How to Use Expert Advice

NICOLÒ CESA-BIANCHI

Università di Milano, Milan, Italy

YOAV FREUND

AT&T Labs, Florham Park, New Jersey

DAVID HAUSSLER AND DAVID P. HELMBOLD

University of California, Santa Cruz, Santa Cruz, California

ROBERT E. SCHAPIRE

AT&T Labs, Florham Park, New Jersey

AND

MANFRED K. WARMUTH

University of California, Santa Cruz, Santa Cruz, California

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 then show how this leads to certain kinds of pattern recognition/learning algorithms with performance bounds that improve on the best results currently

This research was done while N. Cesa-Bianchi was visiting UC, Santa Cruz, and was partially supported by the "Progetto finalizzato sistemi informatili e calcolo parallelo" of CNR under grant 91.00884.69.115.09672."

D. Haussler, M. K. Warmuth, and Y. Freund were supported by ONR grant N00014-91-J-1162 and NSF grant IRI 91-23692.

Authors' addresses: N. Cesa-Bianchi, Università di Milano, Milan, Italy, e-mail: cesabian@dsi.unimi.it; Y. Freund and R. E. Schapire, AT&T Labs, 180 Park Ave., Florham Park, NJ 07932-0971, e-mail: {yoav, schapire}@research.att.com; D. Haussler, D. P. Helmbold, and M. K. Warmuth, University of California, Santa Cruz, Santa Cruz, CA.

known in this context. We also compare our analysis to the case in which log loss is used instead of the expected number of mistakes.

Categories and Subject Descriptors: I.2.1 **[Artificial Intelligence]:** Applications and Expert Systems; I.2.2 **[Artificial Intelligence]:** Automatic Programming–*automatic analysis of algorithms*; I.2.6 **[Artificial Intelligence]:** Learning–*knowledge acquisition*

General Terms: Algorithms

1. *Introduction*

A central problem in statistics and 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 (e.g., [Haussler et al. 1994]). 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 that have been explored in the information theory literature. We then give applications of these results to the theory of pattern recognition [Vapnik 1982] and PAC learning [Valiant 1984].

We take the extreme position, as advocated by Dawid and Vovk in the theory of prequential probability [Dawid 1984; 1991; 1996; Vovk 1993], Rissanen in his theory of stochastic complexity [Rissanen 1978; 1986; Rissanen and Langdon Jr. 1981; Yamanishi 1995], and Cover, Lempel and Ziv, Feder and others in the theory of universal prediction and data compression of individual sequences,[1] 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 making your prediction you are allowed to hear the predictions of a (fixed) finite set $\mathscr{E} = \{\mathscr{E}_1, \ldots, \mathscr{E}_N\}$ 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 $x_1, \ldots, x_t$ 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

[1] See, for example, Feder et al. [1992], Merhav and Feder [1993], Cover [1965], Cover and Shanar [1977], Hannan [1957], Vovk [1993], and Chung [1994].

measure of loss,[2] 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.

Imagine that the above prediction game is played for ℓ days. 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 $\mathbf{y} = y_1, \ldots, y_\ell$. During the ℓ days, you accumulate a total loss $L(\mathbf{y}) = \ {}_{t=1}^{\ell} |\hat{y}_t - y_t|$, where $\hat{y}_t \in [0, 1]$ 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 outcome sequence* **y** *is produced by nature*.[3] Specifically, if we let $L_{\mathscr{E}}(\mathbf{y})$ denote the minimum total loss of any expert on the particular sequence **y**, then your goal is to minimize the maximum of the difference $L(\mathbf{y}) - L_{\mathscr{E}}(\mathbf{y})$ over all possible binary sequences **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 and any prespecified sequence length ℓ, there is a strategy that minimizes the maximum of the difference $L(\mathbf{y}) - L_{\mathscr{E}}(\mathbf{y})$ over all possible binary outcome sequences **y** of length ℓ. While this min/max strategy can be implemented in some cases, it is not practical in general. However, we define an algorithm, called **P** for "Predict", that is simple and efficient, and performs essentially as well as the min/max strategy. Actually **P** is a family of algorithms that is related to the algorithm studied by Vovk [1990] and the Bayesian, Gibbs, and "weighted majority" methods studied by a number of authors,[4] as well as the method developed by Feder et al. [1992]. We show that **P** performs quite well in the sense defined above so that, for example, given any finite set $\mathscr{E}$ of weather forecasting experts, **P** is guaranteed not to perform much worse than the best expert in $\mathscr{E}$, no matter what the actual weather turns out to be. The algorithm **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.

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

[3] 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 [Fiat et al. 1991a; 1991b; 1994], except that we look at the difference in performance rather than the ratio.

[4] See, for example, Littlestone and Warmuth [1994], Littlestone et al. [1995], Haussler et al. [1994], Sompolinsky et al. [1992], Seung et al. [1992], Haussler and Barron [1992], and Hembold and Warmuth [1995].

In particular, letting $L_P(\mathbf{y})$ denote the total loss of algorithm **P** on the sequence **y** and $L_{\mathscr{E}}(\mathbf{y})$ the loss of the best expert on **y** as above, we show (Theorem 4.5.1) that for all binary[5] outcome sequences **y** of length ℓ,

$$L_P(\mathbf{y}) - L_{\mathscr{E}}(\mathbf{y}) \le \sqrt{\frac{\ell \ln(|\mathscr{E}| + 1)}{2}} + \frac{\log_2(|\mathscr{E}| + 1)}{2},$$

and that no algorithm can improve the multiplicative constant of the square-root term for $|\mathscr{E}|$, $\ell \to \infty$, where $|\mathscr{E}|$ is the number of experts.

Previous work has shown how to construct an algorithm A such that the ratio $L_A(\mathbf{y})/L_{\mathscr{E}}(\mathbf{y})$ approaches 1 in the limit [Vovk 1990; Littlestone and Warmuth 1994; Feder et al. 1992]. In fact, Vovk [1990] described an algorithm with the same bound as the one we give in Theorem 4.2.1 for the algorithm **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 $L_A(\mathbf{y}) - L_{\mathscr{E}}(\mathbf{y})$. 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 L_1 distance between a random binary string uniformly chosen from $\{0, 1\}^{\ell}$ and a set of N points in $[0, 1]^{\ell}$. This answer to a basic combinatorial question may be of independent interest.

The remainder of this paper is organized as follows: In Section 3, we characterize exactly the performance of the best possible prediction strategy using a min/max analysis. Section 4 describes the algorithm **P** and shows that it achieves the optimal bound given above. In Section 4.4, we show that, if the loss $L_{\mathscr{E}}(\mathbf{y})$ of the best expert is given to the algorithm *a priori*, then **P** can be tuned so that

$$L_P(\mathbf{y}) - L_{\mathscr{E}}(\mathbf{y}) \le \sqrt{L_{\mathscr{E}}(\mathbf{y}) \ln|\mathscr{E}|} + \frac{\log_2 |\mathscr{E}|}{2}.$$

In Section 4.6, we show that even when no knowledge of $L_{\mathscr{E}}(\mathbf{y})$ is available, one can use a doubling trick to obtain a bound on $L_P(\mathbf{y}) - L_{\mathscr{E}}(\mathbf{y})$ that is only a small constant factor larger than the above bound. This algorithm can nearly match the performance of the best expert on all prefixes of an infinite sequence **y**.

Finally, in Section 5, we show how the results we have obtained can be applied in another machine learning context. We describe a pattern recognition problem 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 instances and the goal is to find a function that will predict the binary label y_t of the next random example (x_t, y_t) correctly. Performance is measured relative to the best binary-valued function in a given class of functions, called the *comparison* class. This kind of relative performance measure is called *regret* in statistics. General solutions to this regret formulation of the pattern recognition problem have been developed by Vapnik [1982], Birge and Massart [1993], and others. This problem can also be described as a special variant of the *probably approxi-*

[5] The algorithm has recently been extended to the case when the outcomes are in the interval [0, 1] with the performance bounds as in the binary case [Haussler et al. 1995].

mately correct (PAC) learning model [Valiant 1984] in which nothing is assumed about the "target concept" that generates the examples other than independence between examples (sometimes referred to as *agnostic learning* [Kearns et al. 1994]), and in which the learning algorithm is not required to return a hypothesis in any specific form. Using the prediction strategy **P**, we develop an algorithm that solves this pattern recognition problem and derive distribution-independent bounds for the performance of this algorithm. These bounds improve by constant factors some of the (more general) bounds obtained by Vapnik [1982] and Talagrand [1994] 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, have worst-case counterparts that 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. *An Overview of the Prediction Problem*

In this section, we define the problem of predicting binary sequences and give an overview of our results on this problem.

We refer to the binary sequence to be predicted as the *outcome sequence*, and we denote it by $\mathbf{y} = y_1, \ldots, y_t, \ldots, y_\ell$, where t is the index of a typical time step or trial, $y_t \in \{0, 1\}$, and ℓ is the length of the sequence. We denote by $\mathbf{y}_t$ the prefix of length t of $\mathbf{y}$, that is, $\mathbf{y}_t = y_1, \ldots, y_t$.

We denote the set of experts by $\mathscr{E} = \{\mathscr{E}_1, \ldots, \mathscr{E}_N\}$, where N is the number of experts. The prediction of expert $\mathscr{E}_i$ at time t is denoted by $\xi_{i,t} \in [0, 1]$ and the prediction of the algorithm at time t is denoted by $\hat{y}_t \in [0, 1]$.

A *prediction algorithm* is an algorithm that at time $t = 1, \ldots, \ell$, receives as input a vector of expert predictions $\langle \xi_{1,t}, \ldots, \xi_{N,t} \rangle$, as well as the predictions made by the experts in the past (i.e., $\langle \xi_{1,1}, \ldots, \xi_{N,1} \rangle, \ldots, \langle \xi_{1,t-1}, \ldots, \xi_{N,t-1} \rangle$), the sequence of past outcomes (i.e., $\mathbf{y}_{t-1}$), and the predictions made by the algorithm in the past (i.e., $\hat{y}_1 \cdots \hat{y}_{t-1}$). The prediction algorithm maps these inputs into its current prediction $\hat{y}_t$.

The loss of prediction algorithm A on a sequence of trials with respect to a sequence of outcomes $\mathbf{y}$ (and set of experts) is defined to be the sum ${}_{t=1}^{\ell} |\hat{y}_t - y_t|$ which is denoted $L_A(\mathbf{y})$. Note that the set of experts will always be understood from context so we can suppress the dependence of $L_A(\mathbf{y})$ on $\mathscr{E}$. Similarly, the loss of expert $\mathscr{E}_i$ with respect to $\mathbf{y}$ is defined to be ${}_{t=1}^{\ell} |\xi_{i,t} - y_t|$ and is denoted $L_{\mathscr{E}_i}(\mathbf{y})$. Finally, the loss of the best expert is denoted by $L_{\mathscr{E}}(\mathbf{y})$; thus, $L_{\mathscr{E}}(\mathbf{y}) = \min_{i=1,\ldots,N} L_{\mathscr{E}_i}(\mathbf{y})$.

Our goal is to find algorithms whose loss $L_A(\mathbf{y})$ is not much larger than $L_{\mathscr{E}}(\mathbf{y})$. Moreover, our ultimate goal is to prove bounds that hold uniformly for all outcome sequences and expert predictions, and that assume little or no prior knowledge on the part of the prediction algorithm.

This problem can be viewed as a game in which the predictor plays against an adversary who generates both the experts' predictions and the outcomes. We assume that both players can observe all of the actions made by the other player up to the current point of time, as well as its own past actions. The game consists of ℓ time steps, and both sides know ℓ before the game begins. We now describe the *binary sequence prediction game*. At each time step, $t = 1 \cdots \ell$, the game proceeds as follows:

—The adversary chooses the experts' predictions, $\xi_{i,t} \in [0, 1]$, for $1 \le i \le N$.

—The predictor generates its prediction $\hat{y}_t \in [0, 1]$.

—The adversary chooses the outcome $y_t \in \{0, 1\}$.

The goal of the predictor in this game is to minimize its *net loss*: $L_A(\mathbf{y})$–$L_{\mathscr{E}}(\mathbf{y})$. The goal of the adversary is to maximize this value.[6] The min/max value for this game, is the worst case net loss of the optimal prediction strategy. We will denote this min/max value by $V_{N,\ell}$.

In the following section, we give the optimal min/max strategy for the predictor and for the adversary in this game. This analysis gives a simple recursive equation for $V_{N,\ell}$. Unfortunately, we don't have a closed form expression that solves this equation. However, using results obtained in Sections 3 and 4, we can show that

$$V_{N,\ell} = (1 + o(1)) \sqrt{\frac{\ell \ln N}{2}},$$

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

In Section 3.1, we analyze the optimal prediction algorithm for a case in which the adversary is somewhat restricted. Using this restriction of the game we find an explicit closed form expression that lower bounds $V_{N,\ell}$. The adversary is restricted in that the predictions of the experts are functions only of the trial number. In other words, each expert is a fixed sequence of ℓ numbers in [0, 1]. We call these *static* experts. We also assume that these sequences are known to the predictor in advance. We derive the exact min/max solution for this restricted game for any choice of the sequences. We obtain our explicit lower bound by analyzing the case in which the N expert sequences are chosen using independent coin flips.

In Section 4, we present a family of prediction algorithms for the general prediction game. The basic algorithm, which we call **P** has a real-valued parameter, β, which controls its behavior. This parameter plays a similar role to the "learning rate" parameter used in gradient based learning algorithms [Haykin 1994]. Different choices of β guarantee different performance bounds for the algorithm. The optimal choice of β is of critical importance and occupies much of the discussion in Sections 4.4–4.6 and also later in Section 5.4.

[6] Formally, an expert in this context is a function of the form $\mathscr{E}_i : ([0, 1] \times \{0, 1\})^* \to [0, 1]$. The interpretation here is that $\mathscr{E}_i$ maps a finite sequence $((\hat{y}_1, y_1), \ldots, (\hat{y}_{t-1}, y_{t-1}))$ of prediction/outcome pairs to a new expert prediction $\xi_{i,t}$. (Note that each $\mathscr{E}_i$ function can compute the value of the other $\mathscr{E}_j$ functions, and thus the experts' predictions can depend on the predictions made by experts in the past, as well as the current time t.)

We analyze three variants of the algorithm, each of which chooses β in a different way, according to the type of knowledge available to the predictor. The first variant chooses β when the predictor knows only an upper bound on the loss of the best expert. The second variant chooses β in a situation where the predictor knows only the length ℓ of the game. The third variant handles the case where the predictor knows nothing at all in advance. Using the analysis of the second case, we get an upper bound for $V_{N,\ell}$ that asymptotically matches the lower bound from Section 3.1.

3. *An Optimal Prediction Strategy*

We now give the optimal prediction algorithm for the binary sequence prediction problem. This algorithm is based on the optimal min/max solution of the binary sequence prediction game described in the previous section, guaranteeing that it has the best possible worst-case performance. However, the algorithm is computationally expensive.

The following function plays a major role in the construction and analysis of the optimal prediction strategy. Let $\mathbb{R}^+$ denote the nonnegative reals, and $\mathbb{N}$ denote the nonnegative integers. We define the function $v : (\mathbb{R}^+)^N \times \mathbb{N} \rightarrow \mathbb{R}^+$ inductively as follows:

$$v(M, 0) = \min_{1 \le i \le N} (M_i) \tag{1}$$

$$v(M, r) = \min_{Z \in [0,1]^N} \frac{v(M + Z, r - 1) + v(M + \mathbf{1} - Z, r - 1)}{2} \tag{2}$$

where the $\mathbf{1}$ in the expression $M + \mathbf{1} - Z$ denotes the vector of N $\mathbf{1}$'s, and M_i is the ith component of vector M. Clearly, this function is well defined and can, in principle, be calculated for any given M and r. We will discuss the complexity of this computation after the proof of Theorem 3.2.

The parameters of the function v are interpreted as follows: The integer r denotes the number of remaining trials, that is, the number of sequence bits that remain to be predicted. The past loss incurred by the expert $\mathscr{E}_i$ when there are r remaining trials will be denoted M_i^r, and M^r will denote the vector $\langle M_1^r, \ldots, M_N^r \rangle$. It is the quantity $v(M^r, r)$ that will be important in our analysis. In some sense, $v(M^r, r)$ is measuring the anticipated loss of the best expert on the entire sequence of trials.

In order to show that our prediction strategy generates predictions that are in the range [0, 1], we will need the following lemma, which shows that the function $v(M, r)$ obeys a Lipschitz condition:

LEMMA 3.1. *For any $r \in \mathbf{N}$ and any $X, Y \in (\mathbb{R}^+)^N$*

$$|v(X, r) - v(Y, r)| \le \|X - Y\|_\infty,$$

where $\|X - Y\|_\infty = max_i|X_i - Y_i|$.

PROOF. The proof is by induction on r:

If $r = 0$, let i_0 be an index that minimizes $\{X_i\}$ and j_0 be an index that minimizes $\{Y_i\}$. Then

$$v(X, 0) - v(Y, 0) = X_{i_0} - Y_{j_0} \leq X_{j_0} - Y_{j_0} \leq \|X - Y\|_\infty.$$

Now suppose $r > 0$ and let us assume that the lemma holds for $r - 1$. Let $Z_0 \in [0, 1]^N$ be a vector that minimizes

$$v(Y, r) = \min_{Z \in [0,1]^N} \frac{v(Y + Z, r - 1) + v(Y + \mathbf{1} - Z, r - 1)}{2}.$$

We get:

$$\begin{aligned}
&v(X, r) - v(Y, r) \\
&= \min_{Z \in [0,1]^N} \frac{v(X + Z, r - 1) + v(X + \mathbf{1} - Z, r - 1)}{2} \\
&\quad - \min_{Z \in [0,1]^N} \frac{v(Y + Z, r - 1) + v(Y + \mathbf{1} - Z, r - 1)}{2} \\
&\leq \frac{v(X + Z_0, r - 1) + v(X + \mathbf{1} - Z_0, r - 1)}{2} \\
&\quad - \frac{v(Y + Z_0, r - 1) + v(Y + \mathbf{1} - Z_0, r - 1)}{2} \\
&= \frac{v(X + Z_0, r - 1) - v(Y + Z_0, r - 1)}{2} \\
&\quad + \frac{v(X + \mathbf{1} - Z_0, r - 1) - v(Y + \mathbf{1} - Z_0, r - 1)}{2} \\
&\leq \frac{\|(X + Z_0) - (Y + Z_0)\|_\infty}{2} + \frac{\|(X + \mathbf{1} - Z_0) - (Y + \mathbf{1} - Z_0)\|_\infty}{2} = \|X - Y\|_\infty
\end{aligned}$$

where the last inequality follows from our inductive hypothesis. □

We now define the prediction strategy **MM** and then prove a theorem showing that this is the optimal prediction strategy. The prediction strategy (see Figure 1) works as follows: On trial t, let $r = \ell - t + 1$ be the number of bits that remain to be predicted, M^r be the vector representing the loss of each of the experts on the sequence seen so far, and Z^r be the vector of current expert predictions, that is, $Z^r = \langle \xi_{1,t}, \ldots, \xi_{N,t} \rangle$. The prediction strategy sets its prediction to be

$$\hat{y}_t = \frac{v(M^r + Z^r, r - 1) - v(M^r + \mathbf{1} - Z^r, r - 1) + 1}{2}. \tag{3}$$

As $\|(M^r + Z^r) - (M^r + \mathbf{1} - Z^r)\|_\infty \leq 1$, we get from Lemma 3.1 that $0 \leq \hat{y}_t \leq 1$; thus, this prediction formula always generates legitimate predictions.

```
Algorithm MM

1. Initialize:
    • $t := 1$      {current trial number}
    • $r := \ell$      {number of remaining trials}
    • $M^{\ell} := \mathbf{0}$      {current cumulative loss vector}
2. While $t \leq \ell$, repeat:
    • Receive the predictions of the $N$ experts, $Z^r = \langle \xi_{1,t}, \ldots, \xi_{N,t} \rangle$.
    • Compute and output prediction
          $\hat{y}_t = \frac{\nu(M^r + Z^r, r - 1) - v(M^r + \mathbf{1} - Z^r, r - 1) + 1}{2}$
      where $v$ is defined by Eq. (1) and (2).
    • Receive the correct outcome $y_t$
    • $M_i^{r-1} := M_i^r + |y_t - \xi_{i,t}|$ for $i = 1, \ldots, N$.
    • $t := t + 1$
    • $r := r - 1$
```

FIG. 1. Description of algorithm **MM**.

The following theorem, the main result of this section, characterizes the loss of this strategy exactly in terms of the function v, and shows moreover that this strategy is the best possible.

THEOREM 3.2. *Let* **MM** *be the prediction strategy described above. Then for any set of experts* $\mathscr{E}$ *and for any outcome sequence* $\mathbf{y}$, *the loss of* **MM** *is bounded by*

$$L_{MM}(\mathbf{y}) - L_{\mathscr{E}}(\mathbf{y}) \leq \frac{\ell}{2} - v(\mathbf{0}, \ell),$$

where ℓ *is the number of prediction trials*, N *is the number of experts*, *and* $\mathbf{0}$ *is a vector of* N *zeros*.

Moreover, **MM** *is optimal in the sense that, for every prediction strategy* A, *there exists a set of experts* $\mathscr{E}$ *and an outcome sequence* $\mathbf{y}$ *for which*

$$L_A(\mathbf{y}) - L_{\mathscr{E}}(\mathbf{y}) \geq \frac{\ell}{2} - v(\mathbf{0}, \ell).$$

Hence $V_{N,\ell} = \ell/2 - \nu(\mathbf{0}, \ell)$.

