_1, y_1), \ldots, (x_\ell, y_\ell)$ of *training examples* is drawn from the product distribution D^ℓ (i.e. each example is drawn independently according to D) and a *PAC learning algorithm* A takes these training examples as input and outputs a function $h = A(\mathbf{s})$ that maps from X into $[0,1]$. The function h is called a *hypothesis*. The *error* of the hypothesis h is defined by $er_D(h) = E_{(x,y)\in D}|h(x) - y|$, where $E_{(x,y)\in D}$ denotes the expectation over (x,y) drawn randomly according to D. In PAC learning, the goal is to minimize this error under the worst-case distribution D. This leads to a kind of L^1 regression problem (see also Kearns and Schapire [19]).

The learning algorithm is given a set $\mathcal{H}$ of mappings from X into $[0,1]$, which is called a *hypothesis space*. These play a role similar to that played by the experts above. Namely, let $er_D(\mathcal{H}) = \inf_{h\in\mathcal{H}} er_D(h)$. Thus $er_D(\mathcal{H})$ is the error of the best hypothesis in $\mathcal{H}$ for the particular distribution D. In the agnostic version of PAC learning, the goal is to find a learning algorithm A that minimizes the maximum over all distributions D of $E_{\mathbf{s}\in D^\ell}(er_D(A(\mathbf{s}))) - er_D(\mathcal{H})$, where $E_{\mathbf{s}\in D^\ell}(\cdot)$ denotes expectation over sequences $\mathbf{s}$ of training examples drawn from the product distribution D^ℓ. Thus a good learning algorithm is one that, for any distribution D, produces a hypothesis that on average has error as close as possible to the best hypothesis in $\mathcal{H}$.[7]

While we cannot provide a minimax solution to this problem, we can use the algorithm developed in the previous section to get good upper bounds on the minimax value in certain important cases, better than those obtained by the only other methods that we are aware of [32, 30].[8] Before stating these bounds, we need to make a few definitions.

Our first definition deals with the issue of optimizing the error on the training examples (called *empirical error*) versus optimizing er_D, the error with respect to the underlying distribution D. This is often referred to as the problem of *overfitting*. Let

$$\widehat{er}_{\ell,D}(\mathcal{H}) = E_{\mathbf{s}\in D^\ell} \inf_{h\in\mathcal{H}} \frac{1}{\ell}\sum_{t=1}^{\ell} |h(x_t) - y_t|.$$

Thus $\widehat{er}_{\ell,D}(\mathcal{H})$ is the expected empirical error of the hypothesis in $\mathcal{H}$ that does best on a random set $\mathbf{s} = (x_1, y_1), \ldots, (x_\ell, y_\ell)$ of ℓ training examples drawn independently according to the distribution D. The quantity

$$er^{\Delta}_{\ell,D}(\mathcal{H}) = er_D(\mathcal{H}) - \widehat{er}_{\ell,D}(\mathcal{H})$$

will be called the *expected overfit* for ℓ training examples. It is clear that this quantity is nonnegative for any ℓ, D and $\mathcal{H}$, since

$$\begin{aligned} er_D(\mathcal{H}) &= \inf_{h\in\mathcal{H}} er_D(h) \\ &= \inf_{h\in\mathcal{H}} E_{\mathbf{s}\in D^\ell} \frac{1}{\ell}\sum_{t=1}^{\ell} |h(x_t) - y_t| \\ &\ge E_{\mathbf{s}\in D^\ell} \inf_{h\in\mathcal{H}} \frac{1}{\ell}\sum_{t=1}^{\ell} |h(x_t) - y_t| \\ &= \widehat{er}_{\ell,D}(\mathcal{H}). \end{aligned} \tag{9}$$

In other words, the expected empirical error of the best hypothesis on the training examples is always smaller than the expected error of the asymptotically best hypothesis on a set of random "test" examples.

Finally, for any hypothesis space $\mathcal{H}$ and sequence $\mathbf{x} = x_1, \ldots, x_\ell$, let us define

$$\mathcal{H}_{|\mathbf{x}} = \{(h(x_1), \ldots, h(x_\ell)) : h \in \mathcal{H}\}.$$

We will call $\mathcal{H}_{|\mathbf{x}}$ the *restriction of $\mathcal{H}$ to $\mathbf{x}$*.