PROOF. The first part of the theorem is proved using induction on the number r of remaining trials. As above, let M^r be an N dimensional vector that describes the losses of each of the N experts on the first $\ell - r$ trials (so r trials remain) and let λ_r denote the loss incurred by **MM** on these first $\ell - r$ trials. Then our inductive hypothesis is a bound on the net loss of **MM** at the end of the game, namely,

$$L_{MM}(\mathbf{y}) - L_{\mathscr{E}}(\mathbf{y}) \leq \lambda_r + \frac{r}{2} - v(M^r, r). \tag{4}$$

It is clear that if we choose $r = \ell$ we get the statement of the theorem, since $M^{\ell} = \mathbf{0}$. We now present the inductive proof of the claim.

For $r = 0$, the claim follows directly from the definitions since $v(M^0, 0)$ is equal to the loss of the best expert at the end of the game, $r/2 = 0$, and λ_0 is the loss of **MM**.

For $r > 0$, let $Z^r = \langle \xi_{1,t}, \ldots, \xi_{N,t} \rangle$ denote the predictions given by the experts at trial $t = \ell - r + 1$ (i.e., when there are r future outcomes to predict). Using the inductive assumption for $r - 1$ and Eq. (3) we can calculate the loss of **MM** at the end of the game; for the two possible values of the next outcome y_t we get that the net loss is bounded by the same quantity which agrees with the claim for r remaining trials.

If $y_t = 0$, then the loss of **MM** up to the next step is $\lambda_{r-1} = \lambda_r + \hat{y}$, and the loss of the experts is $M^{r-1} = M^r + Z^r$. Using the inductive assumption we get that the net loss at the end of the game will be at most

$$\begin{aligned}
&\lambda_{r-1} + \frac{r-1}{2} - v(M^{r-1}, r-1) \\
&\quad = \lambda_r + \frac{v(M^r + Z^r, r-1) - v(M^r + \mathbf{1} - Z^r, r-1) + 1}{2} \\
&\qquad + \frac{r-1}{2} - v(M^r + Z^r, r-1) \\
&\quad = \lambda_r + \frac{r}{2} - \frac{v(M^r + Z^r, r-1) + v(M^r + \mathbf{1} - Z^r, r-1)}{2}.
\end{aligned}$$

Similarly, if $y_t = 1$, then the loss of **MM** at the next step is $\lambda_{r-1} = \lambda_r + 1 - \hat{y}$, and the loss of the experts is $M^{r-1} = M^r + \mathbf{1} - Z^r$, and we get that the net loss at the end of the game will be at most

$$\begin{aligned}
&\lambda_{r-1} + \frac{r-1}{2} - v(M^{r-1}, r-1) \\
&\quad = \lambda_r + 1 - \frac{v(M^r + Z^r, r-1) - v(M^r + \mathbf{1} - Z^r, r-1) + 1}{2} \\
&\qquad + \frac{r-1}{2} - v(M^r + \mathbf{1} - Z^r, r-1) \\
&\quad = \lambda_r + \frac{r}{2} - \frac{v(M^r + Z^r, r-1) + v(M^r + \mathbf{1} - Z^r, r-1)}{2}.
\end{aligned}$$

Thus, for either value of $y_t \in \{0, 1\}$, we have that

$$\begin{aligned}
&L_{MM}(\mathbf{y}) - L_{\mathscr{E}}(\mathbf{y}) \\
&\quad \leq \left(\lambda_r + \frac{r}{2} - \frac{v(M^r + Z^r, r-1) + v(M^r + \mathbf{1} - Z^r, r-1)}{2} \right)
\end{aligned}$$

$$\begin{aligned}
&\le \max_{Z\in[0,1]^N}\left(\lambda_r + \frac{r}{2} - \frac{v(M^r+Z,\, r-1) + v(M^r+\mathbf{1}-Z,\, r-1)}{2}\right)\\
&= \lambda_r + \frac{r}{2} - \min_{Z\in[0,1]^N}\frac{v(M^r+Z,\, r-1) + v(M^r+\mathbf{1}-Z,\, r-1)}{2}\\
&= \lambda_r + \frac{r}{2} - v(M^r, r). \qquad (5)
\end{aligned}$$

This completes the induction, and the proof of the first part of the theorem.

The proof of the lower bound proceeds similarly. Let A be any prediction strategy, let r be the number of trials remaining, let M^r be the vector describing the loss of each expert up to the current trial when r trials remain, and let λ_r be the loss incurred by A up to this current trial. The natural adversarial choice for the experts' predictions on the current trial t is any vector $Z^r = \langle \xi_{1,t}, \ldots, \xi_{N,t}\rangle$ which minimizes the right-hand side of Eq. (2) (the definition of $v(M^r, r)$). If $\hat{y}_t$ is A's prediction, then the adversary chooses the outcome y_t that maximizes A's loss on the trial, $|\hat{y}_t - y_t|$.

We prove by induction on r that this adversary forces the net loss of any algorithm to be at least

$$L_A(\mathbf{y}) - L_{\mathscr{E}}(\mathbf{y}) \ge \lambda_r + \frac{r}{2} - v(M^r, r).$$

As above, equality holds when $r = 0$.

For the inductive step, let t be the trial number when r trials remain. Recall that λ_{r-1} is either $\lambda_r + \hat{y}_t$ or $\lambda_r + 1 - \hat{y}_t$ and that M^{r-1} is either $M^r + Z^r$ or $M^r + \mathbf{1} - Z^r$ depending on the value of y_t. Thus, by the inductive hypothesis and the definition of the adversary

$$\begin{aligned}
&L_A(\mathbf{y}) - L_{\mathscr{E}}(\mathbf{y})\\
&\quad\ge \max\left\{\lambda_r + \hat{y}_t + \frac{r-1}{2} - v(M^r+Z^r,\, r-1),\ \lambda_r + 1 - \hat{y}_t + \frac{r-1}{2}\right.\\
&\qquad\qquad\left. - v(M^r+\mathbf{1}-Z^r,\, r-1)\right\}\\
&\quad\ge \frac{1}{2}\left(\lambda_r + \hat{y}_t + \frac{r-1}{2} - v(M^r+Z^r,\, r-1) + \lambda_r + 1 - \hat{y}_t + \frac{r-1}{2}\right.\\
&\qquad\qquad\left. - v(M^r+\mathbf{1}-Z^r,\, r-1)\right)\\
&\quad= \lambda_r + \frac{r}{2} - \frac{v(M^r+Z^r,\, r-1) + v(M^r+\mathbf{1}-Z^r,\, r-1)}{2}
\end{aligned}$$

$$= \lambda_r + \frac{r}{2} - v(M^r, r).$$

This completes the induction. Choosing $r = \ell$ gives the stated lower bound. □

We have thus proven that the prediction strategy **MM**, described above, achieves the optimal bounds on the net-loss of any prediction strategy. However, in order to use this strategy as a prediction algorithm we need to describe how to calculate the values $v(M, r)$. At first, this calculation might seem forbiddingly complex, as it involves minimizing a recursively defined function over all choices of Z in the continuous domain $[0, 1]^N$. Fortunately, as we now show, the minimal value is always achieved at one of the corner points of the cube $Z \in \{0, 1\}^N$, so that the minimization search space is finite, albeit exponential. We prove this claim using the following lemma:

LEMMA 3.3. *For any fixed* $0 \le r \le \ell$, *the function* $v(M, r)$ *is concave, that is, for any* $0 \le \alpha \le 1$, *and for any* $X, Y \in (\mathbb{R}^+)^N$*:*

$$v(\alpha X + (1-\alpha)Y, r) \ge \alpha v(X, r) + (1-\alpha)v(Y, r).$$

PROOF. As usual, we prove the lemma by induction on r.
For $r = 0$, suppose i_0 is the index that minimizes

$$v(\alpha X + (1-\alpha)Y, 0) = \min_{1 \le i \le N} (\alpha x_i + (1-\alpha)y_i).$$

Then the convex combination of $v(X, 0)$ and $v(Y, 0)$ can be bounded as follows:

$$\alpha \min_{1 \le i \le N} (x_i) + (1-\alpha) \min_{1 \le i \le N} (y_i) \le \alpha x_{i_0} + (1-\alpha)y_{i_0} = v(\alpha X + (1-\alpha)Y, 0).$$

For $r > 0$, let $Z_0 \in [0, 1]^N$ be a choice of the argument that minimizes

$$v(\alpha X + (1-\alpha)Y, r)$$
$$= \min_{Z \in [0,1]^N} \frac{v(\alpha X + (1-\alpha)Y + Z, r-1) + v(\alpha X + (1-\alpha)Y + \mathbf{1} - Z, r-1)}{2}$$

Then we get

$$v(\alpha X + (1-\alpha)Y, r)$$
$$= \frac{v(\alpha X + (1-\alpha)Y + Z_0, r-1) + v(\alpha X + (1-\alpha)Y + \mathbf{1} - Z_0, r-1)}{2}$$
$$= \frac{v(\alpha(X + Z_0) + (1-\alpha)(Y + Z_0), r-1) + v(\alpha(X + \mathbf{1} - Z_0) + (1-\alpha)(Y + \mathbf{1} - Z_0), r-1)}{2}.$$

Using the induction assumption we can bound each of the two terms and get that

$$
\begin{aligned}
&v(\alpha X + (1-\alpha)Y, r)\\
&\quad\geq \frac{\alpha v(X + Z_0, r-1) + (1-\alpha)v(Y + Z_0, r-1)}{2}\\
&\qquad + \frac{\alpha v(X + \mathbf{1} - Z_0, r-1) + (1-\alpha)v(Y + \mathbf{1} - Z_0, r-1)}{2}\\
&\quad= \alpha\frac{v(X + Z_0, r-1) + v(X + \mathbf{1} - Z_0, r-1)}{2}\\
&\qquad + (1-\alpha)\,\frac{v(Y + Z_0, r-1) + v(Y + \mathbf{1} - Z_0, r-1)}{2}\\
&\quad\geq \alpha \min_{Z\in[0,1]^N} \frac{v(X + Z, r-1) + v(X + \mathbf{1} - Z, r-1)}{2}\\
&\qquad + (1-\alpha) \min_{Z\in[0,1]^N} \frac{v(Y + Z, r-1) + v(Y + \mathbf{1} - Z, r-1)}{2}\\
&\quad= \alpha v(X, r) + (1-\alpha)v(Y, r). \qquad \square
\end{aligned}
$$

If we fix M and view the function

$$\frac{(v(M + Z, r-1) + v(M + \mathbf{1} - Z, r-1))}{2}$$

as a function of Z, we see that it is simply a positive constant times the sum of two concave functions and thus it also is concave. Therefore, the minimal value of this function over the closed cube $Z \in [0, 1]^N$ is achieved in one of the corners of the cube.

This means that the function $v(M, r)$ can be computed recursively by minimizing over the 2^N (Boolean) choices of the experts' predictions. Each of these choices involves two recursive calls and the recursion has to be done to depth r. Therefore, a total of $2^{r(N+1)}$ recursive calls are made, requiring time $O(N2^{r(N+1)})$.

Dynamic programming leads to a better algorithm for calculating $v(M, r)$. However, it is still exponential in N. An interesting question is whether $v(M, r)$ can be computed efficiently.

To summarize this section, we have described an optimal prediction algorithm and given a recursive formula which defines its worst case loss, and thereby obtained a recursive formula for $V_{N,\ell}$. We do not have a closed-form equation for $V_{N,\ell}$. However, we can always calculate it exactly in finite time (see Figure 5 for the values of $V_{N,\ell}$ for some small ranges of N and ℓ). Moreover, the following section provides a simple adversarial strategy that generates a lower bound on the optimal net loss $V_{N,\ell}$ and Section 4 provides a simple prediction algorithm that generates an upper bound on $V_{N,\ell}$. As we will see, these two bounds are quite tight.

3.1. Prediction using Static Experts.

The strategy described above can be refined to handle certain special cases. As an example of this technique, we show in this section how to handle the case that all the experts are *static* in the sense that their predictions do *not* depend either on the observed outcomes or on the learner's predictions.[7] That is, each expert can be viewed formally as a function $\mathscr{E}_i : \{1, \ldots, \ell\} \rightarrow [0, 1]$ with the interpretation that the prediction at time t is $\xi_{i,t} = \mathscr{E}_i(t)$. We assume further that the learner knows this function and thus can compute the future predictions of all the experts. Thus, the adversary must choose the static experts at the beginning of the game and reveal this choice to the learning algorithm. The adversary still chooses each outcome y_t on-line as before. The resulting game is called the *binary sequence prediction game with static experts* and its min/max value is denoted $V_{N,\ell}^{(static)}$.

Since this game is easier for the minimizing player (the predictor) than the general game, it is clear that $V_{N,\ell}^{(static)} \leq V_{N,\ell}$. When $N = 2$, the values of the two games are the same for all ℓ. However, a calculation shows that $V_{3,4}^{(static)} < V_{3,4}$ with strict inequality, so the general sequence prediction game is actually harder in the worst case than the same game with static experts. The actual values are $V_{3,4}^{(static)} = 1$ and $V_{3,4} = \frac{17}{16}$.

We give below a characterization of the optimal prediction and adversarial strategies for the binary sequence prediction game with static experts. In fact we go further and analyze the game explicitly for every possible choice of the static experts. The resulting min/max values have a simple geometric interpretation. For real vectors $\mathbf{x}$ and $\mathbf{y}$ of length ℓ, let $\|\mathbf{x} - \mathbf{y}\|_1 = {}_{t=1}^{\ell} |x_t - y_t|$. Let $\mathscr{E} = \{\mathscr{E}_1, \ldots, \mathscr{E}_N\}$ be a set of N static experts. For any expert $\mathscr{E}_i$, its loss on the bit sequence $\mathbf{y}$ is ${}_{t=1}^{\ell} |\mathscr{E}_i(t) - y_t| = \|\mathscr{E}_i - \mathbf{y}\|_1$, viewing $\mathscr{E}_i$ as a vector in $[0, 1]^\ell$. Thus $L_{\mathscr{E}}(\mathbf{y}) = \min_i \|\mathscr{E}_i - \mathbf{y}\|_1$. We define the *average covering radius* of $\mathscr{E}$, denoted $R(\mathscr{E})$, as the average l_1 distance from a bit sequence $\mathbf{y}$ to the nearest expert in $\mathscr{E}$, that is

$$R(\mathscr{E}) = E_{\mathbf{y}} L_{\mathscr{E}}(\mathbf{y}) = E_{\mathbf{y}} \min_i \|\mathscr{E}_i - \mathbf{y}\|_1,$$

where $E_{\mathbf{y}}$ denotes expectation over a uniformly random choice of $y \in \{0, 1\}^\ell$.

We will use the following convexity result, an analog of Lemma 3.3.

LEMMA 3.1.1. *Let* $\mathscr{E} = \{\mathscr{E}_i\}$ *and* $\mathscr{F} = \{\mathscr{F}_i\}$ *be two sets of* N *vectors in* $[0, 1]^\ell$ *and let* $0 \leq \alpha \leq 1$. *Then*

$$R(\alpha\mathscr{E} + (1 - \alpha)\mathscr{F}) \geq \alpha R(\mathscr{E}) + (1 - \alpha) R(\mathscr{F}),$$

where $\alpha\mathscr{E} + (1 - \alpha)\mathscr{F}$ *is the set of* N *vectors* $\{\alpha\mathscr{E}_i + (1 - \alpha)\mathscr{F}_i\}$.

PROOF

$$\begin{aligned} R(\alpha\mathscr{E} + (1 - \alpha)\mathscr{F}) &= E_{\mathbf{y}} \min_i \sum_t |\alpha\mathscr{E}_{i,t} + (1 - \alpha)\mathscr{F}_{i,t} - y_t| \\ &= E_{\mathbf{y}} \min_i \sum_t (|\alpha\mathscr{E}_{i,t} - \alpha y_t| + |(1 - \alpha)\mathscr{F}_{i,t} - (1 - \alpha) y_t|) \end{aligned}$$

[7] In an earlier version of this paper [Cesa-Bianchi et al. 1993], we incorrectly claimed that the same analysis also applied to all *simulatable* experts, that is, experts whose predictions can be calculated as a function only of the preceding outcomes.

$$
\begin{aligned}
&= E_{\mathbf{y}} \min_i \left(\alpha\|\mathcal{E}_i - \mathbf{y}\|_1 + (1-\alpha)\|\mathcal{F}_i - \mathbf{y}\|_1\right)\\
&\geq E_{\mathbf{y}}\left(\alpha \min_i \|\mathcal{E}_i - \mathbf{y}\|_1 + (1-\alpha) \min_i \|\mathcal{F}_i - \mathbf{y}\|_1\right)\\
&= \alpha R(\mathcal{E}) + (1-\alpha) R(\mathcal{F}),
\end{aligned}
$$

where the second equality follows from a case analysis of $y_t = 0$ and $y_t = 1$, combined with the fact that $\mathcal{E}_{i,t}$, $\mathcal{F}_{i,t} \in [0, 1]$. □

THEOREM 3.1.2. *Let $\mathcal{E}$ be a set of static experts whose current and future predictions are accessible to the prediction algorithm. Then there exists a prediction strategy* **MS** *such that for every sequence* **y**, *we have*

$$L_{MS}(\mathbf{y}) - L_{\mathcal{E}}(\mathbf{y}) = \frac{\ell}{2} - R(\mathcal{E}).$$

Moreover, **MS** *is optimal in the sense that for every prediction strategy A, there exists a sequence* **y** *such that*

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

Hence

$$V_{N,\ell}^{(static)} = \frac{\ell}{2} - \min_{\mathcal{E}} R(\mathcal{E}),$$

where the minimum is over all sets $\mathcal{E}$ of N vectors in $\{0, 1\}^\ell$.

PROOF. For any prediction strategy A, the expected value of $L_A - L_{\mathcal{E}}$ with respect to a uniformly random choice of $\mathbf{y} \in \{0, 1\}^\ell$ is simply $\ell/2 - R(\mathcal{E})$ since we expect any algorithm to have loss $\ell/2$ on an entirely random sequence, and $R(\mathcal{E})$ is the expected loss of the best expert in $\mathcal{E}$. Thus, there must be some sequence **y** for which $L_A(\mathbf{y}) - L_{\mathcal{E}}(\mathbf{y})$ is at least as great as this expectation; this proves the second part of the theorem.

The first part of the theorem can be proved using the technique in Section 3 with only minor modifications, which we sketch briefly. First, the function v is redefined to take account of the fact that the experts' predictions are prespecified. As the predictions of the experts correspond to vectors in $[0, 1]^\ell$, we can think about them as rows in an $\ell \times N$ matrix. We can calculate the average covering radius by considering one column (i.e., game iteration) at a time. That is, we define the new function $\tilde{v}$ as follows:

$$
\begin{aligned}
\tilde{v}(M, 0) &= \min_i M_i\\
\tilde{v}(M, r) &= \frac{\tilde{v}(M + Z^r, r-1) + \tilde{v}(M + \mathbf{1} - Z^r, r-1)}{2}
\end{aligned}
$$

where $Z^r = \langle \xi_{1,t}, \ldots, \xi_{N,t}\rangle$ is the experts' predictions at trial $t = \ell - r + 1$.

The (re)proof of Lemma 3.1 for $\tilde{v}$ is similar, except that we no longer minimize over $Z \in [0, 1]^N$, and in the case that $r > 0$, Z_0 is replaced by Z^r.

The new prediction strategy **MS** computes its prediction at time $t = \ell - r + 1$ as before with the obvious changes:

$$\hat{y}_t = \frac{\tilde{v}(M^r + Z^r, r - 1) - \tilde{v}(M^r + \mathbf{1} - Z^r, r - 1) + 1}{2}.$$

The induction argument given in the first part of the proof of Theorem 3.2 holds with little modification. The function v is obviously replaced by $\tilde{v}$, and the inductive hypothesis given by Eq. (4) is modified so that equality holds for every outcome sequence:

$$L_{MS}(\mathbf{y}) - L_{\mathcal{E}}(\mathbf{y}) = \lambda_r + \frac{r}{2} - \tilde{v}(M^r, r).$$

Also, Eq. (5) becomes the equality:

$$\begin{aligned} L_{MS}(\mathbf{y}) - L_{\mathcal{E}}(\mathbf{y}) &= \left(\lambda_r + \frac{r}{2} - \frac{\tilde{v}(M^r + Z^r, r - 1) + \tilde{v}(M^r + \mathbf{1} - Z^r, r - 1)}{2} \right) \\ &= \lambda_r + \frac{r}{2} - \tilde{v}(M^r, r). \end{aligned}$$

By expanding $\tilde{v}(\mathbf{0}, \ell)$ according to the recursive definition, we find that

$$\begin{aligned} \tilde{v}(\mathbf{0}, \ell) &= \frac{1}{2^\ell} \sum_{\mathbf{y} \in \{0, 1\}^\ell} \tilde{v}\left(\sum_{r=1}^{\ell} (Z^r(1 - y_{\ell-r+1}) + (\mathbf{1} - Z^r) y_{\ell-r+1}), 0 \right) \\ &= \frac{1}{2^\ell} \sum_{\mathbf{y} \in \{0, 1\}^\ell} \tilde{v}(\langle \|\mathcal{E}_i - \mathbf{y}\|_1 \rangle_{i=1\cdots N}, 0) \\ &= \frac{1}{2^\ell} \sum_{\mathbf{y} \in \{0, 1\}^\ell} \min_i \|\mathcal{E}_i - \mathbf{y}\|_1 \\ &= E_{\mathbf{y}} \min_i \|\mathcal{E}_i - \mathbf{y}\|_1 = R(\mathcal{E}). \end{aligned}$$

Finally, it follows directly from the first two statements of the theorem that

$$V_{N,\ell}^{(static)} = \frac{\ell}{2} - \inf_{\mathcal{E}} R(\mathcal{E}),$$

where the infimum is over all sets $\mathcal{E}$ of N vectors in $[0, 1]^\ell$. However, in light of Lemma 3.1.1, $R(\mathcal{E})$ must be minimized by some extremal $\mathcal{E}$, that is, by $\mathcal{E} \subseteq \{0, 1\}^\ell$. The last statement of the theorem follows. □

Theorem 3.1.2 tells us how to compute the worst-case performance of the best possible algorithm for any set of static 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 3.1.2 implies that the loss of the optimal algorithm **MS** is worse than the loss of the best expert by the following amount:

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

This result was previously proved by Cover [1965]; we obtain it as a special case.

Strategy **MS** 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). This is why we need the experts to be static. In general, we do not know how to efficiently compute this expectation exactly. However, the expectation can be *estimated* by sampling a polynomial number of randomly chosen future outcome sequences. Thus, there exists an efficient randomized variation of **MS** that is arbitrarily close to optimal.

3.2. AN ASYMPTOTIC LOWER BOUND ON $V_{N,\ell}$. We now use Theorem 3.1.1 to give an asymptotic lower bound on the performance of any prediction algorithm. To do this, we need to show that there are sets $\mathscr{E}$ of N vectors in $\{0, 1\}^{\ell}$ with small $R(\mathscr{E})$. We do this with a random construction, using the following lemma:

LEMMA 3.2.1. *For each ℓ, $N \geq 1$, 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

$$\liminf_{N\to\infty} \liminf_{\ell\to\infty} \frac{(\ell/2) - E(A_{\ell,N})}{\sqrt{(\ell/2) ln\ N}} \geq 1.$$

PROOF. See Appendix A. □

From this we get

COROLLARY 3.2.2. *For all N, ℓ, let $R_{N,\ell} = min_{\mathscr{E}}\ R(\mathscr{E})$, where the minimum is over all $\mathscr{E} \subseteq \{0, 1\}^{\ell}$ of cardinality N. Then*

$$\liminf_{N\to\infty} \liminf_{\ell\to\infty} \frac{\ell/2 - R_{N,\ell}}{\sqrt{(\ell/2) ln\ N}} \geq 1.$$

PROOF. Clearly

$$\min_{\mathscr{E}} R(\mathscr{E}) \leq E(R(\mathscr{E})) = E(A_{\ell,N}),$$

where the expectation is over the independent random choice of N binary vectors in $\mathscr{E}$, and $A_{\ell,N}$ is as defined in Lemma 3.2.1. Hence, the result follows directly from that lemma. □

Finally, we obtain

Algorithm **P**(β)

1. All initial weights $\{\mathrm{w}_{1,1}, \ldots ,w_{N,1}\}$ 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 $\hat{y}_t$ 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 $\hat{y}_t = F_\beta(r_t)$.
3. After the correct outcome y_t is observed, the weight vector is updated in the following way.
    - For each $i = 1$ to N, $w_{i,t+1} = w_{i,t}\, U_\beta(|\xi_{i,t} - y_t|)$.

Definition of $F_\beta(r)$ and $U_\beta(q)$.

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

$$1 + \frac{\ln((1-r)\beta+r)}{2\ln(2/(1+\beta))} \le F_\beta(r) \le \frac{-\ln(1-r+r\beta)}{2\ln(2/(1+\beta))}, \tag{6}$$

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

$$\beta^q \le U_\beta(q) \le 1 - (1-\beta)q, \tag{7}$$

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

FIG. 2. Description of algorithm **P**(β), with parameter $0 \le \beta < 1$.

THEOREM 3.2.3

$$\liminf_{N\to\infty} \liminf_{\ell\to\infty} \frac{V_{N,\ell}}{\sqrt{(\ell/2)\ln N}} \ge \liminf_{N\to\infty} \liminf_{\ell\to\infty} \frac{V_{N,\ell}^{(static)}}{\sqrt{(\ell/2)\ln N}} \ge 1.$$

PROOF. Follows Corollary 3.2.2, Theorem 3.1.2, and the fact that $V_{N,\ell} \ge V_{N,\ell}^{(static)}$. □

Hence, for any $\epsilon > 0$, there exist sufficiently large N and ℓ such that

$$V_{N,\ell} \ge (1-\epsilon)\sqrt{(\ell/2)\ln N}.$$

4. *Some Simple Prediction Algorithms*

In this section, we present a parameterized prediction algorithm **P** for combining the predictions of a set of experts. Unlike the optimal strategy outlined in Section 3, algorithm **P** can be implemented efficiently. The analysis of **P** will give an upper bound for the min/max value $V_{N,\ell}$ that asymptotically matches the lower bound derived in the previous section.

4.1. THE ALGORITHM **P**. The prediction algorithm **P** is given in Figure 2. It works by maintaining a (nonnegative) *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 **P** then makes a prediction that is some function of this weighted average. Then **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 $|\xi_{i,t} - y_t|$. 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 Eqs. (6) and (7), respectively, in Figure 2. These are called the *prediction* and *update* functions, respectively. In terms of our analysis, the exact choice for these functions is not important, as long as they lie in the allowed range. In fact, different choices could be made at different times. The following lemma shows that these ranges are nonempty.

LEMMA 4.1.1. *For any* $0 \le \beta < 1$ *and* $0 \le a \le 1$,

(1) $1 + \dfrac{ln((1-a)\beta + a)}{2\, ln\, 2/(1+\beta)} \le \dfrac{-\, ln(1 - a + a\beta)}{2\, ln\, 2/(1+\beta)}$

(2) $\beta^a \le 1 - a(1 - \beta)$.

PROOF. We begin by proving part (1). The inequality can be rewritten as

$$1 + \frac{\ln[(\beta - a\beta + a)(1 - a + a\beta)]}{2 \ln 2/(1+\beta)} \le 0.$$

Since $0 \le \beta < 1$, this is in turn equivalent to

$$\ln[(\beta - \alpha\beta + a)(1 - a + a\beta)] \le 2 \ln \frac{1+\beta}{2}.$$

Exponentiating both sides yields

$$(\beta - a\beta + a)(1 - a + a\beta) \le \left(\frac{1+\beta}{2}\right)^2$$

which holds since $xy \le ((x+y)/2)^2$ for all real x and y (here we take $x = \beta - a\beta + a$ and $y = 1 - a + a\beta$).

To prove part (2), notice that $f(a) = \beta^a$ is convex since it has nonnegative second derivative for all $\beta > 0$. Thus, by definition of convex function,

$$f(\alpha x_0 + (1-\alpha)x_1) \le \alpha f(x_0) + (1-\alpha) f(x_1)$$

for all x_0, x_1 and all $0 \le \alpha \le 1$. The proof is then concluded by choosing $x_0 = 0$, $x_1 = 1$, and $\alpha = 1 - a$. □

4.2. The Performance of Algorithm $\mathbf{P}(\beta)$. Algorithm $\mathbf{P}$'s performance is summarized by the following theorem, which generalizes a similar result of Vovk [1990].

Theorem 4.2.1. *For any* $0 \le \beta < 1$, *for any set* $\mathscr{E}$ *of* N *experts, and for any binary sequence* $\mathbf{y}$ *of length* ℓ, *the loss of* $\mathbf{P}(\beta)$ *satisfies*

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

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

Lemma 4.2.2

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

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

$$|\hat{y}_t - y_t| \le \frac{1}{2\ ln\ 2/(1+\beta)}\ ln\left(\frac{\sum_{i=1}^N w_{i,t}}{\sum_{i=1}^N w_{i,t+1}}\right). \qquad (8)$$

The lemma then follows from summing the above inequality for $t = 1, \ldots, \ell$. We first lower bound the numerator of the right-hand-side of the above inequality:

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

where the inequality follows from Eq. (7), and the last equality is verified by a case analysis using the fact that $y_t \in \{0, 1\}$. Thus, Eq. (8) is implied by

$$|\hat{y}_t - y_t| \le -\frac{\ln(1-(1-\beta)|r_t - y_t|)}{2\ \ln\ 2/(1+\beta)}.$$

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 (6) which we assumed for the prediction function. □

Proof of Theorem 4.2.1. All initial weights equal 1 and thus $^{N}_{i=1}\ w_{i,1} = N$. Let j be an expert with minimum total loss on $\mathbf{y}$, that is, $^{\ell}_{t=1}\ |\xi_{j,t} - y_t| = L_{\mathscr{E}}(\mathbf{y})$. Since, by Eq. (7), $U_\beta(q) \ge \beta^q$, we have that

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

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

The theorem now follows from Lemma 4.2.2. □

4.3. DISCUSSION OF THE ALGORITHM. Although our algorithm allows any update function $U_\beta(q)$ between the exponential β^q (used by Vovk in his related work [Vovk 1990]) and the linear function $1 - (1 - \beta)q$ that upper bounds it, it turns out that the linear update has a nice Bayesian interpretation, and thus in some sense may be preferable.

To get this Bayesian interpretation, 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 ith 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

$$p_{i,t} = \eta + (1 - 2\eta)\xi_{i,t}, \tag{9}$$

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 easily obtain the following result:

THEOREM 4.3.1. *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 $w_{i,t}/(\ {}_{j=1}^{N} w_{j,t})$ is the posterior probability that the outcome sequence is being generated from the distribution defined in (9) above for the ith 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.

PROOF. Initially $w_{i,1} = 1$ for all i, hence the normalized initial weights are the uniform prior distribution, as required. It suffices to show that for each time $t \geq 1$, the ratio of successive weights $w_{i,t+1}/w_{i,t}$ is proportional to the ratio $P(i|y_1, \ldots, y_t)/P(i|y_1, \ldots, y_{t-1})$ of successive posterior probabilities (with the same constant of proportionality for all i), where $P(i|y_1, \ldots, y_t)$ denotes the posterior probability that the sequence is being generated from the distribution of the ith expert given $y_1, \ldots, y_t$. However, using Bayes rule

$$\frac{P(i|y_1, \ldots, y_t)}{P(i|y_1, \ldots, y_{t-1})} \propto \frac{P(y_1, \ldots, y_t|i)}{P(y_1, \ldots, y_{t-1}|i)}$$

$$= \begin{cases} p_{i,t} & \text{if } y_t = 1 \\ 1 - p_{i,t} & \text{if } y_t = 0 \end{cases},$$

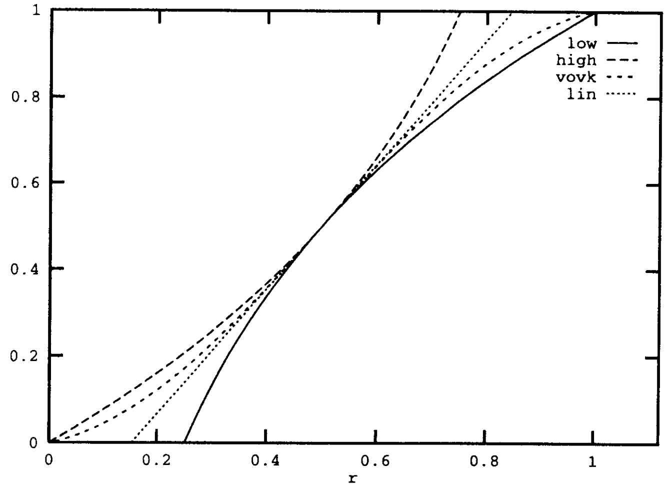

FIG. 3. This figure shows the upper (high) and lower (low) bounds on the possible values of the prediction function F_β for $\beta = 0$ (Inequality (6)). Also shown are two possible choices for F_β, a piecewise linear function (lin) given in (10), and the function that has been suggested by Vovk's work (vovk) given in (11).

where $p_{i,t}$ is as defined in (9) above, and $P(y_1, \ldots, y_t|i)$ denotes the probability of $y_1, \ldots, y_t$ under the distribution defined above for the ith expert. Using Eq. (9) with the substitution $\eta = \beta/(1+\beta)$, this implies that

$$\frac{P(i|y_1, \ldots, y_t)}{P(i|y_1, \ldots, y_{t-1})} \propto \begin{cases} \beta + (1-\beta)\xi_{i,t} & \text{if } y_t = 1 \\ 1-(1-\beta)\xi_{i,t} & \text{if } y_t = 0 \end{cases}$$
$$= 1-(1-\beta)|\xi_{i,t} - y_t|.$$

As this is precisely the factor by which the weights are updated after seeing y_t, this is the ratio of successive weights $w_{i,t+1}/w_{i,t}$. □

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 $|\hat{y}_t - y_t|$. 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 3, 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 Helmbold and Warmuth [1995] and Feder et al. [1992]. Hence, we must deviate from the Bayes method at this step. This leads to the requirements we have specified for the prediction function $F_\beta(r)$.

One function that satisfies the requirements for F_β is the piecewise linear function[8]

$$F_\beta(r) = \begin{cases} 0 & \text{if} \quad r \le 1/2 - c \\ 1/2 - (1 - 2r)/4c & \text{if} \quad 1/2 - c \le r \le 1/2 + c \\ 1 & \text{if} \quad r \ge 1/2 + c \end{cases} \tag{10}$$

where

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

Another possible choice for F_β is suggested by Vovk's work[9] [Vovk 1990]

$$F_\beta(r) = \frac{\ln(1 - r + r\beta)}{\ln(1 - r + r\beta) + \ln((1-r)\beta + r)}. \tag{11}$$

Figure 3 contains a plot of these functions when $\beta = 0$, along with the upper and lower bounds on F_β given in Inequality (6). Recall that $\beta = 0$ corresponds to the case when there is no noise. In that case $-\ln(1 - r)$ is the *information gain* when the outcome is zero and $-\ln(r)$ is the information gain when the outcome is one. Furthermore, the prediction function (11) is the normalized information gain when the outcome is zero. See Helmbold and Warmuth [1995] for a more detailed discussion. As the noise increases, $\beta \to 1$ and all four curves converge to the identity function.

Finally, we note that the parameterized bound given in Theorem 4.2.1 on the performance of algorithm **P** was first proved by Vovk [1990] for his version of F_β and the exponential update. Also, Littlestone and Warmuth [1994] prove a bound for their algorithm *WMC*, which has the same form as the bound of Theorem 4.2.1, except the denominator $2 \ln 2/(1 + \beta)$ is replaced by the smaller function $1 - \beta$. Their algorithm uses the prediction function $F_\beta(r_t) = r_t$ and works for the more general setting when the outcome y_t can be in the interval $[0, 1]$ as opposed to being binary. For the noise-free case ($\beta = 0$), their algorithm becomes the Gibbs algorithm (see discussion in Helmbold and Warmuth [1995]). The bound of Theorem 4.2.1 (with denominator $2 \ln (2/(1 + \beta))$) was recently also obtained by Kivinen and Warmuth [1994] for the case when the outcomes are in $[0, 1]$. Curiously enough, the denominator of $\ln (2/(1 + \beta))$ is obtained by the Weighted Majority algorithm of Littlestone and Warmuth [1994], which assumes that the outcomes are binary and predicts binary as well (see Cesa-

[8] A similar piecewise linear function was suggested by Feder et al. [1992] in a related context.

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

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

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 need not satisfy our Inequality (6). However, when the experts' predictions are all in $\{0, 1\}$, then Vovk's prediction function is equivalent to the one described in Eq. (11).

Bianchi et al. [1996] for a detailed treatment of the case when the outcomes are binary).

4.4. PERFORMANCE FOR BOUNDED $L_{\mathscr{E}}$. 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. When $L_{\mathscr{E}}(\mathbf{y})$ is replaced by K, the upper bound from Theorem 4.2.1 can be written

$$L(\beta) = \frac{\ln N - K \ln \beta}{2 \ln 2/(1+\beta)}.$$

It has been shown by Vovk and others [Vovk 1990; Cesa-Bianchi et al. 1996] that $L^* = \inf\{L(\beta) : 0 \leq \beta < 1\}$ is the unique value of L satisfying

$$L = \frac{\log_2 N}{2} + L \cdot H\left(\frac{K}{2L}\right),$$

where $H(p)$ is the binary entropy, $-p \log_2(p) - (1-p)\log_2(1-p)$. This minimum is achieved when $\beta = K/(2L^* - K)$. However, it is difficult to explicitly solve for L^* and the corresponding β. A recent paper by Cesa-Bianchi et al. [1996] shows how binary search can be used to choose a value for β that yields the bound $\lceil L^* \rceil$. In this paper, we give an explicit choice of β as a function of $\log(N)/K$, which approximately minimizes

$$\frac{\ln N - K \ln \beta}{2 \ln (2/(1+\beta))}$$

and leads to good closed-form bounds (see Figure 4).

We will use the following function in our choice of β.

$$g(z) = \frac{1}{1 + 2z + z^2/\ln 2}. \tag{12}$$

We give $g(\infty)$ its natural value of 0. The key property of this function is the following inequality.

LEMMA 4.4.1. *For any $z > 0$ or $z = \infty$,*

$$\frac{z^2 - \mathit{ln}\ g(z)}{2\ \mathit{ln}\ (2/(1+g(z)))} \leq 1 + z + \frac{z^2}{2\ \mathit{ln}\ 2}.$$

PROOF. See Appendix B. □

Another simple inequality that we need in the proof is given in the following lemma.

LEMMA 4.4.2. *For all $0 < \beta < 1$*

$$\frac{-\mathit{ln}(\beta)}{2\ \mathit{ln}\ 2/(1+\beta)} \geq 1.$$

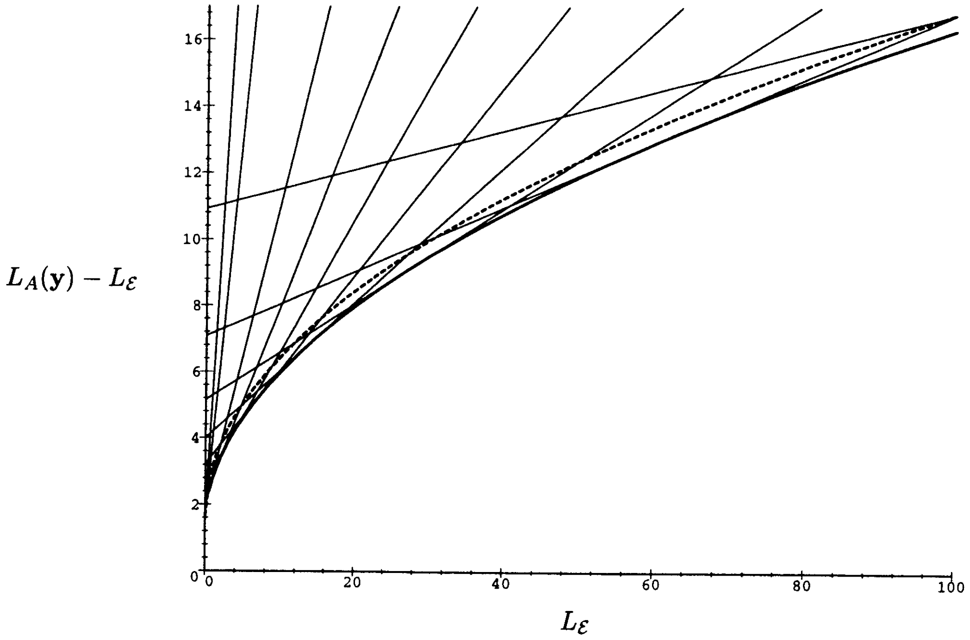

FIG. 4. This figure describes the bounds obtained by algorithm $\mathbf{P}(\beta)$ when an upper bound on $L_{\mathscr{E}}$ is given. The horizontal axis corresponds to the known upper bound and the vertical axis to $L_{\mathbf{P}(\beta)} - L_{\mathscr{E}}$. The number of experts is assumed to be 10. The thin straight lines correspond to the upper bounds achieved by choosing β to be one of 0.001, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8. The continuous curve corresponds to the bound achieved when β is chosen as in Theorem 4.4.3, and the dotted curve corresponds to the upper bound given in the theorem.

PROOF. Since $\beta < 1$, the lemma is equivalent to $\ln(\beta) \leq \ln\ ((1 + \beta)/2)^2$, which follows from the trivial inequality

$$\beta \leq \left(\frac{1 + \beta}{2}\right)^2. \qquad \square$$

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

THEOREM 4.4.3. *Pick any positive integer N and nonnegative real K. If*

$$\beta = g\left(\sqrt{\frac{\ln N}{K}}\right)$$

for the g defined in Eq. (12), *then for any set $\mathscr{E}$ of N experts and for any sequence* $\mathbf{y}$ *such that* $L_{\mathscr{E}}(\mathbf{y}) \leq K$ *we have*

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

PROOF. The proof is trivial when $N = 1$, since the algorithm makes the same predictions as the single expert. For the remainder of the proof, we assume that $N \geq 2$, so

$$\beta = g\left(\sqrt{\frac{\ln N}{K}}\right)$$

is strictly less than 1. From Theorem 4.2.1, we know that for *any* choice of $\beta \in [0, 1)$

$$L_{\mathbf{P}(\beta)}(\mathbf{y}) \leq \frac{\ln N - L_{\mathscr{E}}(\mathbf{y}) \ln \beta}{2 \ln 2/(1 + \beta)}. \tag{13}$$

We rewrite (13) as

$$L_{\mathbf{P}(\beta)}(\mathbf{y}) \leq L_{\mathscr{E}}(\mathbf{y}) + \frac{\ln N}{2 \ln 2/(1 + \beta)} + L_{\mathscr{E}}(\mathbf{y})\left(\frac{-\ln \beta}{2 \ln 2/(1 + \beta)} - 1\right).$$

From Lemma 4.4.2, we know that

$$-\frac{\ln(\beta)}{2 \ln(2/(1 + \beta))} \geq 1,$$

and from the conditions of the theorem we know that $L_{\mathscr{E}}(\mathbf{y}) \leq K$. Based on these we get that

$$\begin{aligned} L_{\mathbf{P}(\beta)}(\mathbf{y}) &\leq L_{\mathscr{E}}(\mathbf{y}) + \frac{\ln N}{2 \ln (2/(1 + \beta))} + K\left(\frac{-\ln \beta}{2 \ln (2/(1 + \beta))} - 1\right) \\ &= L_{\mathscr{E}}(\mathbf{y}) + K\left(\frac{z^2 - \ln \beta}{2 \ln (2/(1 + \beta))} - 1\right), \end{aligned}$$

where

$$z = \sqrt{\frac{\ln N}{K}}.$$

Since β was chosen to be $g(z)$, we use the inequality of Lemma 4.4.1 to obtain

$$L_{\mathbf{P}(\beta)}(\mathbf{y}) \leq L_{\mathscr{E}}(\mathbf{y}) + \sqrt{K \ln N} + \frac{\log_2 N}{2},$$

completing the proof. □

To get a feel for the bound given in Theorem 4.4.3, it may be helpful to consider the average per-trial loss guaranteed by the bound. Letting $\alpha = K/\ell$, we get:

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

Thus, for large ℓ, the average loss of $\mathbf{P}$ approaches that of the best expert. The rate of convergence of the average loss depends on α: for "small" α, the rate of convergence is roughly $O(1/\ell)$ (for large ℓ and N fixed); for fairly large α (say $\Theta(1)$, so that K is linear in ℓ), the middle term dominates, giving a slower convergence rate of $O(1/\sqrt{\ell})$.