[7] Typically, tail bounds are also given that bound the probability that the hypothesis returned is significantly worse than the best hypothesis in $\mathcal{H}$. Our current methods do not provide these, but standard "confidence boosting" methods can be applied on top of them to achieve good tail bounds [14, 21]. More direct methods are given by Littlestone and Warmuth [23].

[8] Bounds given by Vapnik ([32], Equation (11)) imply a bound in the same form as the second bound in Theorem 16, but with an additional factor of 2 in the leading term. However, these bounds do hold in more general cases than the one we consider here. Talagrand gives similar general bounds without the factor of 2, but with an unspecified constant K in the lower order term. It is not clear that the constant K can be made small enough to get practical bounds for small sample size ℓ.

Theorem 16 *For any instance space X and any hypothesis space $\mathcal{H}$ on X, there exists a PAC learning strategy A such that for all ℓ and all distributions D on $X \times \{0,1\}$*

$$E_{\mathbf{s}\in D^\ell}(\mathrm{er}_D(A(\mathbf{s}))) - \mathrm{er}_D(\mathcal{H}) \leq$$

$$\frac{E_{\mathbf{x}}\sqrt{\ln(|\mathcal{H}_{|\mathbf{x}}|+1)}}{\sqrt{2(\ell+1)}} + \frac{E_{\mathbf{x}}(\log_2(|\mathcal{H}_{|\mathbf{x}}|+1))}{2(\ell+1)} - \mathrm{er}^{\Delta}_{\ell+1,D}(\mathcal{H}),$$

where $E_{\mathbf{x}}$ denotes expectation over $\mathbf{x} = x_1, \ldots x_{\ell+1}$, each x_t drawn independently at random according to the marginal of D on X.

There also exists a PAC learning strategy A such that for all ℓ and all distributions D on $X \times \{0,1\}$

$$\begin{aligned} & E_{\mathbf{s}\in D^\ell}(\mathrm{er}_D(A(\mathbf{s}))) - \mathrm{er}_D(\mathcal{H}) \\ \leq\ & \frac{\sqrt{\widehat{\mathrm{er}}_{\ell+1,D}(\mathcal{H})}(\sqrt{V}+1)}{\sqrt{\ell+1}} + \frac{V/\ln 2 + 3\sqrt{V}}{\ell+1} - \mathrm{er}^{\Delta}_{\ell+1,D}(\mathcal{H}) \\ \leq\ & \frac{\sqrt{\mathrm{er}_D(\mathcal{H})}(\sqrt{V}+1)}{\sqrt{\ell+1}} + \frac{V/\ln 2 + 3\sqrt{V}}{\ell+1} - \mathrm{er}^{\Delta}_{\ell+1,D}(\mathcal{H}), \end{aligned}$$

where $V = E_{\mathbf{x}} \ln |\mathcal{H}_{|\mathbf{x}}|$.

Using the results from the previous section, the proof of this theorem rests on the following lemma.

Lemma 17 *Let A be a hold-one-out prediction strategy. Then A can be converted into a PAC learning strategy B such that for any hypothesis space $\mathcal{H}$, any ℓ, and any distribution D on $X \times \{0,1\}$,*

$$E_{\mathbf{s}\in D^\ell}(\mathrm{er}_D(B(\mathbf{s}))) - \mathrm{er}_D(\mathcal{H}) =$$

$$\frac{1}{\ell+1} E_{(\mathbf{x},\mathbf{y})\in D^{\ell+1}}\left(L^H_A(\mathbf{y}) - L_{\mathcal{H}_{|\mathbf{x}}}(\mathbf{y})\right) - \mathrm{er}^{\Delta}_{\ell+1,D}(\mathcal{H}),$$

where $E_{(\mathbf{x},\mathbf{y})\in D^{\ell+1}}$ denotes expectation over $\mathbf{x} = x_1, \ldots x_{\ell+1}$ and $\mathbf{y} = y_1, \ldots, y_{\ell+1}$, each (x_t, y_t) drawn independently at random according to D, $1 \leq t \leq \ell+1$.

Sketch of proof of Lemma 17:

The PAC learning strategy B works as follows. For any sequence of examples $\mathbf{s} = (x_1, y_1), \ldots, (x_\ell, y_\ell)$ and any instance x, the value of the function $h = B(\mathbf{s})$ on input x is defined by inserting x into a random position in the sequence $\mathbf{s}$ and running the hold-one-out prediction strategy A to predict the label of x, using the hypotheses in $\mathcal{H}$ as experts [17]. More formally, $h(x) = \frac{1}{\ell+1}\sum_{t=1}^{\ell+1} \psi_t$, where ψ_t is the output of A when A is given as input the sequence of instances $\mathbf{x} = x_1, \ldots, x_{t-1}, x, x_t, \ldots, x_\ell$, the set $\mathcal{H}_{|\mathbf{x}}$ of experts, and the observed outcomes $\mathbf{y} = y_1, \ldots, y_{t-1}, ?, y_t, \ldots, y_\ell$, where '?' denotes the location of the missing outcome to be predicted. It can be shown that the strategy B has the desired performance; details are omitted. ∎

Sketch of proof of Theorem 16:

The first bound follows directly from Theorem 14, using the above lemma, with A being the hold-one-out prediction strategy from Theorem 14. The second and third bounds follow from Theorem 15 and the above lemma, with A being the hold-one-out prediction strategy from Theorem 15, using the Cauchy-Schwarz inequality. ∎

It is easy to see that the constants in the leading terms of the bounds in Theorem 16 are the best possible. The argument is similar to our previous lower bound arguments. We assume that the distribution D is such that for a random example (x, y), the value y is 1 with probability 1/2 and 0 with probability 1/2, independent of x. Hence, every hypothesis h has $\mathrm{er}_D(h) = 1/2$. This implies that $E_{\mathbf{s}\in D^\ell}(\mathrm{er}_D(A(\mathbf{s}))) - \mathrm{er}_D(\mathcal{H}) = 0$ for any hypothesis space H and algorithm A. On the other hand, suppose that the N hypotheses in H are chosen randomly such that they predict 1 with probability 1/2 and 0 with probability 1/2 on a random instance x (this is not hard to arrange). Then Lemma 8 implies that the expected overfit $\mathrm{er}^{\Delta}_{\ell+1,D}(H)$ is $(1+o(1))\frac{\sqrt{\ln N}}{\sqrt{2\ell}}$. The expected overfit appears with a minus sign on the right hand side of the first bound in Theorem 16. Hence for this bound to be nonnegative, as required in this case, the constant in the first term on the right hand side must be at least $(1+o(1))/\sqrt{2}$. This shows that this constant cannot be improved in general. The same is true for the leading constant in the other bounds of Theorem 16.

8 Extensions

We are currently considering several extensions of these results. One issue is the use of other loss functions. Since a prediction strategy defines a conditional probability distribution on the next bit given the values of the previous bits, a natural choice of loss function is the information gained by seeing the next bit, with respect to this conditional distribution. Hence if the strategy predicts $y_t = 1$ with probability ψ_t and $y_t = 0$ with probability $1 - \psi_t$, then the loss at time t will be $-\log \psi_t$ if $y_t = 1$ and $-\log(1-\psi_t)$ if $y_t = 0$. We call this *log loss*. The nice thing about the log loss is that for any prediction strategy A, the total log loss on $y_1, \ldots y_\ell$, denoted $L_A(\mathbf{y})$, is the total information gained from the sequence $\mathbf{y}$, under the conditional distributions represented by A. Moreover, any distribution on $\{0,1\}^\ell$ induces a conditional distribution on the t^{th} bit given any values for the previous $t-1$ bits for all $1 \leq t \leq \ell$, and hence defines a prediction strategy. Conversely, any prediction strategy A on $\{0,1\}^\ell$ defines a probability distribution P_A on $\{0,1\}^\ell$. It is easy to see that for the log loss, $L_A(\mathbf{y}) = -\log P_A(\mathbf{y})$.