4.5. Performance for Known Sequence Length. As a corollary of Theorem 4.4.3, 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. Theorem 3.2.3 shows that this guaranteed difference is very close to optimal.

Theorem 4.5.1. *Let $\beta = g(\sqrt{2\, ln(N+1)/\ell})$. Then for any set $\mathscr{E}$ of N experts, and for any sequence* $\mathbf{y}$ *of length ℓ there is a prediction algorithm $\mathbf{P}'(\beta)$ such that*

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

Proof. As the length of the sequence is ℓ, the largest possible loss is ℓ; however, this bound can be easily decreased to $\ell/2$. To do so, we add to the N experts of $\mathbf{P}$ a single new expert whose predictions are the inverse of the predictions of the first expert, that is $\xi_{N+1,t} = 1 - \xi_{1,t}$. We denote the algorithm that uses the expanded pool of experts by $\mathbf{P}'$. It is easy to see that for any $\mathbf{y}$, either $L_{\mathscr{E}_1} \le \ell/2$ or $L_{\mathscr{E}_{N+1}} \le \ell/2$. Thus, for the increased pool of experts we have $L_{\mathscr{E}} \le \ell/2$ and from Theorem 4.4.3 we get the statement of the theorem. □

We remark that while the bound stated in Theorem 4.5.1 holds for all ℓ, there is a slightly better bound on $\mathbf{P}'(\beta)$ for the given choice of β when $\ell \to \infty$ (and N remains fixed):

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

This can be proved by a Taylor expansion of the bound given in Theorem 4.2.1.

Combining Theorem 3.2.3 and Theorem 4.5.1, we see that $\mathbf{P}'(\beta)$'s performance is very close to optimal for sufficiently large N and ℓ, and we thereby obtain an upper bound on the min/max value $V_{N,\ell}$ of the general binary sequence prediction game defined in Section 3.

Theorem 4.5.2. *For all N, ℓ,*

$$V_{N,\ell} \le \sqrt{\frac{\ell\, ln(N+1)}{2}} + \frac{log_2(N+1)}{2}$$

and

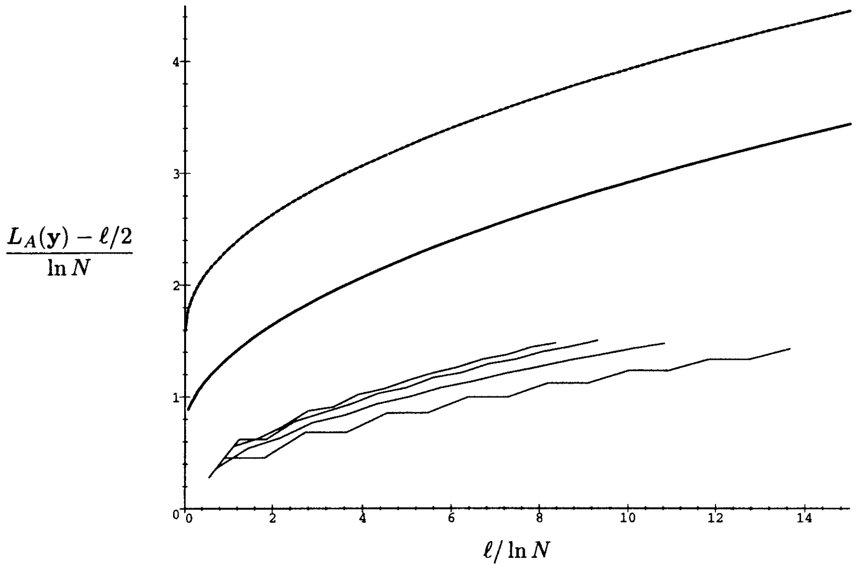

FIG. 5. This figure describes the relationship between the upper bounds guaranteed by $\mathbf{P}'(\beta)$ when the length of the sequence is given to the algorithm as input and the corresponding min/max values. The min/max values are scaled so that they can all be compared to the same upper bound. The horizontal axis corresponds to the length of the sequence divided by ln(N), where N is the number of experts, and the vertical axis corresponds to $(L_{\mathbf{P}'(\beta)} - \ell/2)/\ln(N)$. The two thick-line curves correspond to the upper bounds given by the algorithm as in Figure 4. The four piece-wise linear graphs correspond to the min/max values for $N = 2, 3, 4, 5$ and $\ell = 1, \ldots, 15$.

$$lim_{N\to\infty} lim_{\ell\to\infty} \frac{V_{N,\ell}}{\sqrt{(\ell/2) ln\ N}} = lim_{N\to\infty} lim_{\ell\to\infty} \frac{V_{N,\ell}^{(static)}}{\sqrt{(\ell/2) ln\ N}} = 1.$$

PROOF. The first statement follows from Theorem 4.5.1, and the second follows from this and Theorem 3.2.3. □

We have thus shown that the ratio between $V_{N,\ell}$ and $L_{\mathbf{P}'(\beta)}(\mathbf{y}) - L_{\mathscr{E}}(\mathbf{y})$ converges to 1 as ℓ and N grow. While this is a rather strict notion of optimality, there is still a gap between the upper and lower bounds and it is interesting to consider the actual numbers to see where improvement might be possible. We give such comparisons in Figures 5, 6, and 7. These comparisons indicate that the lower bound is very close to the min/max value even for small values of N and ℓ. The space for improvement is mostly in the upper bounds, that is, in improving the prediction algorithm or its analysis.

As a final note, we also get from Theorem 4.5.1 an interesting geometric corollary concerning the average covering radius of a set of binary vectors. Recall that we defined the average covering radius of $\mathscr{E} \subseteq \{0, 1\}^\ell$ by

$$R(\mathscr{E}) = E_{\mathbf{y}}\min_i\|\mathscr{E}_i - \mathbf{y}\|_1,$$

where $E_{\mathbf{y}}$ denotes expectation over a uniformly random choice of $y \in \{0, 1\}^\ell$, and for all N, ℓ, we defined $R_{N,\ell} = \min_{\mathscr{E}} R(\mathscr{E})$, where the minimum is over all $\mathscr{E} \subseteq \{0, 1\}^\ell$ of cardinality N.

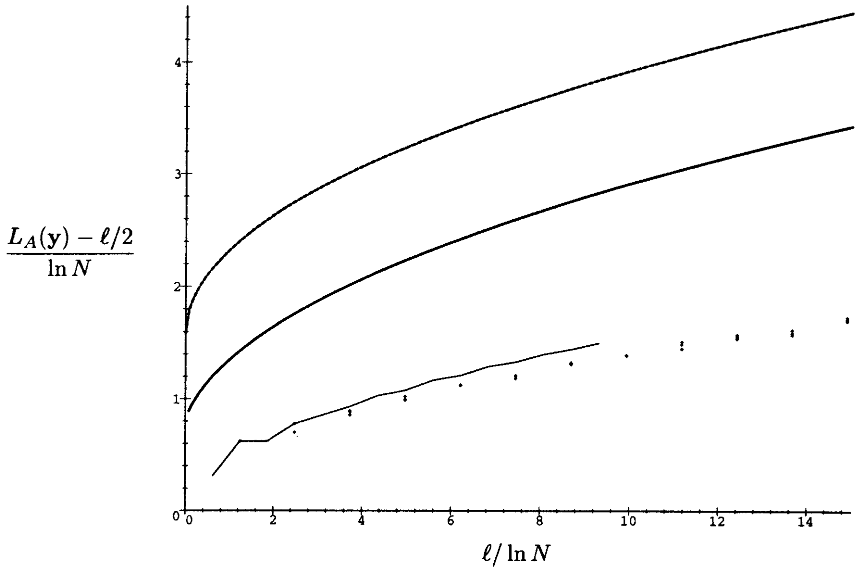

FIG. 6. This figure describes the relationship between the min/max value for $N = 4$ (the piece-wise linear graph) and the lower bound achieved by randomly selected static experts (the cross marks). Three different random choices are given for each selected sequence length in order to provide an estimate of the spread of this statistical lower bound.

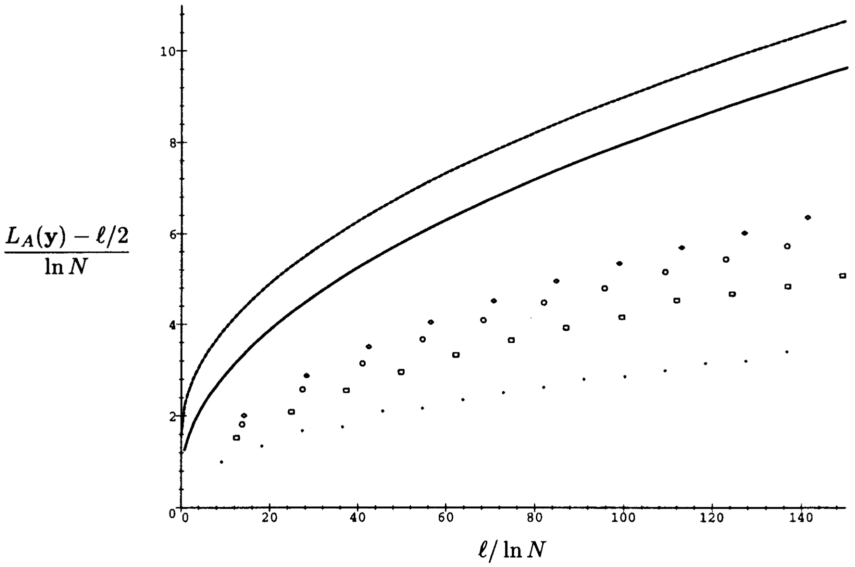

FIG. 7. This figure describes the relationship between randomly generated lower bounds and the upper bounds for longer sequences. The cross, square, circle and diamond marks correspond to the lower bounds for $N = 2, 4, 8, 16$, respectively.

COROLLARY 4.5.3. *For all N, ℓ,*

$$R_{N,\ell} \geq \frac{\ell}{2} - \sqrt{\frac{\ell \, ln(N+1)}{2}} - \frac{log_2(N+1)}{2}$$

```
Algorithm P*(a, c):

Parameters $a > 0$ and $c > 1$ are constants.    $\left\{\text{good choices are } a = 2 \text{ and } c = \left(\frac{1+\sqrt{5}}{2}\right)^2\right\}$
for $z := 0$ to $\infty$ do                  {$z$ is the loop iteration counter}
    $k_2 := a^2c^z \ln N$;                   {guess a bound on best expert's loss}
    $b_z := k_z + \sqrt{k_z \ln N} + \frac{\log_2 N}{2}$    {loss bound if guess correct}
    Reset the weight of each expert to 1.
    repeat
        run $\mathbf{P}(g(\sqrt{\ln(N)/k_z}))$ to generate a prediction
    until the total loss in this loop exceeds $b_z$.
```

FIG. 8. Description of Algorithm **P***.

and

$$\mathit{lim}_{N\to\infty}\ \mathit{lim}_{\ell\to\infty} \frac{(\ell/2) - R_{N,\ell}}{\sqrt{(\ell/2)\ \mathit{ln}\ N}} = 1$$

PROOF. Follows from Theorems 4.5.2 and 3.1.2, since $V_{N,\ell}^{(static)} \leq V_{N,\ell}$. □

4.6. PREDICTION WITHOUT PRIOR KNOWLEDGE. In the previous sections, we showed how to tune β so that $\mathbf{P}(\beta)$ (or, more precisely, its slight variant $\mathbf{P}'(\beta)$ performs well when either a bound on the loss of the best expert or the length ℓ of the sequence is known to the algorithm. Here, we present a version of the algorithm, algorithm **P***, that uses neither the length of the sequence nor the loss of the best expert. Algorithm **P*** repeatedly guesses different loss bounds until it guesses a bound greater than the remaining loss of the best expert. The gap between this algorithm's loss and the loss of the best expert is only a factor of (roughly) 4 greater than the gap when the loss of the best expert is known.

Algorithm **P*** (see Figure 8) takes two parameters, a and c, which control how it guesses loss bounds. We show later that one reasonable choice for these parameters is $a = 2$ and $c = ((1 + \sqrt{5})/2)^2$.

At the start of each iteration z of the outer loop, a bound k_z on the best expert's remaining loss is guessed. Algorithm **P*** resets the experts' weights to 1 and uses algorithm $\mathbf{P}(g(\sqrt{(\ln N)/k_z}))$ (for the function g defined in Eq. (12)) to generate predictions. If the bound k_z is correct, then the remaining loss will be no greater than a value b_z calculated using Theorem 4.4.3. If the total loss incurred by algorithm **P** during the iteration exceeds b_z, then the guessed bound on the loss of the best expert is incorrect[10] and algorithm **P*** increases the guessed bound by a factor of c and proceeds to the next iteration of the outer loop. Note that the first iteration is iteration number zero ($z = 0$).

Before analyzing algorithm **P***, we state a few simple facts that will be needed. First, from the description of the algorithm,

[10] The bounds of this section also hold if instead we use the following stopping criterion: "Until the loss of the best expert in this loop exceeds k_z."

$$
\begin{aligned}
b_z &= k_z + \sqrt{k_z \ln N} + \frac{\log_2 N}{2} \\
&= k_z + ac^{z/2} \ln N + \frac{1}{2} \log_2 N \\
&= k_z + \left(ac^{z/2} + \frac{1}{2 \ln 2} \right) \ln N. \qquad (14)
\end{aligned}
$$

Also, since at most one unit of loss is incurred by any prediction, the loss incurred by algorithm **P*** during any iteration number z of the outer loop is at most $b_z + 1$.

LEMMA 4.6.1. *If algorithm* **P*** *exits iteration number z of the outer loop, then, for all* $\mathscr{E}_i \in \mathscr{E}$, *the loss incurred by* $\mathscr{E}_i$ *while algorithm* **P*** *is executing iteration number z of the outer loop is greater than* k_z.

PROOF. If some expert incurs loss at most k_z during loop iteration number z, then algorithm **P** has loss at most b_z during this iteration (by Theorem 4.4.3) and iteration number z is not exited. □

Let $\mathbf{y}_z$ be the subsequence of outcomes seen during iteration number z of the outer loop. The loss of an expert $\mathscr{E}_i$ while algorithm **P*** is executing iteration number z may not be the same as $L_{\mathscr{E}_i}(\mathbf{y}_z)$. This is because the experts can be algorithms whose state changes based on the outcomes seen. Expert $\mathscr{E}_i$ may make different predictions on $\mathbf{y}_z$ after having seen the outcomes in previous loop iterations than it would make on $\mathbf{y}_z$ without having seen the other outcomes. It is important that we reset only the weights of the experts that are maintained by **P** and *not* the internal states of the experts before calling algorithm **P** as we want to compare the loss of **P*** with $L_{\mathscr{E}}(\mathbf{y})$.

LEMMA 4.6.2. *Pick any* $a > 0$ *and* $c > 1$. *If "last" is the number of the last loop iteration entered by* **P***(a, c) *on some sequence* $\mathbf{y}$, *then*

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

PROOF. If *last* $= 0$, then the lemma trivially holds, so we continue under the assumption that *last* ≥ 1. If iteration number z of the outer loop is exited when algorithm **P*** runs on sequence $\mathbf{y}$ then

$$
L_{\mathscr{E}}(\mathbf{y}) > \sum_{j=0}^{z} k_j = \sum_{j=0}^{z} a^2 c^j \ln N = a^2 (\ln N) \frac{c^{z+1} - 1}{c - 1}.
$$

Since *last* ≥ 1 and iteration number last is entered, iteration number *last* $- 1$ is exited. Thus,

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

Solving for *last* yields the desired result. □

The above lemma shows that algorithm **P*** executes the outer loop a finite number of times whenever the loss of the best expert is bounded. Thus, our bounds on algorithm **P*** hold even for infinite sequences, as long as the loss of the best expert is finite over the infinite sequence.

We now return to bounding the total loss of algorithm **P***.

THEOREM 4.6.3. *Let $\mathcal{E}$ be a set of N experts*, **y** *be any sequence, and* ϕ *be the golden ratio* $(1 + \sqrt{5})/2$. *If* $L_{\mathcal{E}}(\mathbf{y})$ *is finite, then for all*

$$a \geq \frac{2(\phi - 1)}{(2 - \sqrt{\phi})\, ln\ N},$$

the difference $L_{P*}(\mathbf{y}) - L_{\mathcal{E}}(\mathbf{y})$ *is at most*

$$\left(\frac{\phi^{3/2}}{\phi - 1} + \frac{0.805\sqrt{\phi}}{4a(ln\ 2)(ln\ \phi)} + \frac{0.805\sqrt{\phi}}{2a(ln\ N)(ln\ \phi)}\right)\sqrt{L_{\mathcal{E}}(\mathbf{y})\, ln\ N} + \left(a + \frac{1}{2\, ln\ 2}\right) ln\ N,$$

when algorithm **P*** *uses parameters* $c = \phi^2$ *and a.*

PROOF. In Appendix C. □

COROLLARY 4.6.4. *If* $N \geq 7$ *and algorithm* **P*** *uses parameters* $c = \phi^2$ *and* $a = 2$, *then for any sequence* **y**,

$$L_{P*}(\mathbf{y}) - L_{\mathcal{E}}(\mathbf{y}) \leq 4\sqrt{L_{\mathcal{E}}(\mathbf{y})\ ln\ N} + 2.8\ ln\ N.$$

Note that the parameter a allows one to trade off (in a limited way) between the constant in front of the ln N term and the constant in front of the $\sqrt{L_{\mathcal{E}}(\mathbf{y}) \ln N}$ term. Furthermore, the constant multiplying the (more important) $\sqrt{L_{\mathcal{E}}(\mathbf{y}) \ln N}$ term can be made arbitrarily close to $\phi^{3/2}/(\phi - 1) \approx 10/3$ by choosing the constant a sufficiently large.

Since the algorithm **P*** is not given the length of the sequence **y**, the bound of Theorem 4.6.3 holds for all prefixes **y** of any infinite sequence **y′**. Different experts might have minimum loss for different prefixes of **y′**, but the loss of **P*** is always close to the best expert on each prefix.

5. *Applications to the Pattern Recognition Problem*

Up until this point we discussed the problem of predicting binary sequences, where the predictions made by the experts are functions of past predictions and outcomes. We turn now to an application of these results to the general pattern recognition problem as was described in the introduction.

Our goal is to approximate a stochastic mapping from an *instance space X* to *labels* {0, 1}. The algorithm observes a set of examples of the stochastic mapping and produces a *hypothesis*, a rule for predicting the labels of new instances. The goal of the learning algorithm is to produce a hypothesis whose error (i.e., probability of mistake) is not much worse than the error of the best function in some known class $\mathcal{H}$ of functions called the *comparison* or *touchstone class* [Kearns et al. 1994]. Outside of the pattern recognition literature, this type of problem might be called by many names, such as L_1 regression with a regret formulation of the loss function (in typical statistics literature, see, for example, Birge and Massart [1993]), or, as mentioned in the introduction, the agnostic version of PAC learning [Kearns et al. 1994]. The terminology we use here is that from the PAC learning literature.

More formally, let D be a probability distribution on $X \times \{0, 1\}$.[11] We assume a sequence $\mathbf{s} = (x_1, y_1), \ldots, (x_\ell, y_\ell)$ of *training examples* is drawn from the product distribution D^ℓ, that is, each example is drawn independently according to D. A *learning algorithm* A, which does not know the distribution D, takes these training examples as input and outputs a hypothesis $h = A(\mathbf{s})$ that maps from X into $[0, 1]$. The *error* of the hypothesis h is defined by $\mathrm{er}_D(h) = E_{(x,y)\sim D}|h(x) - y|$, where $E_{(x,y)\sim D}$ denotes the expectation over (x, y) drawn randomly according to D.

The learning algorithm is given *a priori* a comparison class $\mathcal{H}$ consisting of a set of mappings from X into $\{0, 1\}$. The functions in the comparison class play a role similar to that played by the experts above. However, while the experts defined in Section 4 are arbitrary prediction strategies, the comparison class contains only fixed functions that do not depend on past predictions and outcomes. Also, we restrict these functions to output either 0 or 1 and not real numbers in the range [0, 1]. On the other hand, the comparison class may be infinite, while the set of experts in Section 4 is assumed to be finite.

Let

$$\mathrm{er}_D(\mathcal{H}) = \inf_{h \in \mathcal{H}} \mathrm{er}_D(h)$$

be the error of the best function in $\mathcal{H}$ for the particular distribution D. The goal of the learning algorithm is to, on average, produce a hypothesis that is almost as good as the best function in the comparison class $\mathcal{H}$ for examples generated by the (unknown) distribution D. That is, the learning algorithm attempts to minimize[12] the regret

$$E_{\mathbf{s}\sim D^\ell}(\mathrm{er}_D(A(\mathbf{s}))) - \mathrm{er}_D(\mathcal{H}). \qquad (15)$$

Bounds on this regret for certain types of learning algorithms can be obtained from the work of Vapnik [1982] and Birge and Massart [1993]. The basic idea of their learning algorithms is to predict according to the *single* hypothesis that suffers the minimal loss over the sample of instances presented to the learner. Vapnik calls this *empirical risk minimization*. In this paper, we obtain better performance bounds by using an algorithm that combines the predictions of *all* the experts, weighted according to their performance on the sample.

We now sketch how the techniques developed in Section 4 for the sequence prediction problem can be applied to the pattern recognition problem. Suppose that $\mathbf{s} = (x_1, y_1), \ldots, (x_\ell, y_\ell)$ is the sequence of random labeled examples

[11] When X is uncountable, appropriate assumptions are made to ensure measurability in what follows.

[12] Typically, in the PAC learning literature, 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 [Haussler et al. 1991; Littlestone 1989]. More direct methods are given by Littlestone and Warmuth [1994].

presented to the learning algorithm, and let x be an instance whose label is to be predicted. The natural way of using a sequence prediction algorithm, such as the algorithm **P**, in this context is to simulate it on the sequence **s**, and then obtain its prediction on the new instance x. Here we regard as experts the set of all possible labelings of the instances $x_1, \ldots, x_\ell, x$ that agree with some function in the comparison class $\mathcal{H}$. Although the cardinality of $\mathcal{H}$ may be infinite, the number of possible binary labelings of the sequence that agree with some function in $\mathcal{H}$ is always finite, and in fact, is polynomial in ℓ if the VC dimension of $\mathcal{H}$ is finite (see Blumer et al. [1989] or Vapnik [1982] for a definition of the VC dimension and its relation to this kind of learning problem).

Unfortunately, we do not know how to analyze an algorithm of this type, since the bounds that we have for our sequence prediction algorithms hold only for the *cumulative* loss over the entire sequence, and not the loss at any particular time step. To handle this difficulty, we define a more complicated scheme that uses the sequence prediction algorithm in a more elaborate way. Instead of placing the unlabeled example at the end of the sequence, we insert it in all possible positions in the sequence **s** and take the average of the predictions so obtained. More precisely, for every choice of index $i = 0, \ldots, \ell$, we insert the unlabeled example between examples i and $i + 1$, producing the sequence $(x_1, y_1), \ldots, (x_i, y_i), (x, ?), (x_{i+1}, y_{i+1}), \ldots, (x_\ell, y_\ell)$. We simulate our prediction algorithm **P** on each of these sequences to obtain $\ell + 1$ predictions of x's label and output their average. A simple argument, which will be given in Section 5.2, bounds the expected error of this learning algorithm. Similar methods were previously used by Helmbold and Warmuth [1995].

Before using algorithm **P** as the sequence prediction algorithm, we need to choose the parameter β. We analyze two methods for tuning β in this context. The first method is to tune β according to the length of the sample, using the results of Section 4.5. These results are described in Section 5.2. The drawback of this method is that the dependence of the regret of the learning algorithm on the sample size ℓ is of order $O(1/\sqrt{\ell})$ even if the loss of the best function in $\mathcal{H}$ is very small. By using a much more sophisticated choice of β we can improve the upper bound on the regret to $O(1/\ell)$ when $\mathrm{er}_D(\mathcal{H})$ is small. These results are described in Section 5.3.

5.1. Further Definitions. Before stating our results, we need to make a few further 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 *over-fitting*. Let

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

Thus, $\mathrm{er}_{\ell,D}\,(\mathcal{H})$ 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

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

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

$$\begin{aligned} \mathrm{er}_D(\mathcal{H}) &= \inf_{h\in\mathcal{H}} \mathrm{er}_D(h) \\ &= \inf_{h\in\mathcal{H}} E_{\mathbf{s}\sim D^\ell} \frac{1}{\ell} \sum_{t=1}^{\ell} |h(x_t) - y_t| \\ &\geq E_{\mathbf{s}\sim D^\ell} \inf_{h\in\mathcal{H}} \frac{1}{\ell} \sum_{t=1}^{\ell} |h(x_t) - y_t| \\ &= \hat{\mathrm{er}}_{\ell,D}(\mathcal{H}). \end{aligned}$$

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.

We also will need a formal notation for the set of all label sequences that agree with some function in $\mathcal{H}$. For any comparison class $\mathcal{H}$ 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 call $\mathcal{H}_{|\mathbf{x}}$ the *restriction of* $\mathcal{H}$ to $\mathbf{x}$.

5.2. The Basic Bound

Theorem 5.2.1. *For any instance space X and any comparison class $\mathcal{H}$ on X, there exists a learning algorithm A such that for all ℓ and all distributions D on $X \times \{0, 1\}$*

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

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.

Proof. We define the learning algorithm A by describing its hypothesis, h. Given the sequence of examples $\mathbf{s} = (x_1, y_1), \ldots, (x_\ell, y_\ell)$, and instance x, we define $h(x)$ as follows: First, for each $1 \leq t \leq \ell + 1$, let $\mathbf{x}^{(t)} = x_1, \ldots, x_{t-1}, x, x_t, \ldots, x_\ell$ and let $\mathscr{E}^{(t)} = \mathcal{H}_{|\mathbf{x}^{(t)}}$. Thus, there is an expert in $\mathscr{E}^{(t)}$ for each possible labeling of $\mathbf{x}^{(t)}$ that agrees with some function in the comparison class $\mathcal{H}$. Note that the experts in $\mathscr{E}^{(t)}$ are the same as the experts in $\mathscr{E}^{(t+1)}$ except that the predictions on trials t and $t + 1$ are swapped due to the different placement of x. Let $N = |\mathscr{E}^{(t)}|$ and $\beta = g(\sqrt{2\ \ln(N + 1)/(\ell + 1)})$. For each $1 \leq t \leq \ell + 1$ let $\hat{y}_t$ denote the prediction of the sequence prediction algorithm $\mathbf{P}'(\beta)$ defined in Section 4.5 after seeing outcomes $y_1, \ldots, y_{t-1}$, and the first t predictions of the experts in $\mathscr{E}^{(t)}$. The value of the function $h = A(\mathbf{s})$ on input x is defined by the average of the $\hat{y}_t$'s, that is, $h(x) = 1/(\ell + 1) \quad {}_{t=1}^{\ell+1} \hat{y}_t$.

To show that this strategy A has the desired performance, first note that

$$\begin{aligned} E_{\mathbf{s}\sim D^\ell}(\mathrm{er}_D(A(\mathbf{s}))) &= E_{\mathbf{s}\sim D^\ell,(x,y)\sim D}|A(\mathbf{s})(x) - y| \\ &= E_{\mathbf{s}\sim D^\ell,(x,y)\sim D}\left|\left(\frac{1}{\ell + 1}\sum_{t=1}^{\ell+1} \hat{y}_t\right) - y\right|, \end{aligned}$$

where $\hat{y}_t$ is as defined in the previous paragraph, and $\mathbf{s} = (x_1, y_1), \ldots, (x_\ell, y_\ell)$.

Because $|(1/n \quad {}_{t=1}^{n} p_t) - c| = 1/n \quad {}_{t=1}^{n} |p_t - c|$ for $c \in \{0, 1\}$ and $0 \le p_t \le 1$, it follows that

$$\begin{aligned} E_{\mathbf{s}\sim D^\ell}(\mathrm{er}_D(A(\mathbf{s}))) &= E_{\mathbf{s}\sim D^\ell,(x,y)\sim D}\, \frac{1}{\ell + 1}\sum_{t=1}^{\ell+1} |\hat{y}_t - y| \qquad (16) \\ &= \frac{1}{\ell + 1}\sum_{t=1}^{\ell+1} E_{\mathbf{s}\sim D^\ell,(x,y)\sim D}|\hat{y}_t - y| \\ &= \frac{1}{\ell + 1}\sum_{t=1}^{\ell+1} E_{(\mathbf{x},\mathbf{y})\sim D^{\ell+1}}|\hat{y}'_t - y_t|, \qquad (17) \end{aligned}$$

where, in analogy with the definition of $\hat{y}_t$, we define $\hat{y}'_t$ as the prediction of $\mathbf{P}'(\beta)$ after observing the outcomes $y_1, \ldots, y_{t-1}$ and the first t predictions of the experts in $\mathscr{H}_{|\mathbf{x}}$, where $\mathbf{x} = x_1, \ldots, x_{\ell+1}$, and $\beta = g(\sqrt{2 \ln(|\mathscr{H}_{|\mathbf{x}}| + 1)/(\ell + 1)})$.

Let $L_{\mathbf{P}'(\beta)}(\mathbf{x}, \mathbf{y}) = \quad {}_{t=1}^{\ell+1} |\hat{y}'_t - y_t|$, the total loss of the prediction strategy $\mathbf{P}'(\beta)$ for instances $\mathbf{x} = x_1, \ldots, x_{\ell+1}$ and outcomes $\mathbf{y} = y_1, \ldots, y_{\ell+1}$, assuming the set of experts is $\mathscr{H}_{|\mathbf{x}}$. It follows from the above that

$$E_{\mathbf{s}\sim D^\ell}(\mathrm{er}_D(A(\mathbf{s}))) = \frac{1}{\ell + 1} E_{(\mathbf{x},\mathbf{y})\sim D^{\ell+1}} L_{\mathbf{P}'(\beta)}(\mathbf{x},\mathbf{y}). \qquad (18)$$

Furthermore, it is clear that for all ℓ

$$\begin{aligned} \widehat{\mathrm{er}}_{\ell,D}(\mathscr{H}) &= E_{(\mathbf{x},\mathbf{y})\sim D^\ell}\, \frac{1}{\ell} \inf_{h\in\mathscr{H}} \sum_{t=1}^{\ell} |h(x_t) - y_t| \qquad (19) \\ &= E_{(\mathbf{x},\mathbf{y})\sim D^\ell}\, \frac{1}{\ell} L_{\mathscr{H}_{|\mathbf{x}}}(\mathbf{x},\mathbf{y}), \end{aligned}$$

where $L_{\mathscr{H}_{|\mathbf{x}}}(\mathbf{x}, \mathbf{y})$ is the total loss of the best expert in $\mathscr{H}_{|\mathbf{x}}$ on the outcome sequence $\mathbf{y}$.

It follows from Eqs. (18) and (19) and the definition of expected over-fit that

$$\begin{aligned} &E_{\mathbf{s}\sim D^\ell}(\mathrm{er}_D(A(\mathbf{s}))) - \mathrm{er}_D(\mathscr{H}) \\ &\quad = E_{\mathbf{s}\sim D^\ell}(\mathrm{er}_D(A(\mathbf{s}))) - \widehat{\mathrm{er}}_{\ell+1,D}(\mathscr{H}) - (\mathrm{er}_D(\mathscr{H}) - \widehat{\mathrm{er}}_{\ell+1,D}(\mathscr{H}) \end{aligned}$$

$$= \frac{1}{\ell + 1} E_{(\mathbf{x},\mathbf{y})\sim D^{\ell+1}} L_{\mathbf{P}'(\beta)}(\mathbf{x},\mathbf{y}) - \frac{1}{\ell + 1} E_{(\mathbf{x},\mathbf{y})\sim D^{\ell+1}} L_{\mathcal{H}_{|\mathbf{x}}}(\mathbf{x},\mathbf{y}) - \mathrm{er}^{\Delta}_{\ell+1,D}\ (\mathcal{H})$$

$$= \frac{1}{\ell + 1} E_{(\mathbf{x},\mathbf{y})\sim D^{\ell+1}} (L_{\mathbf{P}'(\beta)}(\mathbf{x},\mathbf{y}) - L_{\mathcal{H}_{|\mathbf{x}}}(\mathbf{x},\mathbf{y})) - er^{\Delta}_{\ell+1,D}(\mathcal{H}).$$

By Theorem 4.5.1., for any $\mathbf{x}$ and $\mathbf{y}$ of length $\ell + 1$,

$$L_{\mathbf{P}'(\beta)}(\mathbf{x},\mathbf{y}) - L_{\mathcal{H}_{|\mathbf{x}}}(\mathbf{x},\mathbf{y}) \le \sqrt{\frac{(\ell + 1)\ln(|\mathcal{H}_{|\mathbf{x}}| + 1)}{2}} + \frac{\log_2(|\mathcal{H}_{|\mathbf{x}}| + 1)}{2}.$$

The result follows. □

It is easy to see that the constant in the leading term of the bound in Theorem 5.2.1 is the best possible. The argument is similar to the lower bound argument we used for prediction strategies. 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 $E_{\mathbf{s}\sim D^\ell}\ (\mathrm{er}_D(A(\mathbf{s}))) - \mathrm{er}_D(\mathcal{H}) = 0$ for any comparison class $\mathcal{H}$ and algorithm A.

Now assume in addition that X is a large finite set and the marginal of D on X has a uniform distribution. Let us choose each of the N functions $h_1, \ldots, h_N$ to be included in the comparison class $\mathcal{H}$ at random by letting $h_i(x) = 1$ with probability 1/2 and $h_i(x) = 0$ with probability 1/2 independently for each i, $1 \le i \le N$, and each instance $x \in X$. Then Lemma 3.2.1 implies that for any fixed sample size $\ell + 1$, in the limit of large X, the expectation (with respect to the random choice of $\mathcal{H}$) of the expected over-fit $\mathrm{er}^{\Delta}_{\ell+1,D}(\mathcal{H})$ is $(1 + o(1))\ \sqrt{\ln N}/\sqrt{2\ell}$. This is because in this limit all the $x_1, \ldots, x_{\ell+1}$ are distinct with probability one, and the values $|h_i(x_t) - y_t|$ are distributed like independent coin flips for $1 \le i \le N$ and $1 \le t \le \ell + 1$. It follows that there exists a sequence of comparison classes $\mathcal{H}$ such that the expected over-fit $\mathrm{er}^{\Delta}_{\ell+1,D}$ $(\mathcal{H})$ is $(1 + O(1))\ \sqrt{\ln N}/\sqrt{2\ell}$.

The expected over-fit appears with a minus sign on the right-hand side of the bound in Theorem 5.2.1. 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.

5.3. Refined Result. The result of the previous theorem can be improved by a more sophisticated choice of β.

Theorem 5.3.1. *For any instance space X and any comparison class $\mathcal{H}$ on X, there exists a learning algorithm A such that for all ℓ and all distributions D on $X \times$* $\{0, 1\}$

$$E_{\mathbf{s}\sim D^\ell}(er_D(A(\mathbf{s}))) - er_D(\mathcal{H})$$
$$\le \frac{\sqrt{\hat{er}_{\ell+1,D}(\mathcal{H})(}\sqrt{T} + 1)}{\sqrt{\ell + 1}} + \frac{T/ln\ 2 + 3\sqrt{T} + 1}{\ell + 1} - er^{\Delta}_{\ell+1,D}(\mathcal{H}) \qquad (20)$$

$$\leq \frac{\sqrt{er_D(\mathcal{H})}\ (\sqrt{T}+1)}{\sqrt{\ell+1}} + \frac{T/ln\ 2 + 3\sqrt{T}+1}{\ell+1} - er^{\Delta}_{\ell+1,D}(\mathcal{H}), \tag{21}$$

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

The proof of this theorem is given in the next section. Our first attempt to prove it followed the proof of the previous theorem with the different choice $\beta = g(\sqrt{(\ln N)/K})$, where K is the best upper bound that can be obtained on the total loss of the best expert in $\mathcal{E}^{(t)}$. Then, in the last step, Theorem 4.4.3 is used instead of Theorem 4.5.1. Since we know all the predictions of the experts and all the outcomes but the one for the instance x, we can estimate the total loss of the best expert to within 1, and choose β accordingly. It remains an open problem to prove a bound on the regret for this approach that is comparable to the bound given in Theorem 5.3.1.

The subtle difficulty we encountered in trying to prove such a bound is in moving from Eq. (16) to Eq. (17). In Eq. (16), $\hat{y}_t$ is the prediction made by the algorithm on the additional instance (x, y) when it is inserted into position t of sequence $\mathbf{s}$. Thus, $\hat{y}_t$ depends on the previous elements of the sequence, the current predictions of the experts, and the choice of β. In Theorem 5.2.1, β is a fixed function of the length of the sequence, and thus the prediction $\hat{y}_t$ is identical to the prediction made by $\mathbf{P}(\beta)$. This is why we can replace $\hat{y}_t$ by $\hat{y}'_t$.

Unfortunately, when we choose β as a function of the examples in $\mathbf{s}$, this substitution of $\hat{y}'_t$ for $\hat{y}_t$ is impossible. Because a different β is chosen for each position t, the sequence of predictions $\hat{y}'_t$ no longer corresponds to the predictions generated by a single run of $\mathbf{P}(\beta)$, and so we cannot derive Eq. (18). (Recall that the performance bound on $\mathbf{P}(\beta)$ requires that β is held constant.)

There are several ways one could attempt to patch this flaw, but despite much effort we were unable to find a simple fix. The approach that was ultimately successful deals directly with prediction when all but one outcome is available. This setting is reminiscent of that obtained when using the "hold-one-out" method of cross validation, commonly used in statistics. Results for this setting are given in the next section, as is the proof of Theorem 5.3.1.

The bounds given in Theorem 5.3.1 are better than those obtained for this kind of pattern recognition problem by the other methods of which we are aware, which are those of Vapnik [1992], Talagrand [1994], and Birge and Massart [1993]. Bounds given by Vapnik [1992, Eq. (11)] imply a bound in the same form as the second bound in Theorem 5.3.1, but with an additional factor of 2 in the leading term. However, Vapnik's bounds hold in more general cases than the one we consider here. Talagrand [1994] gives similar general bounds without the factor of 2, but with an unspecified constant in the lower-order term. It is not clear that this unspecified constant can be made small enough to get practical bounds for small sample size ℓ. Bounds obtained by Birge and Massart [1993] also contain constants that are difficult to bound. Thus, our approach to the pattern recognition problem through worst-case analysis of the sequence prediction problem appears to be a fruitful one.

5.4. The Hold-One-Out Model of Prediction and Proof of Theorem 5.3.1. In this subsection, we discuss a slightly different prediction problem. After developing a theory of this prediction problem, we will be in a position to prove Theorem 5.3.1.

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 $\mathscr{E} = \{\mathscr{E}_1, \ldots, \mathscr{E}_N\}$ be a set of experts. In this section, we will assume that each expert $\mathscr{E}_i$ is a function from X into [0, 1], that is, the ith expert's prediction at time t, denoted $\xi_{i,t}$, depends only on the instance x_t, and not on previous outcomes or instances. As in Section 3.1, we call such experts *static*.[13] For a fixed sequence $\mathbf{x}$ of instances, they are equivalent to the static experts defined there. As in the previous sections, the total loss of the ith expert is $L_{\mathscr{E}_i}(\mathbf{x}, \mathbf{y}) = \sum_{t=1}^{\ell} |\xi_{i,t} - y_t|$, and the total loss of the best expert is $L_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) = \min_{1 \le i \le N} L_{\mathscr{E}_i}(\mathbf{x}, \mathbf{y})$.

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 $\mathscr{E}_{i,t}$, $1 \le i \le N$, $1 \le t \le \ell$, giving the advice of each expert on each instance, and the outcomes $y_1, \ldots, y_{t-1}, y_{t+1}, \ldots, y_\ell$, that is, all outcomes except y_t. Given this input, a hold-one-out prediction algorithm produces a prediction $\hat{y}_t \in [0, 1]$. The *total hold-one-out* loss of the prediction algorithm A on outcome sequence $\mathbf{y}$ is defined in analogy with the on-line prediction loss as $HL_A(\mathbf{x}, \mathbf{y}) = \sum_{t=1}^{\ell} |\hat{y}_t - y_t|$. 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. The motivation for the name "hold-one-out" loss comes from the similarity to the cross-validation procedure of the same name used in statistics [Stone 1977].

The following example illustrates the use of the total hold-one-out loss. Consider a classroom setting in which an instructor is trying to teach students to perform a classification task of some type, say to distinguish earthquakes from underground nuclear explosions, based on seismographic data. Suppose that the teacher has collected a sequence of labeled examples $(x_1, y_1), \ldots, (x_\ell, y_\ell)$, where for each t, $1 \le t \le \ell$, the instance x_t is a vector of seismic measurements and the label y_t is a binary value, with 1 representing earthquake and 0 representing underground explosion. Let $\mathbf{x} = x_1, \ldots, x_\ell$ and $\mathbf{y} = y_1, \ldots, y_\ell$. The teacher shows each of the examples to the students (the experts in this example), in random order, first showing them the measurement vector x_t, then asking each student to predict the classification y_t, and finally providing actual label y_t as feedback. A prediction is a number $p \in [0, 1]$ and the loss is $|p - y_t|$ as above. However, instead of considering total loss, here the teacher only counts the loss on the last example shown, considering the other examples to be merely training cases. The choice of which example is shown last (called the "test" example) is random. Now imagine that you are auditing the class because of your extremely limited knowledge of seismology. Nevertheless, you still want to impress the teacher in hopes of eventually being admitted to the program. Can

[13] Thus, a static expert is simply a regression function (or "p-concept" [Kearns and Schapire 1994]) from the instance space X into [0, 1], the value of which represents a conditional probability of the label 1 given the input instance x_t.

you or any algorithm A, after seeing all the instances $x_1, \ldots, x_\ell$, hearing all the students' predictions for each of these instances, including the test instance, and seeing all the labels except that of the test instance, predict the label of the test instance in such a way that your expected loss, averaged over possible choices of the test instance, is not much more than that of the best student in the class?

Instead of averaging over all choices of the last instance, we can equivalently consider the experiment in which the examples stay in the fixed order $(x_1, y_1), \ldots, (x_\ell, y_\ell)$, but for t from 1 to ℓ we perform a series of experiments with the algorithm A, each time covering only the label y_t and forcing the algorithm to predict this label, based on the ℓ instances, the prediction of each expert on each instance, and the label of all the instances except x_t. Clearly, the total hold-one-out loss $HL_A(\mathbf{x}, \mathbf{y})$ is the total loss obtained by all these experiments. Thus, the average loss of the algorithm in predicting a randomly chosen test instance is just $HL_A(\mathbf{x}, \mathbf{y})/\ell$.