It is well known that for the log loss, for any set $\mathcal{E}$ of N experts (i.e., distributions) there is a prediction strategy A such that for any sequence $\mathbf{y}$, $L_A(\mathbf{y}) - L_{\mathcal{E}}(\mathbf{y}) \leq \log N$, where $L_{\mathcal{E}}(\mathbf{y})$ is the total log loss of the best expert for $\mathbf{y}$ [26, 6, 36, 13, 37]. The strategy is just the Bayes algorithm with uniform prior on the distributions represented by the experts. We have done an exact minimax analysis of this case as well, and the result is quite simple. For each $\mathbf{y} \in \{0,1\}^\ell$ and each expert $\mathcal{E}_i \in \mathcal{E}$, let $P_i(\mathbf{y})$ denote the probability of $\mathbf{y}$ under expert $\mathcal{E}_i$. Define the probability of $\mathbf{y}$ for the algorithm A by

$$P_A(\mathbf{y}) = \frac{\max_{1\leq i\leq N} P_i(\mathbf{y})}{\sum_{\mathbf{y}\in\{0,1\}^\ell} \max_{1\leq i\leq N} P_i(\mathbf{y})}.$$

Then A minimizes the maximum of the difference $L_A(\mathbf{y}) - L_{\mathcal{E}}(\mathbf{y})$ over all sequences $\mathbf{y}$. Furthermore, this difference is the same for all sequences $\mathbf{y}$:

$$L_A(\mathbf{y}) - L_{\mathcal{E}}(\mathbf{y}) = \log \sum_{\mathbf{y}\in\{0,1\}^\ell} \max_{1\leq i\leq N} P_i(\mathbf{y}) \leq \log N.$$

Other topics we are investigating focus on extensions of the results to the case when the set of experts is infinite, and to the case when the outcome is real-valued rather than binary-valued. In both these cases, results by Littlestone,

Long and Warmuth [23, 22] can be applied to extend the methods we have presented here. We have also obtained some related results for the quadratic loss in these cases (see also Vovk [34]).

Acknowledgements

We would like to thank Meir Feder, Yuval Peres, Nick Littlestone and Michael Kearns for helpful suggestions and discussions of this material.

References

[1] T. M. Cover. Behaviour of sequential predictors of binary sequences. In *Transactions of the Fourth Prague Conference on Information Theory, Statistical Decision Functions, Random Processes*, pages 263–272. Publishing House of the Czechoslovak Academy of Sciences, 1965.

[2] T. M. Cover and A. Shanhar. Compound Bayes predictors for sequences with apparent Markov structure. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-7(6):421–424, June 1977.

[3] A. Dawid. Prequential data analysis. *Current Issues in Statistical Inference*, to appear.

[4] A. P. Dawid. Statistical theory: The prequential approach. *Journal of the Royal Statistical Society, Series A*, pages 278–292, 1984.

[5] A. P. Dawid. Prequential analysis, stochastic complexity and Bayesian inference. *Bayesian Statistics 4*, to appear.

[6] A. DeSantis, G. Markowski, and M. N. Wegman. Learning probabilistic prediction functions. In *Proceedings of the 1988 Workshop on Computational Learning Theory*, pages 312–328. Morgan Kaufmann, 1988.

[7] M. Feder, N. Merhav, and M. Gutman. Universal prediction of individual sequences. *IEEE Transactions on Information Theory*, 38:1258–1270, 1992.

[8] A. Fiat, D. Foster, H. Karloff, Y. Rabani, Y. Ravid, and S. Vishwanathan. Competitive algorithms for layered graph traversal. In *32nd Annual Symposium on Foundations of Computer Science*, pages 288–297, 1991.

[9] A. Fiat, R. Karp, M. Luby, L. McGeoch, D. Sleator, and N. Young. Competitive paging algorithms. *Journal of Algorithms*, 12:685–699, 1991.

[10] A. Fiat, Y. Rabani, and Y. Ravid. Competitive k-server algorithms. In *31st Annual Symposium on Foundations of Computer Science*, pages 454–463, 1990.

[11] J. Galambos. *The Asymptotic Theory of Extreme Oreder Statistics*. R. E. Kreiger, second edition, 1987.