Note that we have restricted our analysis of the hold-one-out loss to the case of static experts. For this type of loss, we must be careful about how much power we give the experts. Consider the case in which there are just two experts $\mathcal{E}_0$ and $\mathcal{E}_1$, and $\mathcal{E}_0$ always predicts that the sequence of binary values $\mathbf{y} = y_1, \ldots, y_\ell$ will have even parity, while $\mathcal{E}_1$ always predicts that $\mathbf{y}$ will have odd parity. Clearly, the predictions of each of these experts for y_t can easily be expressed as a function of the values $y_1, \ldots, y_{t-1}, y_{t+1}, \ldots, y_\ell$, ignoring the instances. Moreover, any sequence $\mathbf{y}$ either has even or odd parity. Thus, for any sequence $\mathbf{y}$, one of the two experts predicts each held out label correctly! Yet for any prediction algorithm A there is always a sequence that forces total loss $\ell/2$, since this is the average loss obtained on a random sequence. It is thus clear that to get a useful worst-case model in the hold-one-out setting, one needs to restrict the experts. Restricting to static experts is one natural choice.

It should be 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 that take into account all the information that is available. Curiously, this is not true, at least in the worst case, as we show below.

Let us define the hold-one-out prediction game for a given N and ℓ by assuming that the adversary chooses a set $\mathcal{E}$ of N static experts, a sequence $\mathbf{x}$ of ℓ instances and a sequence $\mathbf{y}$ of ℓ outcomes, and then the predictor is given ℓ separate prediction problems based on these choices, where in each problem a different outcome is held out and must be predicted on the basis of the other information as described above. Let $V_{N,\ell}^{(H)}$ denote the min/max value of this game, that is, the minimum over all hold-one-out prediction strategies A of the maximum over all choices of the adversary of the difference $HL_A(\mathbf{x}, \mathbf{y}) - L_{\mathcal{E}}(\mathbf{x}, \mathbf{y})$. It turns out that this min/max value is the same as that of the on-line prediction game with static experts given in Theorem 3.1.2.

Before we state the analog of Theorem 3.1.2 for the hold-one-out prediction game, recall that we defined the average covering radius of $S \subseteq \{0, 1\}^\ell$ as $R(S) = E_{\mathbf{y}} \min_{\mathbf{s} \in S} \|\mathbf{s} - \mathbf{y}\|_1$, where $E_{\mathbf{y}}$ denotes expectation over a uniformly random choice of $y \in \{0, 1\}^{\ell}$, and that for any set of functions $\mathscr{E}$ from X into [0, 1] and any sequence $\mathbf{x} = x_1, \ldots, x_\ell$ of instances in X, we defined $\mathscr{E}_{|\mathbf{x}} = \{(f(x_1), \ldots, f(x_\ell)) : f \in \mathscr{E}\}$.

THEOREM 5.4.1. *Let $\mathscr{E}$ be a set of static experts and $\mathbf{x}$ be a sequence of ℓ instances. Then there exists a hold-one-out prediction strategy A such that for every sequence $\mathbf{y}$, we have*

$$HL_A(\mathbf{x}\ ,\mathbf{y}) - L_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) = \frac{\ell}{2} - R(\mathscr{E}_{|\mathbf{x}}).$$

Moreover, A is optimal in the sense that for every hold-one-out prediction strategy B, there exists a sequence $\mathbf{y}$ such that

$$HL_B(\mathbf{x}, \mathbf{y}) - L_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) \geq \frac{\ell}{2} - R(\mathscr{E}_{|\mathbf{x}}).$$

Hence

$$V^{(H)}_{N,\ell} = V^{(static)}_{N,\ell} = \frac{\ell}{2} - \min_{S} R(S),$$

where the minimum is over all sets S of N vectors in $\{0, 1\}^{\ell}$.

PROOF. We simply let A be the optimal on-line prediction strategy **MS** from the proof of Theorem 3.1.2, used as a hold-one-out prediction strategy, ignoring the outcomes $y_{t+1}, \ldots, y_\ell$ when predicting the outcome y_t. Since the net loss $HL_A(\mathbf{x}, \mathbf{y}) - L_{\mathscr{E}}(\mathbf{x}, \mathbf{y})$ is the same for the hold-one-out game as it is for on-line prediction, this gives the first statement of the theorem. The second statement follows from the fact that if $\mathbf{y}$ is chosen at random, then the expectation of HL_B $(\mathbf{x}, \mathbf{y}) - L_{\mathscr{E}}(\mathbf{x}, \mathbf{y})$ is equal to the right-hand-side for any hold-one-out prediction strategy B. Finally, the last statement follows by the same argument used in the proof of Theorem 3.1.2 to prove the analogous statement. □

The optimal algorithm **MS** is not very efficient. We get a simple, efficient, and nearly optimal hold-one-out prediction strategy by using the on-line prediction algorithm **P**. From the above theorem and Theorems 3.2.3 and 4.5.1, we have:

THEOREM 5.4.2. *Let $\mathbf{P}(\beta)$ be the on-line prediction algorithm defined in Section* 4. *For all ℓ and N, if β is chosen to be $g(\sqrt{2\, ln(N+1)/\ell})$, where g is as defined in Eq.* (12), *then for any set $\mathscr{E}$ of N static experts, and any sequences $\mathbf{x}$ and $\mathbf{y}$ of length ℓ, the total hold-one-out loss of $\mathbf{P}$ is bounded by*

$$HL_{\mathbf{P}}(\mathbf{x}, \mathbf{y}) - L_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) \leq \sqrt{\frac{\ell\, ln(N+1)}{2}} + \frac{log_2(N+1)}{2},$$

and the constant in the leading term on the right-hand-side cannot be improved.

When the value $L_{\mathscr{E}}(\mathbf{x}, \mathbf{y})$ is given, we can use algorithm **P** with an appropriately tuned β (as in Theorem 4.4.3) to get a better hold-one-out prediction algorithm. In this case, we get an algorithm that has hold-one-out loss at most $L_{\mathscr{E}}(\mathbf{x}, \mathbf{y})$ +

```
Algorithm B(t):

{The algorithm receives a sequence of instances, $\mathbf{x} = x_1, \ldots, x_\ell$, a sequence of binary outcomes, $\mathbf{y} = y_1, \ldots, y_{t-1}, ?, y_{t+1}, \ldots, y_\ell$, where the $t$th position is marked with a "?", and the predictions $\mathscr{E}_{i,j}$ of each expert $\mathscr{E}_i$ for $1 \le i \le N$ on each instance $x_j$ for $1 \le j \le \ell$. The algorithm produces a prediction $\hat{y}_t$ for the held out outcome $y_t$.}

1. Pick $r \in [0, 1]$ uniformly at random;
2. Compute $L_{\mathrm{obs}}(t) = \min_i \sum_{j \ne t} |\mathscr{E}_{i,j} - y_j|$;
3. Compute $L_{\mathrm{est}}(t) = (\lceil \sqrt{L_{\mathrm{obs}}(t) + 1} - r \rceil + r)^2$;
4. Compute $\beta = g(\sqrt{\ln N / L_{\mathrm{est}}(t)})$, where $g$ is the function defined in (12). Run algorithm $\mathbf{P}(\beta)$ on
   the sequence of instances $x_1, \ldots, x_t$ and observations $y_1, \ldots, y_{t-1}$, and predict with the $\hat{y}_t$ (for $y_t$)
   generated by $\mathbf{P}$.
```

FIG. 9. Description of Algorithm **B** for hold-one-out prediction.

$\sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) \ln N} + \log_2 N/2$. When neither this value nor the length of the sequence is available, algorithm **P***, which iteratively guesses the loss of the best expert, can be used. However, algorithm **P*** ignores the extra information provided and its bound has a factor greater than one multiplying the $\sqrt{L_{\mathscr{E}} \ln N}$ term. It is better to use the observed losses of the experts on the $\ell - 1$ outcomes provided to estimate $L_{\mathscr{E}}(\mathbf{x}, \mathbf{y})$. Unfortunately, we are unable to show that when these estimates are plugged directly into algorithm **P**, a small total loss results. As mentioned in Section 5.3, 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 and let β be a function of the estimate. A little randomization is used to ensure that the estimate is likely to be the same regardless of which label is held out. The resulting algorithm is algorithm **B**, described in Figure 9. The estimated loss is determined in Step (3). We show that for this choice of the estimate, the probability that all of the estimates are the same increases with the loss of the best expert.

Note that the hypothesis of algorithm **B** is probabilistic since it depends on a value r chosen uniformly at random in the interval [0, 1]. It is easy to get a deterministic version of algorithm **B**: Run algorithm **B** q times in parallel, where the ith copy uses the fixed i/q as its choice for r ($0 \le i \le q - 1$). The new deterministic algorithm **DB** simply predicts with the average of the q predictions. We still need to specify the choice of q. As q grows the worst-case loss of algorithm **DB** converges to the expected worst-case loss of algorithm **B**, where the latter expectation is over the uniform choice of $r \in [0, 1]$. We choose

$$q = \ell + \left(\sqrt{\ell + 1} + 1 \right) \sqrt{\ln N} + \frac{\ln N}{2 \ln 2},$$

where ℓ is the number of trials. For this choice, we prove in the theorem below that the worst-case loss of algorithm **DB** is at most one larger than the bound we prove on the worst-case expected loss of algorithm **B**.

THEOREM 5.4.3. *The hold-one-out prediction algorithms* **B** *and* **DB** *have the property that for any* **x**, *any set of static experts* $\mathscr{E}$, *and any sequence* **y**

$$E_{r\sim[0,\,1]}(HL_{\mathbf{B}}(\mathbf{x},\,\mathbf{y})) \le L_{\mathscr{E}}(\mathbf{x},\,\mathbf{y})(\sqrt{\ln N} + 1) + 3\sqrt{\ln N} + log_2\, N \text{ and}$$

$$HL_{\mathbf{DB}}(\mathbf{x},\,\mathbf{y}) \le L_{\mathscr{E}}(\mathbf{x},\,\mathbf{y}) + \sqrt{L_{\mathscr{E}}(\mathbf{x},\,\mathbf{y})}(\sqrt{\ln N} + 1) + 3\sqrt{\ln N} + log_2\, N \ + 1.$$

Recall that in the case when $L_{\mathscr{E}}(\mathbf{x}, \mathbf{y})$ is given to the algorithm, the algorithm $\mathbf{P}$ with its parameter β properly tuned as a function of $L_{\mathscr{E}}(\mathbf{x}, \mathbf{y})$ has hold-one-out loss at most $L_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) + \sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) \ln N} + \log_2 N/2$ (see Theorem 4.4.3). Note that the bounds of the above theorem for algorithms that do not have $L_{\mathscr{E}}(\mathbf{x}, \mathbf{y})$ available are not too much larger. We develop the proof of this theorem in a sequence of lemmas.

LEMMA 5.4.4. *Choose any set of experts $\mathscr{E}$, and sequences $\mathbf{x}$ and $\mathbf{y}$ of length ℓ. For each $r \in [0, 1]$ we have that for all $1 \le t \le \ell$,*

$$L_{est}(t) \in \{L_r^-,\ L_r^+\},$$

where

$$L_r^- = \left(\left\lceil \sqrt{L_{\mathscr{E}}(\mathbf{x},\,\mathbf{y})} - r \right\rceil + r\right)^2 \text{ and } L_r^+ = \left\lceil \left(\sqrt{L_{\mathscr{E}}(\mathbf{x},\,\mathbf{y}) + 1} - r \right\rceil + r \right)^2.$$

PROOF. Since the loss in any trial lies in [0, 1], we have

$$L_{\text{obs}}(t) \le L_{\mathscr{E}}(\mathbf{x},\,\mathbf{y}) \le L_{\text{obs}}(t) + 1,$$

$$L_{\mathscr{E}}(\mathbf{x},\,\mathbf{y}) \le L_{\text{obs}}(t) + 1 \le L_{\mathscr{E}}(\mathbf{x},\,\mathbf{y}) + 1 \text{ and}$$

$$\sqrt{L_{\text{obs}}(t) + 1} \in \left[\sqrt{L_{\mathscr{E}}(\mathbf{x},\,\mathbf{y})},\ \sqrt{L_{\mathscr{E}}(\mathbf{x},\,\mathbf{y}) + 1}\right].$$

This interval is of length at most 1. Thus, the ceiling function in the computation of $L_{est}(t)$ can take at most two values and the lemma follows. □

Note that the set $\{L_r^-,\ L_r^+\}$ depends on r but not on t. Thus, for each $r \in [0, 1]$, the two possible values for $L_{\text{est}}\ (t)$ are the same for all choices of t. We will show that for most r the two values for $L_{\text{est}}\ (t)$ are actually the same for all t.

Let $L_r(t)$ be the loss of $\mathbf{B}(t)$ when predicting the single value $\mathbf{y}_t$ after seeing all ℓ examples except the label $\mathbf{y}_t$ and picking the value r. When r is drawn uniformly at random from [0, 1], the expected total loss of $\mathbf{B}(t)$, summed over choices of t, is

$$E_{r\sim[0,\,1]}(HL_{\mathbf{B}}(\mathbf{x},\,\mathbf{y})) = \sum_{t=1}^{\ell} \int_0^1 L_r(t)dr = \int_0^1 \left(\sum_{t=1}^{\ell} L_r(t) \right) dr. \tag{22}$$

We now consider the expectation over $r \in [0, 1]$ of ${}^{\ell}_{t=1}\, L_r(t)$.

LEMMA 5.4.5. *Choose any set of experts $\mathscr{E}$, and sequences $\mathbf{x}$ and $\mathbf{y}$ of length ℓ, and let L_r^- and L_r^+ be defined as in Lemma 5.4.4.*

Then, for any $r \in [0, 1]$, such that for all $1 \le t \le \ell$, we have $L_{est}(t) = L_r^-$,

$$\sum_{t=1}^{\ell} L_r(t) \leq L_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) + \left(\sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y})} + 1 \right) \sqrt{ln\ N} + \frac{log_2 N}{2} = low.$$

Similarly, for any $r \in [0,1]$ *such that for all* $1 \leq t \leq \ell$ *we have* $L_{est}(t) = L_r^+$,

$$\sum_{t=1}^{\ell} L_r(t) \leq L_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) + \left(\sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) + 1} + 1 \right) \sqrt{ln\ N} + \frac{log_2 N}{2} = high.$$

PROOF. We only prove the first bound. The proof of the second bound is identical. Since $L_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) \leq L_{\text{obs}}(t) + 1 \leq L_{\text{est}}(t) = L_r^-$, we can apply Theorem 4.4.3:

$$\sum_{t=1}^{\ell} L_r(t) \leq L_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) + \sqrt{L_r^- \ln N} + \frac{log_2 N}{2}.$$

Because $\lceil x - r \rceil + r \leq x - r + 1 + r = x + 1$, we have

$$L_r^- \leq \left(\sqrt{L_{\mathscr{E}}\ (\mathbf{x}, \mathbf{y})} + 1 \right)^2.$$

Thus, the RHS of inequality (23) is upper-bounded by "low". □

In the proof of the following most important lemma of this section, we show that most of the time we get a total loss of "low" and only rarely a total loss of at most "low + high". The resulting upper bound is only slightly larger than "low".

LEMMA 5.4.6. *For any set of experts* $\mathscr{E}$ *and sequence* $\mathbf{y}$ *of length* ℓ,

$$E_{r \sim [0,1]}(HL_{\mathbf{B}}(\mathbf{x}, \mathbf{y})) \leq low + \left(\sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) + 1} - \sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y})} \right) high,$$

where low and high are defined as in Lemma 5.4.5.

PROOF. Let us first consider the case when r is such that $L_r^- = L_r^+$: Then each $\mathbf{B}(t)$ chooses $L_{\text{est}}(t) = L_r^-$ and by Lemma 5.4.5:

$$\sum_{t=1}^{\ell} L_r(t) \leq \text{low}. \tag{24}$$

In the remaining case r is such that $L_r^- \neq L_r^+$: Now the $\mathbf{B}(t)$ might use either $L_{\text{est}}(t) = L_r^-$ or $L_{\text{est}}(t) = L_r^+$ for each t. In that case, the sum of the $L_r(t)$ is at most the sum of $L_r(t)$ when all $L_{\text{est}}(t) = L_r^-$ plus the sum of the $L_r(t)$ when all $L_{\text{est}}(t) = L_r^+$:

$$\sum_{t=1}^{\ell} L_r(t) \leq \text{low} + \text{high}. \tag{25}$$

Let $\mathbb{Z} + r = \{k + r : k \in \mathbb{Z}\}$ be the set of integers shifted by $r \in [0, 1]$. We will first show that $L_r^- \neq L_r^+$ iff a point from $\mathbb{Z} + r$ lies in interval $[\sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y})}, \sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) + 1})]$, which is of length at most one. (Note that L_r^- and L_r^+ are the values obtained when applying the mapping $d_r(x) = (\lceil x - r \rceil + r)^2$ to the left and right boundary of the interval.) If a point $k + r$ lies in the interval, then it and the left boundary of the interval map to $(k + r)^2$. Also, any point in the interval that is larger than $k + r$ (including the right boundary of the interval) maps to $(k + 1 + r)^2$. On the other hand, if $L_r^- \neq L_r^+$, then let p be the largest point in the interval that maps to L_r^-. Clearly, p must be in $\mathbb{Z} + r$.

The probability that $L_r^- \neq L_r^+$ equals the probability that the interval $[\sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y})}, \sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) + 1}]$ contains a point of $\mathbb{Z} + r$. Since r is drawn uniformly in [0, 1] and since the interval has length at most one, this probability equals the length of the interval, that is $\sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) + 1} - \sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y})}$. This allows us to average inequalities (24) and (25) to get

$$\begin{aligned}
\sum_{t=1}^{\ell} L_{\mathbf{B}(t)} &= \int_0^1 \left(\sum_{t=1}^{\ell} L_r(t) \right) dr \\
&\leq \left(1 - \left(\sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) + 1} - \sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y})} \right) \right) \text{low} \\
&\quad + \left(\sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) + 1} - \sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y})} \right) (\text{low} + \text{high}) \\
&= \text{low} + \left(\sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) + 1} - \sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y})} \right) \text{high}. \qquad \square
\end{aligned}$$

PROOF OF THEOREM 5.4.3. For the first part of the theorem, which is a bound on $E_{r \sim [0,\,1]}(HL_{\mathbf{B}}(\mathbf{x}, \mathbf{y}))$, what remains to be done is to simplify the upper bound of Lemma 5.4.6. First observe that

$$\begin{aligned}
\left(\sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) + 1} - \sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y})} \right) \text{high} &\leq \frac{1}{\sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) + 1}} \text{high} \\
&\leq \sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y})} + \left(1 + \frac{1}{\sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) + 1}} \right) \sqrt{\ln N} \\
&\quad + \left(\frac{1}{\sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) + 1}} \right) \frac{\log_2 N}{2}.
\end{aligned}$$

Plugging this into the bound of the lemma, we get

$$\begin{aligned}
E_{r \sim [0,1]}(HL_{\mathbf{B}}(\mathbf{x}, \mathbf{y})) \leq L_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) &+ \sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y})} \left(\sqrt{\ln N} + 1 \right) \\
&+ \left(2 + \frac{1}{\sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) + 1}} \right) \sqrt{\ln N} \\
&+ \left(\frac{1}{\sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) + 1}} + 1 \right) \frac{\log_2 N}{2}
\end{aligned}$$

$$\leq L_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) + \sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y})}(\sqrt{\ln N} + 1) + 3\sqrt{\ln N} + \frac{\log_2 N}{2}.$$

For the second part, view algorithm **DB** as a version of algorithm **B** where r is chosen uniformly from the finite set $\{i/q : 0 \leq i \leq q - 1\}$ instead of uniformly from the continuous interval [0, 1]. (Recall that $q = \ell + (\sqrt{\ell + 1} + 1)\sqrt{\ln N} + \log_2 N/2$ and this choice of q is at least as large as the value *high*.) In Lemma 5.4.6, we showed that the expected hold-one-out loss is at most low + p high, where p is the probability of the event that the set $\{k + r : k \in \mathbb{Z}\}$ has a point in the interval $[\sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y})}, \sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) + 1}]$. If $r \in [0, 1]$, then p equals the length of the interval and in the case $r \in \{i/q : 0 \leq i \leq q - 1\}$ the probability p equals the length plus or minus $1/q$. Since $q \geq$ high, we get the following upper bound on the total hold-one-out loss of algorithm **DB**:

$$HL_{\mathbf{B}}(\mathbf{x}, \mathbf{y}) \leq \text{low} + \left(\sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) + 1} - \sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y})} + \frac{1}{q}\right) \text{high}$$

$$\leq \text{low} + \left(\sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) + 1} - \sqrt{L_{\mathscr{E}}(\mathbf{x}, \mathbf{y})} + \frac{1}{\text{high}}\right) \text{high}.$$

Thus, the bound in the second part is at most one larger than the bound proven in the first part. □

We are finally now in a position to return to the pattern recognition problem considered in Section 5. The next lemma generalizes the argument given in the proof of Theorem 5.2.1 to give a general method for converting hold-one-out prediction strategies to learning algorithms that solve the pattern recognition problem.

LEMMA 5.4.7. *Let A be a hold-one-out prediction strategy. Then A can be converted into a learning strategy B such that for any comparison class $\mathscr{H}$, any ℓ, and any distribution D on $X \times \{0, 1\}$,*

$$E_{\mathbf{s}\sim D^{\ell}}(\text{er}_D(B(\mathbf{s}))) - \text{er}_D(\mathscr{H}) = \frac{1}{\ell + 1} E_{(\mathbf{x},\mathbf{y})\sim D^{\ell+1}}(HL_A(\mathbf{x}, \mathbf{y}) - L_{\mathscr{H}_{|\mathbf{x}}}(\mathbf{x}, \mathbf{y}))$$

$$- \text{er}^{\Delta}_{\ell+1,D}(\mathscr{H}),$$

where $E_{(\mathbf{x},\mathbf{y})\sim D^{\ell+1}}$ 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, $1 \leq t \leq \ell + 1$.

PROOF OF LEMMA 5.4.7. The 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, let $\hat{y}_t$ denote 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 $\mathscr{H}_{|\mathbf{x}}$ 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 tth outcome to be predicted. Now the value of the function $h = B(\mathbf{s})$ on input x is defined by the average of the $\hat{y}_t$'s, that is, $h(x) = (\ell + 1)^{-1} \; {}_{t=1}^{\ell+1} \hat{y}_t$.

To show that this strategy B has the desired performance, first note the following

$$E_{\mathbf{s}\sim D^\ell}(\mathrm{er}_D(B(\mathbf{s}))) = E_{\mathbf{s}\sim D^\ell,(x,y)\sim D}|B(\mathbf{s})(x) - y|$$

$$= E_{\mathbf{s}\sim D^\ell,(x,y)\sim D}\left|\left(\frac{1}{\ell+1}\sum_{t=1}^{\ell+1}\hat{y}_t\right) - y\right|, \tag{26}$$

where $\hat{y}_t$ is as defined in the previous paragraph, and $\mathbf{s} = (x_1, y_1), \ldots, (x_\ell, y_\ell)$.

Because $|(\frac{1}{n}\ {}^{n}_{t=1}\, p_t) - c| = \frac{1}{n}\ {}^{n}_{t=1}\, |p_t - c|$ for $c \in \{0, 1\}$ and $0 \le p_t \le 1$, it follows that

$$E_{\mathbf{s}\sim D^\ell}(\mathrm{er}_D(B(\mathbf{s}))) = E_{\mathbf{s}\sim D^\ell,(x,y)\sim D}\frac{1}{\ell+1}\sum_{t=1}^{\ell+1}|\hat{y}_t - y|$$

$$= \frac{1}{\ell+1}\sum_{t=1}^{\ell+1}E_{\mathbf{s}\sim D^\ell,(x,y)\sim D}|\hat{y}_t - y|$$

$$= \frac{1}{\ell+1}\sum_{t=1}^{\ell+1}E_{(\mathbf{x},\mathbf{y})\sim D^{\ell+1}}|\hat{y}'_t - y_t|, \tag{27}$$

where $\hat{y}'_t$ is the output of A when A is given as input the sequence of instances $\mathbf{x} = x_1, \ldots, x_{\ell+1}$, the set $\mathcal{H}_{|\mathbf{x}}$ of experts, and the observed outcomes $\mathbf{y} = y_1, \ldots, y_{t-1}, ?, y_{t+1}, \ldots, y_{\ell+1}$, where "?" denotes the location of the missing outcome to be predicted. Thus, by the definition of the hold-one-out prediction loss

$$E_{\mathbf{s}\sim D^\ell}(\mathrm{er}_D(B(\mathbf{s}))) = \frac{1}{\ell+1}E_{(\mathbf{x},\mathbf{y})\sim D^{\ell+1}}\sum_{t=1}^{\ell+1}|\hat{y}'_t - y_t|$$

$$= \frac{1}{\ell+1}E_{(\mathbf{x},\mathbf{y})\sim D^{\ell+1}}HL_A(\mathbf{x}, \mathbf{y}), \tag{28}$$

where $HL_A(\mathbf{x}, \mathbf{y})$ denotes the total hold-one-out prediction loss of the strategy A on instances $\mathbf{x}$ and outcomes $\mathbf{y}$, assuming the set of experts used is $\mathcal{H}_{|\mathbf{x}}$.

Furthermore, it is clear that

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

$$= E_{(\mathbf{x},\mathbf{y})\sim D^\ell}\frac{1}{\ell}L_{\mathcal{H}_{|\mathbf{x}}}(\mathbf{x}, \mathbf{y}). \tag{29}$$

It follows from Eqs. (28) and (29) and the definition of expected over-fit that

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

$$= E_{\mathbf{s}\sim D^\ell}(\mathrm{er}_D(B(\mathbf{s}))) - \widehat{\mathrm{er}}_{\ell+1,D}(\mathcal{H}) - (\mathrm{er}_D(\mathcal{H}) - \widehat{\mathrm{er}}_{\ell+1,D}(\mathcal{H}))$$

$$= \frac{1}{\ell+1}E_{(\mathbf{x},\mathbf{y})\sim D^{\ell+1}}HL_A(\mathbf{x}, \mathbf{y}) - \frac{1}{\ell+1}E_{(\mathbf{x},\mathbf{y})\sim D^{\ell+1}}L_{\mathcal{H}_{|\mathbf{x}}}(\mathbf{x}, \mathbf{y}) - \mathrm{er}^{\Delta}_{\ell+1,\mathrm{D}}\,(\mathcal{H})$$

$$= \frac{1}{\ell + 1} E_{(\mathbf{x},\mathbf{y})\sim D^{\ell+1}}(HL_A(\mathbf{x}, \mathbf{y}) - L_{\mathcal{H}_{|\mathbf{x}}}(\mathbf{x}, \mathbf{y})) - \mathrm{er}^{\Delta}_{\ell+1,\mathrm{D}}(\mathcal{H}). \qquad \square$$

Finally, we can now complete the

PROOF OF THEOREM 5.3.1. From Theorem 5.4.3 and the above lemma, with A being the algorithm **DB**, it follows that

$$E_{\mathbf{s}\sim D^{\ell}}(\mathrm{er}_D(A(\mathbf{s}))) - \mathrm{er}_D(\mathcal{H}) \le \frac{E_{(\mathbf{x},\mathbf{y})\sim D^{\ell+1}}\left[\sqrt{L_{\mathcal{H}_{|\mathbf{x}}}(\mathbf{x}, \mathbf{y})}\ (\sqrt{\ln|\mathcal{H}_{|\mathbf{x}}|} + 1)\right]}{\ell + 1}$$
$$+ \frac{E_{\mathbf{x}} \ln|\mathcal{H}_{|\mathbf{x}}|}{(\ell + 1)\ln 2} + \frac{3E_{\mathbf{x}}\sqrt{\ln|\mathcal{H}_{|\mathbf{x}}|} + 1}{\ell + 1} - \mathrm{er}^{\Delta}_{\ell+1,D}(\mathcal{H}). \tag{30}$$

Hence by the Cauchy–Schwarz inequality (applied in the first line below) and by Jensen's inequality (applied in the second line),

$$E_{\mathbf{s}\sim D^{\ell}}(\mathrm{er}_D(A(\mathbf{s}))) - \mathrm{er}_D(\mathcal{H}) \le \frac{\sqrt{E_{(\mathbf{x},\mathbf{y})\sim D^{\ell+1}} L_{\mathcal{H}_{|\mathbf{x}}}(\mathbf{x}, \mathbf{y})}\ (\sqrt{E_{\mathbf{x}} \ln|\mathcal{H}_{|\mathbf{x}}|} + 1)}{\ell + 1}$$
$$+ \frac{E_{\mathbf{x}} \ln|\mathcal{H}_{|\mathbf{x}}|}{(\ell + 1)\ln 2} + \frac{3\sqrt{E_{\mathbf{x}} \ln|\mathcal{H}_{|\mathbf{x}}|} + 1}{\ell + 1} - \mathrm{er}^{\Delta}_{\ell+1,D}(\mathcal{H}).$$

Since $T = E_{\mathbf{x}} \ln |\mathcal{H}_{|\mathbf{x}}|$ and since Eq. (29) implies that

$$E_{(\mathbf{x},\mathbf{y})\sim D^{\ell+1}}(L_{\mathcal{H}_{|\mathbf{x}}}(\mathbf{x}, \mathbf{y})) = (\ell + 1)\widehat{\mathrm{er}}_{\ell+1,D}(\mathcal{H}),$$

Eq. (20) follows. From this, Eq. (21) follows by simply noting that $\widehat{\mathrm{er}}_{\ell+1,D}(\mathcal{H}) \le \mathrm{er}_D(\mathcal{H})$. □

Note that, for sake of simplicity, the bounds stated the Theorem 5.3.1 are actually weaker than what we prove in Eq. (30).

6. *Worst-Case Loss Bounds for the Log Loss*

It is interesting to relate the min/max analysis, given in Section 3, to results on the problem of optimal universal sequential coding studied by Shtarkov [1975; 1987].

The problem of sequential coding is similar to the problem studied in this paper, with two major differences:

(1) The loss function that is studied in this paper is $|p - y|$. This loss corresponds to the probability of making a mistake if making a prediction by flipping a random coin whose bias is p. The study of sequential coding, on the other hand, is interested in the *log loss* function

$$-y \ln p - (1 - y) \ln (1 - p).$$

This loss function is closely related to the minimal average coding length that can be achieved by using the given predictions (see Rissanen and Langdon [1981]).

(2) The predictions made by the "experts" as defined here are not restricted; they can depend on any information that is available to the experts. The corresponding concept in Shtarkov's paper is that of a "source." A source is an expert whose prediction at time t depends only on the previous outcomes: $y_1, \ldots, y_{t-1}$. We call such experts "simulatable" because their future predictions can be simulated by feeding them with future outcomes. The predictions of a simulatable expert can be viewed as a conditional distribution $p(y_t|y_1, \ldots, y_{t-1})$. This means that any simulatable expert can be identified with a distribution over the set of infinite binary sequences. Assuming all our experts are simulatable, we denote by P_i the distribution associated with expert i. Similarly, if we fix a prediction algorithm A that combines a fixed set of experts we can associate with it a distribution P_A.

It is well known that for the log loss, for any set $\mathscr{E}$ of N experts there is a prediction strategy A such that for any sequence $\mathbf{y}$, $L_A(\mathbf{y}) - L_{\mathscr{E}}(\mathbf{y}) \le \log N$, where $L_{\mathscr{E}}(\mathbf{y})$ is the total log loss of the best expert for $\mathbf{y}$.[14,15] The strategy is just the Bayes algorithm with uniform prior on the distributions represented by the experts.

A min/max optimal prediction algorithm is known for the case in which the experts are simulatable and ℓ, the number of iterations, is known in advance. This result is given by Shtarkov [1987, Theorem 1]. For completeness, we restate the theorem and its proof here using our terminology.

THEOREM 6.1 (SHTARKOV). *For each $\mathbf{y} \in \{0, 1\}^\ell$ and each expert $\mathscr{E}_i \in \mathscr{E}$, let $P_i(\mathbf{y})$ denote the probability of $\mathbf{y}$ under expert $\mathscr{E}_i$. Define the probability of $\mathbf{y}$ for the algorithm A by*

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

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

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

PROOF. Since $L_A(\mathbf{y}) = -\log P_A(\mathbf{y})$ and $L_{\mathscr{E}}(\mathbf{y}) = -\log \max_{1\le i\le N} P_i(\mathbf{y})$, it follows from the definition of P_A that

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

for all $\mathbf{y}$. Clearly this value is at most $\log N$. Furthermore, A can be interpreted as a Bayes algorithm for predicting the bits of $\mathbf{y}$ under the log loss, where the prior probability of $\mathbf{y}$ is given by

[14] This inequality holds even if the experts are not simulatable.

[15] See, for example, Rissanen [1986], Desantis et al. [1988], Vovk [1992], Haussler and Barron [1992], Yamanishi [1995], and Kivinen and Warmuth [1994].

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

Since A is Bayes and has the same regret $L_A(\mathbf{y}) - L_{\mathscr{E}}(\mathbf{y})$ for each $\mathbf{y}$, it follows that A is min/max. Otherwise, there would exist another algorithm A' with average regret with respect to this prior that is less than the Bayes optimal algorithm, which would yield a contradiction. □

It is instructive to contrast the simplicity of the algorithms and analysis for log loss to the relative complexity involved in the analysis of the algorithms in this paper, which aim to minimize the absolute loss. This suggests that, when given the choice, one might be better off choosing to use the log loss. However, in many situations there is no such choice because the goal is to minimize the number of mistakes and not to minimize the length of a coding of the sequence.

7. *Conclusions*

In this paper, we prove worst-case loss bounds for on-line learning for the absolute loss, and give applications in pattern recognition. We bound the additional loss of the algorithm over the loss of the best expert. Apart from the game-theoretic analysis, our main upper bound is obtained essentially by tuning an algorithm that was first introduced by Vovk (Theorem 4.4.3). Other loss functions for the expert framework are considered in Vovk [1990] and Haussler et al. [1995].

The paper leaves many open problems. Our lower bounds only address the case when a bound on the length of the sequence of examples is known. We would like to have lower bounds for the case when the sequence is of unbounded length but the loss of the best expert lies below a bound that is known to the algorithm. In other words, are there lower bounds that match the upper bounds of Theorem 4.4.3?

For the case when the algorithm has no prior knowledge of the loss of the best expert (Theorem 4.6.3), can the constant in front of the square root be lowered and the algorithm be simplified? We would also like to generalize our upper bounds of Theorem 4.4.3 to the case when the set of experts is infinite. Assume the expert $\mathscr{E}_i$ has initial weight w_i and the total weight ${}_{i=1}^{\infty} w_i$ of all experts is one. We would like to get bounds of the following form that hold for arbitrary outcome sequence $\mathbf{y}$:

$$L_A(\mathbf{y}) \le \inf_{1 \le i \le \infty} \left(L_{\mathscr{E}_i}(\mathbf{y}) + c \sqrt{L_{\mathscr{E}_i} \ln\left(\frac{1}{w_i}\right)} + c' \ln\left(\frac{1}{w_i}\right), \right),$$

where the constants c and c' are as low as possible. Weaker bounds that are not in the above form have been given by Littlestone and Warmuth [1994].

Our new bounds proven for the PAC model (Section 5) are better that previous bounds but the algorithms are very complicated. Is the hold-one-out model necessary to prove the bounds given for the PAC model? Can the same bounds be obtained by simpler algorithms?

The upper bound for the main algorithm **P** of this paper (Theorem 4.2.1) has recently been generalized by Kivinen and Warmuth [1994] to the case when the outcomes lie in the interval [0, 1] instead being restricted to be binary as done in this paper. The new result can be used as a starting point for generalizing the results for the PAC model to the case when the hypotheses have range [0, 1] instead of {0, 1}.

Appendix A. Proof of Lemma 3.2.1.

The proof is based on the fact that the distribution of $A_{\ell,N}$, after proper rescaling and shifting, converges to a limit distribution. However, as convergence of the distributions does not imply convergence of the expected values, we need to use a slightly more involved argument.

Let $Y_{\ell,i}$ be a normalized version of $S_{\ell,i}$, with mean 0 and variance 1

$$Y_{\ell,i} = \frac{S_{\ell,i} - \ell/2}{\sqrt{\ell/2}}, \tag{31}$$

and let $B_{\ell,N}$ be

$$B_{\ell,N} = \frac{\min_{1 \le i \le N}\{Y_{\ell,i}\}}{\sqrt{2 \ln N}} = \frac{A_{\ell,N} - \ell/2}{\sqrt{(\ell/2)\ln N}}. \tag{32}$$

It suffices to show that $\quad \epsilon > 0 \quad N_0 \quad N > N_0 \quad \ell_0 \quad \ell \ge \ell_0$

$$E(B_{\ell,N}) \le -1 + \epsilon. \tag{33}$$

In order to prove this claim, we upper bound the expectation by a sum as follows:

$$E(B_{\ell,N}) \le \left(-1 + \frac{\epsilon}{3} \right) P\left(B_{\ell,N} \le -1 + \frac{\epsilon}{3} \right) + 0P(B_{\ell,N} \le 0) + \int_0^\infty P(B_{\ell,N} \ge C)dc. \tag{34}$$

We start by bounding the third term in (34). In general, we have that

$$P(B_{\ell,N} \ge c) = \prod_{i=1}^{N} P\left(\frac{S_{\ell,i} - \ell/2}{\sqrt{\ell(\ln N)/2}} \ge c \right), \tag{35}$$

and as the expected value of $S_{\ell,i}$ is $\ell/2$, we can bound the RHS using Hoeffding's bound:

$$\begin{aligned} P\left(\frac{S_{\ell,i} - \ell/2}{\sqrt{\ell(\ln N)/2}} \ge c \right) &= P\left(S_{\ell,i} \ge \frac{\ell}{2} + \ell \left(\frac{c\sqrt{(\ln N)/2}}{\sqrt{\ell}} \right) \right) \\ &\le \exp\left(-2\ell \left(\frac{c\sqrt{(\ln N)/2}}{\sqrt{\ell}} \right)^2 \right) \\ &= \exp(-c^2 \ln N). \end{aligned} \tag{36}$$

Plugging this back into the integral, we get

$$\int_0^\infty P(B_{\ell,N} \geq c)dc \leq \int_0^\infty \exp(-c^2 N \ln N)dc = \frac{1}{2}\sqrt{\frac{\pi}{N \ln N}} \leq \frac{\epsilon}{3} \tag{37}$$

for sufficiently large N.

It remains to bound the first term in Eq. (34). Let c be an arbitrary real number. From the central limit theorem, it follows that

$$P(Y_{\ell,i} \geq c) \overset{\ell\to\infty}{\to} P(\Phi_i \geq c), \tag{38}$$

where Φ_i are independent random variables from the normal distribution $\mathcal{N}(0, 1)$. From this, we get that

$$\begin{aligned} P(\sqrt{2 \ln N}\, B_{\ell,N} \leq c) &= P\left(\min_{1\leq i\leq N} Y_{\ell,i} \leq c\right) \\ &= 1 - \prod_{i=1}^{N} P(Y_{\ell,i} > c) \overset{\ell\to\infty}{\to} 1 - \prod_{i=1}^{N} P(\Phi_i > c) \\ &= P(\Theta_N \leq c), \end{aligned} \tag{39}$$

where $\Theta_N = \min_{1\leq i\leq N} \{\Phi_i\}$. On the other hand, asymptotic analysis of the extreme order statistics of the normal distribution (see Galambos [1987], Section 2.3.2, Eqs. (59), (60)) shows that

$$P\left(\frac{\Theta_N - a_N}{b_N} \leq c\right) \overset{N\to\infty}{\to} 1 - \exp(-e^c), \tag{40}$$

where

$$a_N = -\sqrt{2 \ln N} + \frac{\ln \ln N + \ln 4\pi}{2\sqrt{2 \ln N}} \qquad \text{and} \qquad b_N = \frac{1}{\sqrt{2 \ln N}}. \tag{41}$$

Combining Eqs. (39) and (40), we get that

$$\lim_{N\to\infty} \lim_{\ell\to\infty} P\left(B_{\ell,N} > \frac{cb_N + a_N}{\sqrt{2 \ln N}}\right) = \exp(-e^c). \tag{42}$$

We now fix c sufficiently large so that $\exp(-e^c) < \epsilon/3$. For N and ℓ sufficiently large we have that

$$P\left(B_{\ell,N} > \frac{cb_N + a_N}{\sqrt{2 \ln N}}\right) < \frac{\epsilon}{3}. \tag{43}$$

Plugging in the definitions of a_N and b_N, we get that

$$P\left(B_{\ell,N} > \frac{c}{2 \ln N} - 1 + \frac{1/2(\ln \ln N + \ln 4\pi)}{2 \ln N}\right) < \frac{\epsilon}{3}. \tag{44}$$

Choosing N large enough we finally get that

$$P\left(B_{\ell,N} > -1 + \frac{\epsilon}{3}\right) < \frac{\epsilon}{3}, \tag{45}$$

which upper bounds the first term in Eq. (34) by $(1 - \epsilon/3)(-1 + \epsilon/3) < -1 + (2/3)\epsilon$. This, combined with the above bound for the third term, completes the proof. □

Appendix B Proof of Lemma 4.4.1

Recall that $z > 0$ or $z = \infty$ and thus

$$g(z) = \frac{1}{1 + 2z + z^2/\ln 2} \in [0, 1).$$

The following inequalities are equivalent to the lemma.

$$\frac{z^2 - \ln(g(z))}{2\ln(2/(1 + g(z)))} \le \frac{1}{2g(z)} + \frac{1}{2}$$

$$0 \le \left(1 + \frac{1}{g(z)}\right) \ln\left(\frac{2}{1 + g(z)}\right) - z^2 + \ln(g(z)).$$

Since $g(0) = 1$, the last inequality holds for $z = 0$. Thus, it suffices to show that the derivative of the RHS is nonnegative for all $z \ge 0$. Taking this derivative we get

$$-\frac{g'(z)\ln(2/(1 + g(z)))}{g(z)^2} - \frac{(1 + (1/g(z)))g'(z)}{1 + g(z)} - 2z + \frac{g'(z)}{g(z)}$$

which simplifies to

$$-\frac{g'(z)\ln(2/(1 + g(z)))}{g(z)^2} - 2z.$$

Note that $g'(z) = -(2 + 2z/\ln 2)g(z)^2$, so the derivative is nonnegative whenever

$$\left(2 + \frac{2z}{\ln 2}\right)\ln\left(\frac{2}{1 + g(z)}\right) - 2z \ge 0. \tag{46}$$

We now consider two cases depending on the value of z. In the first case,

$$0 < z \le \frac{3\ln 2 - 4\ln^2 2}{2\ln 2 - 1} \approx .4$$

and we use the approximation $\ln(1 + x) \ge x/(1 + x)$. With this approximation,

$$\ln\left(\frac{2}{1+g(z)}\right) = \ln\left(1 + \frac{1-g(z)}{1+g(z)}\right) \geq \frac{1-g(z)}{2}.$$