[12] J. Hannan. Approximation to Bayes risk in repeated play. In *Contributions to the theory of games*, volume 3, pages 97–139. Princeton University Press, 1957.

[13] D. Haussler and A. Barron. How well do Bayes methods work for on-line prediction of $\{+1, -1\}$ values? In *Proceedings of the Third NEC Symposium on Computation and Cognition*. SIAM, to appear.

[14] D. Haussler, M. Kearns, N. Littlestone, and M. K. Warmuth. Equivalence of models for polynomial learnability. *Information and Computation*, 95:129–161, 1991.

[15] D. Haussler, M. Kearns, and R. Schapire. Bounds on the sample complexity of Bayesian learning using information theory and the VC dimension. *Machine Learning*, to appear.

[16] D. Haussler, N. Littlestone, and M. Warmuth. Predicting $\{0,1\}$-functions on randomly drawn points. Technical Report UCSC-CRL-90-54, University of California Santa Cruz, Computer Research Laboratory, Dec. 1990. To appear, *Information and Computation*.

[17] D. Helmbold and M. K. Warmuth. On weak learning. In *Proceedings of the Third NEC Research Symposium on Computational Learning and Cognition*. SIAM, to appear.

[18] D. P. Helmbold and M. K. Warmuth. Some weak learning results. In *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, pages 399–412, 1992.

[19] M. J. Kearns and R. E. Schapire. Efficient distribution-free learning of probabilistic concepts. In *31st Annual Symposium on Foundations of Computer Science*, pages 382–391, 1990.

[20] M. J. Kearns, R. E. Schapire, and L. M. Sellie. Toward efficient agnostic learning. In *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, pages 341–352, 1992.

[21] N. Littlestone. From on-line to batch learning. In *Proceedings of the Second Annual Workshop on Computational Learning Theory*, pages 269–284. Morgan Kaufmann, 1989.

[22] N. Littlestone, P. M. Long, and M. K. Warmuth. On-line learning of linear functions. In *Proceedings of the Twenty Third Annual ACM Symposium on Theory of Computing*, pages 465–475, 1991.

[23] N. Littlestone and M. Warmuth. The weighted majority algorithm. In *30th Annual IEEE Symposium on Foundations of Computer Science*, pages 256–261, 1989. Long version: UCSC tech. rep. UCSC-CRL-91-28.

[24] N. Merhav and M. Feder. Universal sequential learning and decision from individual data sequences. In *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, pages 413–427, 1992.

[25] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[26] J. Rissanen. Stochastic complexity and modeling. *The Annals of Statistics*, 14(3):1080–1100, 1986.

[27] J. Rissanen and G. G. Langdon, Jr. Universal modeling and coding. *IEEE Transactions on Information Theory*, IT-27(1):12–23, Jan. 1981.

[28] H. S. Seung, H. Sompolinsky, and N. Tishby. Statistical mechanics of learning from examples. *Physical Review A*, 45(8):6056–6091, 1992.

[29] H. Sompolinsky, N. Tishby, and H. Seung. Learning from examples in large neural networks. *Physical Review Letters*, 65:1683–1686, 1990.

[30] M. Talagrand. Sharper bounds for Gaussian and empirical processes. *Annals of Probability*, to appear.

[31] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–42, 1984.

[32] V. Vapnik. Principles of risk minimization for learning theory. In J. E. Moody, S. J. Hanson, and R. P. Lippman, editors, *Advances in Neural Information Processing Systems 4*. Morgan Kaufmann, 1992.

[33] V. N. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, 1982.

[34] V. G. Vovk. Aggregating strategies. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 371–383. Morgan Kaufmann, 1990.

[35] V. G. Vovk. Prequential probability theory. Unpublished manuscript, 1990.

[36] V. G. Vovk. Universal forcasting algorithms. *Information and Computation*, 96(2):245–277, Feb. 1992.

[37] K. Yamanishi. A loss bound model for on-line stochastic prediction strategies. In *Proceedings of the Fourth Annual Workshop on Computational Learning Theory*, pages 290–302. Morgan Kaufmann, 1991.