Plugging back into Inequality (46), we see that the derivative is nonnegative whenever

$$\left(2 + \frac{2z}{\ln 2}\right) \frac{1-g(z)}{2} - 2z \geq 0.$$

By multiplying the above with $1/g(z)$, we get the following equivalent inequalities:

$$\left(1 + \frac{z}{\ln 2}\right)\left(2z + \frac{z^2}{\ln 2}\right) - 2z\left(1 + 2z + \frac{z^2}{\ln 2}\right) \geq 0$$

$$\frac{3z^2}{\ln 2} + \frac{z^3}{\ln^2 2} - 4z^2 - 2\frac{z^3}{\ln 2} \geq 0$$

$$3 \ln 2 - 4 \ln^2 2 + z - 2z \ln 2 \geq 0,$$

which holds due to the assumption that $z \leq (3 \ln 2 - 4 \ln^2 2)/(2 \ln 2 - 1)$. Now we assume that $z \geq (3 \ln 2 - 4 \ln^2 2)/(2 \ln 2 - 1)$. Note that $(1 - g(z))/(1 + g(z))$ is an increasing function which approaches 1 as $z \to \infty$. Furthermore, under the assumptions of this case, $g(z) \leq (2 \ln 2 - 1)^2/(1 - \ln 2) < 1/2$ and $(1 - g(z))/(1 + g(z)) > 1/3$. Thus, we can underestimate $\ln(1 + x)$ by interpolating between $x = 1/3$ and $x = 1$ (with $(1 - g(z))/(1 + g(z)) = x$)

$$\ln\left(1 + \frac{1-g(z)}{1+g(z)}\right) \geq \frac{3}{2}\left(1 - \frac{1-g(z)}{1+g(z)}\right) \ln\left(\frac{4}{3}\right) + \frac{3}{2}\left(\frac{1-g(z)}{1+g(z)} - \frac{1}{3}\right) \ln 2.$$

Thus, for the values of z considered in this case, the following equivalent form of (46)

$$\ln\left(1 + \frac{1-g(z)}{1+g(z)}\right) - \frac{z}{1 + (z/\ln 2)} \geq 0$$

holds whenever

$$\frac{3}{2}\left(1 - \frac{1-g(z)}{1+g(z)}\right)\ln\left(\frac{4}{3}\right) + \frac{3}{2}\left(\frac{1-g(z)}{1+g(z)} - \frac{1}{3}\right)\ln 2 - \frac{z}{1 + (z/\ln 2)} \geq 0$$

$$\frac{3g(z)}{1+g(z)} \ln\left(\frac{4}{3}\right) + \frac{1-2g(z)}{1+g(z)}\ln 2 - \frac{z}{1 + (z/\ln 2)} \geq 0$$

$$\left(3g(z)\ln\left(\frac{4}{3}\right) + (1 - 2g(z))\ln 2\right)\left(1 + \frac{z}{\ln 2}\right) - z(1 + g(z)) \geq 0$$

$$\left(3 \ln\left(\frac{4}{3}\right) - \ln 2 + 2z \ln 2 + z^2\right)\left(1 + \frac{z}{\ln 2}\right) - z\left(2 + 2z + \frac{z^2}{\ln 2}\right) \geq 0$$

$$3\ \ln\left(\frac{4}{3}\right) - \ln 2 + 2z\ \ln 2 + z^2 + \frac{3z\ \ln(4/3)}{\ln 2} - z - 2z \geq 0$$

$$3\ \ln\left(\frac{4}{3}\right) - \ln 2 + z\left(2\ \ln 2 + \frac{3\ \ln(4/3)}{\ln 2} - 3\right) + z^2 \geq 0$$

Finally, we observe that this polynomial is always positive, obtaining its minimum of about 0.13 when $z \approx 0.18$. □

Appendix C. Proof of Theorem 4.6.3

First, if $L_{\mathscr{E}}(\mathbf{y}) \leq a^2 \ln N$, then the algorithms first guess k_0 is an upper bound on the loss of the best expert, and by Theorem 4.4.3 the loss of **P*** is bounded by at most

$$L_{\mathscr{E}}(\mathbf{y}) + \sqrt{a^2(\ln N)^2} + \frac{1}{2}\log_2 N = L_{\mathscr{E}}(\mathbf{y}) + \left(a + \frac{1}{2\ \ln 2}\right) \ln N,$$

satisfying the theorem. We proceed with the assumption that $L_{\mathscr{E}}(\mathbf{y}) > a^2 \ln N$.

Let *last* be the largest iteration number in which a prediction was made by algorithm **P***. Let $L_{last,\mathscr{E}_i}$ be the loss incurred by the expert $\mathscr{E}_i$ while algorithm **P*** is executing iteration number *last*, and let $L_{last,\mathscr{E}}$ be the minimum $L_{last,\mathscr{E}_i}$ over $\mathscr{E}_i \in \mathscr{E}$. If $L_{last,\mathscr{E}} \leq k_{last}$, then by Theorem 4.4.3 the loss of algorithm **P*** during iteration number last is at most

$$L_{last,\mathscr{E}} + \sqrt{k_{last} \ln N} + \frac{1}{2}\log_2 N = L_{last,\mathscr{E}} + \left(ac^{last/2} + \frac{1}{2\ \ln 2}\right) \ln N.$$

If $L_{last,\mathscr{E}} > k_{last}$, then, as there are no more iterations after *last* (implying that **P*** makes only one additional prediction following the last prediction in which the loss of algorithm **P*** is at most b_{last}), the loss of algorithm **P*** during iteration number *last* is at most

$$b_{last} + 1 \leq L_{last,\mathscr{E}} + \left(ac^{last/2} + \frac{1}{2\ \ln 2}\right)\ln N + 1.$$

Using the above and the fact that the loss incurred by **P*** during any iteration z is at most $b_z + 1$, we can bound $L_{P^*}(\mathbf{y})$,

$$L_{P^*}(\mathbf{y}) \leq L_{last,\mathscr{E}} + \left(ac^{last/2} + \frac{1}{2\ \ln 2}\right)\ln N + 1 + \sum_{z=0}^{last-1} (b_z + 1).$$

Using Eq. (14),

$$L_{P^*}(\mathbf{y}) \leq L_{last,\mathscr{E}} + \left(ac^{last/2} + \frac{1}{2\ \ln 2}\right)\ln N + 1$$

$$+ \sum_{z=0}^{last-1} \left(k_z + \left(ac^{z/2} + \frac{1}{2 \ln 2} \right) \ln N + 1 \right)$$

$$\leq L_{last,\mathscr{E}} + \sum_{z=0}^{last-1} k_z + \sum_{z=0}^{last} \left(\left(ac^{z/2} + \frac{1}{2 \ln 2} \right) \ln N + 1 \right).$$

Lemma 4.6.1 implies that $L_{\mathscr{E}}(\mathbf{y})$, the loss of the best expert, is at least $L_{last,\mathscr{E}} + {}_{z=0}^{last-1} k_z$. Using this fact,

$$L_{P^*}(\mathbf{y}) \leq L_{\mathscr{E}}(\mathbf{y}) + (last + 1)\left(1 + \frac{\ln N}{2 \ln 2} \right) + \sum_{z=0}^{last} ac^{z/2} \ln N. \tag{47}$$

We now work on the second and third terms separately. We will use the following lemma to help simplify the second term.

Lemma C.1. *For all $x \geq 0$, $ln(1 + x) \leq 0.805 \sqrt{x}$.*

Proof (of Lemma). It is slightly easier to show that for all $z \geq 0$, $\ln(1 + z^2) - 0.805z \leq 0$. The inequality clearly holds at $z = 0$ and $z = \infty$. By differentiating, we see that the extrema are at

$$z = \frac{1 \pm \sqrt{1 - (0.805)^2}}{0.805}.$$

Plugging these values in show that both of these (local) extrema are negative, so $\ln(1 + z^2) - 0.805z \leq 0$ for all $z \geq 0$. □

We return to the proof of the theorem by applying Lemma 4.6.2 followed by Lemma C.1 to the second term.

$$\begin{aligned}
(last + 1)\left(1 + \frac{\ln N}{2 \ln 2}\right) &\leq 1 + \frac{\log_2 N}{2} + \left(1 + \frac{\ln N}{2 \ln 2}\right)\log_c\left(1 + \frac{L_{\mathscr{E}}(\mathbf{y})(c-1)}{a^2 \ln N}\right) \\
&\leq 1 + \frac{\log_2 N}{2} + \left(1 + \frac{\ln N}{2 \ln 2}\right)\frac{0.805}{\ln c}\sqrt{\frac{L_{\mathscr{E}}(\mathbf{y})(c-1)}{a^2 \ln N}} \\
&= 1 + \frac{\log_2 N}{2} + \left(1 + \frac{\ln N}{2 \ln 2}\right)\frac{0.805\sqrt{(c-1)}}{a \ln N \ln c}\sqrt{L_{\mathscr{E}}(\mathbf{y})\ln N} \\
&= 1 + \frac{\log_2 N}{2} + \left(\frac{0.805\sqrt{(c-1)}}{a \ln N \ln c} + \frac{0.805\sqrt{(c-1)}}{a(2 \ln 2)\ln c}\right)\sqrt{L_{\mathscr{E}}(\mathbf{y})\ln N.}
\end{aligned}$$

For the third term of Eq. (47) we sum the geometric series and then apply Lemma 4.6.2.

$$\sum_{z=0}^{last} ac^{z/2} \ln N = a \ln N \frac{\sqrt{c}^{last+1} - 1}{\sqrt{c} - 1}$$

$$\leq a \ \ln N \, \frac{\sqrt{c\left(1 + \dfrac{L_{\mathscr{E}}(\mathbf{y})(c-1)}{a^2 \ln N}\right)}}{\sqrt{c} - 1} - \frac{a \ \ln N}{\sqrt{c} - 1}.$$

We continue with the approximation $\sqrt{1 + x} \leq \sqrt{x} + 1/\sqrt{4x}$ and then use the assumption that $L_{\mathscr{E}}(\mathbf{y}) \geq a^2 \ln N$.

$$\begin{aligned}
\sum_{z=0}^{last} a c^{z/2} \ln N &\leq \frac{a\sqrt{c}\ \ln N}{\sqrt{c} - 1}\left(\sqrt{\frac{L_{\mathscr{E}}(\mathbf{y})(c-1)}{a^2 \ln N}} + \sqrt{\frac{a^2 \ln N}{4L_{\mathscr{E}}(\mathbf{y})(c-1)}}\right) - \frac{a \ \ln N}{\sqrt{c} - 1} \\
&\leq \frac{\sqrt{c(c-1)}}{\sqrt{c} - 1}\sqrt{L_{\mathscr{E}}(\mathbf{y}) \ln N} + \frac{a\sqrt{c}\ \ln N}{2(\sqrt{c} - 1)\sqrt{c - 1}} - \frac{a \ \ln N}{\sqrt{c} - 1} \\
&= \frac{\sqrt{c(c-1)}}{\sqrt{c} - 1}\sqrt{L_{\mathscr{E}}(\mathbf{y}) \ln N} - a \ \ln N \, \frac{2\sqrt{c - 1} - \sqrt{c}}{2(\sqrt{c} - 1)\sqrt{c - 1}}.
\end{aligned}$$

Plugging these results back into (47) yields

$$\begin{aligned}
L_{P^*}(\mathbf{y}) \leq L_{\mathscr{E}}(\mathbf{y}) + 1 + \frac{\ln N}{2 \ln 2} - \frac{a \ \ln N(2\sqrt{c - 1} - \sqrt{c})}{2(\sqrt{c} - 1)\sqrt{c - 1}} \\
+ \left(\frac{0.805\sqrt{c - 1}}{a \ \ln N \ \ln c} + \frac{0.805\sqrt{c - 1}}{a(2 \ \ln 2)\ln c} + \frac{\sqrt{c(c-1)}}{\sqrt{c} - 1}\right)\sqrt{L_{\mathscr{E}}(\mathbf{y}) \ln N}.
\end{aligned}$$

We use ϕ to denote the golden ratio, $(1 + \sqrt{5})/2$, and recall that $\phi^2 - 1 = \phi$. The $\sqrt{c(c-1)}/(\sqrt{c} - 1)$ term is minimized at $c = \phi^2$, where it is $\phi^{3/2}/(\phi - 1)$, or about 3.33 (less than 3.3302).

With c set to ϕ^2, the factor in front of the $\sqrt{L_{\mathscr{E}}(\mathbf{y}) \ln N}$ term is less than

$$\left(\frac{\phi^{3/2}}{(\phi - 1)} + \frac{0.805\sqrt{\phi}}{4a \ \ln 2 \ \ln \phi} + \frac{0.805\sqrt{\phi}}{2a \ \ln N \ \ln \phi}\right).$$

We now turn our attention to the coefficient of the $\ln N$ term together with the "+1". For $c = \phi^2$, this factor is

$$\frac{1}{\ln N} + \frac{1}{2 \ln 2} - \frac{a(2 - \sqrt{\phi})}{2(\phi - 1)}$$

and is less than $1/(2 \ln 2)$ for all $a \geq 2(\phi - 1)/((2 - \sqrt{\phi}) \ln N)$, completing the proof of the theorem. □

ACKNOWLEDGMENTS. We thank Meir Feder, Yuval Peres, Nick Littlestone and Michael Kearns for helpful suggestions and discussions of this material. We also thank the two anonymous referees for their valuable comments on an earlier draft of this paper.

REFERENCES

BIRGE, L., AND MASSART, P. 1993. Rates of convergence for minimum contrast estimators. *Prob. Theory Rel. Fields 97*, 113–150.

BLUMER, A., EHRENFEUCHT, A., HAUSSLER, D., AND WARMUTH, M. K. 1989. Learnability and the Vapnik–Chervonenkis dimension. *J. ACM 36*, 4, 929–965.

CESA-BIANCHI, N., FREUND, Y., HELMBOLD, D. P., HAUSSLER, D., SCHAPIRE, R. E., AND WARMUTH, M. K. 1993. How to use expert advice. In *Proceedings of the 25th Annual ACM Symposium on the Theory of Computing* (San Diego, Calif., May 16–18). ACM, New York, pp. 382–391.

CESA-BIANCHI, N., FREUND, Y., HELMBOLD, D. P., AND WARMUTH, M. K. 1996. On-line prediction and conversion strategies. *Mach. Learn.*, to appear.

CHUNG, T. H. 1994. Approximate methods for sequential decision making using expert advice. In *Proceedings of the 7th Annual ACM Conference on Computational Learning Theory* (New Brunswick, N.J., July 12–15). ACM, New York, pp. 183–189.

COVER, T. M. 1965. Behaviour of sequential predictors of binary sequences. In *Transactions of the 4th Prague Conference on Information Theory, Statistical Decision Functions, Random Processes*. Publishing House of the Czechoslovak Academy of Sciences.

COVER, T. M., AND SHANAR, A. 1977. Compound Bayes predictors for sequences with apparent Markov structure. *IEEE Trans. Syst. Man Cybernet. SMC-7*, 6 (June), 421–424.

DAWID, A. P. 1984. Statistical theory: The prequential approach. *J. Roy. Stat. Soc., Series A*, 278–292.

DAWID, A. 1991. Prequential analysis, stochastic complexity and Bayesian inference. In *Bayesian Statistics*, vol. 4. Oxford University Press, pp. 109–125.

DAWID, A. P. 1996. Prequential data analysis. *Curr. Iss. Stat. Inference*, to appear.

DESANTIS, A., MARKOWSKI, G., AND WEGMAN, M. N. 1988. Learning probabilistic prediction functions. In *Proceedings of the 1988 Workshop on Computational Learning Theory*. Morgan-Kaufmann, San Mateo, Calif., pp. 312–328.

FEDER, M., MERHAV, N., AND GUTMAN, M. 1992. Universal prediction of individual sequences. *IEEE Trans. Inf. Theory 38*, 1258–1270.

FIAT, A., FOSTER, D., KARLOFF, H., RABANI, Y., RAVID, Y., AND VISWANATHAN, S. 1991a. Competitive algorithms for layered graph traversal. In *Proceedings of the 32nd Annual Symposium on Foundations of Computer Science*. IEEE, New York, pp. 288–297.

FIAT, A., KARP, R., LUBY, M., MCGEOCH, L., SLEATOR, D., AND YOUNG, N. 1991b. Competitive paging algorithms. *J. Algorithms 12*, 688–699.

FIAT, A., RABANI, Y., AND RAVID, Y. 1994. Competitive k-server algorithms. *J. Comput Syst. Sci. 48*, 3, 410–428.

GALAMBOS, J. 1987. *The Asymptotic Theory of Extreme Order Statistics*, 2nd ed. R. E. Kreiger.

HANNAN, J. 1957. Approximation to Bayes risk in repeated play. In *Contributions to the Theory of Games*, vol. 3. Princeton University Press, Princeton, N.J., pp. 97–139.

HAUSSLER, D., AND BARRON, A. 1992. How well do Bayes methods work for on-line prediction of $\{+1, -1\}$ values? In *Proceedings of the 3rd NEC Symposium on Computation and Cognition*. SIAM.

HAUSSLER, D., KEARNS, M., LITTLESTONE, N., AND WARMUTH, M. K. 1991. Equivalence of models for polynomial learnability. *Inf. Comput. 95*, 129–161.

HAUSSLER, D., KEARNS, M., AND SCHAPIRE, R. 1994. Bounds on the sample complexity of Bayesian learning using information theory and the VC dimensions. *Mach. Learn. 14*, 84–114.

HAUSSLER, D., KIVINEN, J., AND WARMUTH, M. K. 1995. Tight worst-case loss bounds for predicting with expert advice. In *Computational Learning Theory: Second European Conference, EuroCOLT '95*. Springer-Verlag, New York, pp. 69–83.

HAUSSLER, D., LITTLESTONE, N., AND WARMUTH, M. K. 1994. Predicting $\{0, 1\}$-functions on randomly drawn points. *Inf. Comput. 115*, 2, 248–292.

HAYKIN, S. 1994. *Neural Networks: A Comprehensive Foundation*. MacMillan, New York.

HELMBOLD, D., AND WARMUTH, M. K. 1995. On weak learning. *J. Comput. Syst. Sci. 50*, 3 (June), 551–573.

KEARNS, M. J., AND SCHAPIRE, R. E. 1994. Efficient distribution-free learning of probabilistic concepts. *J. Comput. Syst. Sci. 48*, 3, 464–497.

KEARNS, M. J., SCHAPIRE, R. E., AND SELLIE, L. M. 1994. Toward efficient agnostic learning. *Mach. Learn. 17*, 115–141.

KIVINEN, J., AND WARMUTH, M. K. 1994. Using experts for predicting continuous outcomes. In *Computational Learning Theory: EuroCOLT '93*. Springer-Verlag, New York, pp. 109–120.

LITTLESTONE, N. 1989. From on-line to batch learning. In *Proceedings of the 2nd Annual Workshop on Computational Learning Theory*. Morgan-Kaufmann, San Mateo, Calif., pp. 269–284.

LITTLESTONE, N., LONG, P. M., AND WARMUTH, M. K. 1995. On-line learning of linear functions. *Computat. Complexity 5*, 1, 1–23.

LITTLESTONE, N., AND WARMUTH, M. K. 1994. The weighted majority algorithm. *Inf. Comput. 108*, 2, 212–261.

MERHAV, N., AND FEDER, M. 1993. Universal schemes for sequential decision for individual data sequences. *IEEE Trans. Inf. Theory, 39*, 4, 1280–1292.

RISSANEN, J. 1978. Modeling by shortest data description. *Automatica 14*, 465–471.

RISSANEN, J. 1986. Stochastic complexity and modeling. *Ann. Stat. 14*, 3, 1080–1100.

RISSANEN, J., AND LANGDON, G. G., JR. 1981. Universal modeling and coding. *IEEE Trans. Inf. Theory IT-27*, 1 (Jan.), 12–23.

SEUNG, H. S., SOMPOLINSKY, H., AND TISHBY, N. 1992. Statistical mechanics of learning from examples. *Phys. Rev A 45*, 8, 6056–6091.

SHTARKOV, YU. M. 1975. Coding of discrete sources with unknown statistics. In *Topics in Information Theory*. North-Holland, Amsterdam, The Netherlands, pp. 559–574.

SHTARKOV, YU. M. 1987. Universal sequential coding of single messages. *Prob. Inf. Transm. 23* (July–September), 175–186.

SOMPOLINSKY, H., TISHBY, N., AND SEUNG, H. S. 1992. Learning from examples in large neural networks. *Phys. Rev. Lett. 65*, 1683–1686.

STONE, C. J. 1977. Cross-validation: A review. *Math. Operationforsch. Statist. Ser Statist. 9*, 127–139.

TALAGRAND, M. 1994. Sharper bounds for Gaussian and empirical processes. *Ann. Prob. 22*, 1, 28–76.

VALIANT, L. G. 1984. A theory of the learnable. *Commun. ACM 27*, 11 (Nov.), 1134–1142.

VAPNIK, V. N. 1982. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, New York.

VAPNIK, V. 1992. Principles of risk minimization for learning theory. In *Advances in Neural Information Processing Systems*, vol. 4. John E. Moody, Steve J. Hanson, and Richard P. Lippman, eds. Morgan Kaufmann, San Mateo, Calif.

VOVK, V. G. 1990. Aggregating strategies. In *Proceedings of the 3rd Annual Workshop on Computational Learning Theory*. Morgan-Kaufmann, San Mateo, Calif., pp. 371–383.

VOVK, V. G. 1992. Universal forecasting algorithms. *Inf. Comput. 96*, 2 (Feb.), 245–277.

VOVK, V. G. 1993. A logic of probability, with application to the foundations of statistics. *J. Roy. Statis Soc, Ser. B-Methodolical 55*, 2, 317–351.

YAMANISHI, K. 1995. A loss bound model for on-line stochastic prediction algorithms. *Inf. Comput. 119*, 1, 39–54.

RECEIVED JUNE 1995; REVISED APRIL 1997; ACCEPTED MARCH 1997