On the Generalization Ability of On-Line Learning Algorithms

Nicolò Cesa-Bianchi, Alex Conconi, and Claudio Gentile

***Abstract*—In this paper, it is shown how to extract a hypothesis with small risk from the ensemble of hypotheses generated by an arbitrary on-line learning algorithm run on an independent and identically distributed (i.i.d.) sample of data. Using a simple large deviation argument, we prove tight data-dependent bounds for the risk of this hypothesis in terms of an easily computable statistic M_n associated with the on-line performance of the ensemble. Via sharp pointwise bounds on M_n, we then obtain risk tail bounds for kernel Perceptron algorithms in terms of the spectrum of the empirical kernel matrix. These bounds reveal that the linear hypotheses found via our approach achieve optimal tradeoffs between hinge loss and margin size over the class of all linear functions, an issue that was left open by previous results.**

A distinctive feature of our approach is that the key tools for our analysis come from the model of prediction of individual sequences; i.e., a model making no probabilistic assumptions on the source generating the data. In fact, these tools turn out to be so powerful that we only need very elementary statistical facts to obtain our final risk bounds.

***Index Terms*—Kernel functions, on-line learning, pattern recognition, perceptron algorithm, statistical learning theory.**

I. Introduction

We view pattern recognition as a problem of learning from examples. An *example* is a pair (x, y), where $x \in \mathcal{X}$ (which we call *instance*) is a data element and $y \in \mathcal{Y}$ is the *label* associated with it. Instances x are tuples of numerical and/or symbolic attributes. Labels y belong to a finite set of symbols (the class elements) or to an interval of the real line, depending on whether the task is classification or regression.

A learning algorithm for pattern recognition (or *learner* for short) takes in input a *training set*, i.e., a multiset of examples $(x_t, y_t) \in \mathcal{X} \times \mathcal{Y}$, and outputs a function $h : \mathcal{X} \to \mathcal{Y}$, assigning a label from $\mathcal{Y}$ to each element of $\mathcal{X}$. We call *hypothesis* any function returned by a learner. Learning algorithms generate hypotheses from specific families of functions, such as linear-threshold functions, decision trees, or multivariate polynomials. We call *hypothesis space* (denoted by $\mathcal{H}$) the family from which a learning algorithm picks its hypotheses.

We allow a learning algorithm to output hypotheses of the form $h : \mathcal{X} \to \mathcal{D}$, where $\mathcal{D}$ is a decision space not necessarily equal to $\mathcal{Y}$. The goodness of hypothesis h on example (x, y) is measured by the quantity $\ell(h(x), y)$, where $\ell : \mathcal{D} \times \mathcal{Y} \to \mathbb{R}$ is a nonnegative and bounded *loss function*. For instance, in binary classification we may take $\mathcal{Y} = \{-1, 1\}$, $\mathcal{D} = [-1, 1]$, and $\ell(h(x), y) = \frac{1}{2}|h(x) - y|$ (the absolute loss function).

We analyze learning algorithms within the framework of statistical pattern recognition (see, e.g., [1]). In this framework, all the examples (x, y) are generated by independent draws from a fixed and unknown probability distribution on $\mathcal{X} \times \mathcal{Y}$. This assumption allows us to view the training set as a statistical sample, and thus, to investigate the learning problem as a problem of statistical inference. In what follows, probabilities $\mathbb{P}$ and expectations $\mathbb{E}$ will be understood with respect to the fixed and unknown underlying distribution according to which all examples are drawn. Random variables are denoted in upper case and their realizations in lower case. We use Z to denote the pair (X, Y) of random variables X and Y taking values in $\mathcal{X}$ and $\mathcal{Y}$, respectively, We also write the training set as a vector-valued random variable

$$Z^n = (Z_1, \dots, Z_n) = ((X_1, Y_1), \dots, (X_n, Y_n)).$$

In general, we would like a hypothesis h to predict well on examples drawn from the same source that generated the training set for h. In statistical pattern recognition, the success of a hypothesis h is measured by the *risk* of h, denoted by $\mathtt{risk}(h)$. This is the expected loss of h on an example (X, Y) drawn from the underlying distribution, $\mathtt{risk}(h) = \mathbb{E}\ell(h(X), Y)$. We identify a generic learner $\mathtt{A}$ with the random hypothesis $\widehat{H} = \widehat{H}(Z^n)$ returned by $\mathtt{A}$ when Z^n is fed in input. Our goal is to keep the risk of $\widehat{H}$ small on most sample realizations, that is,

$$\mathbb{P}\left(\mathtt{risk}(\widehat{H}) \geq \inf_{h \in \mathcal{H}} \mathtt{risk}(h) + \varepsilon\right) \leq \delta$$

for small enough $\varepsilon > 0$ and $0 < \delta < 1$. Here the probability is taken with respect to the distribution of the training sample Z^n.

To achieve this goal, we can use the method of uniform convergence, whose study was pioneered by Vapnik and Chervonenkis (VC) [2] (see also [3], [4]). Let $\mathtt{risk}_{\mathrm{emp}}(h)$ be the empirical risk of h on a sample Z^n

$$\mathtt{risk}_{\mathrm{emp}}(h) = \frac{1}{n}\sum_{t=1}^{n} \ell(h(X_t), Y_t).$$

Uniform convergence means that, for all probability distributions, the empirical risk of h is, with high probability, close to its true risk uniformly over all $h \in \mathcal{H}$. In the case of a class $\mathcal{H}$ of

Manuscript received March 4, 2003; revised February 17, 2004. This work was supported in part by the European Commission under the KerMIT Project IST-2001-25341. The material in this paper was presented in part at the Conference on Advances in Neural Information Processing Systems, Vancouver, BC, Canada, Dec. 2001. An extended abstract of this paper appeared in *Advances in Neural Information Processing Systems 14*. Cambridge, MA: MIT Press, 2002, pp. 359–366.

N. Cesa-Bianchi and A. Conconi are with DSI, the University of Milan, 20135 Milan, Italy.

G. Gentile is with the Dipartimento di Informatica e Comunicazione, University of Insubria, 21100 Varese, Italy.

Communicated by A. E. Nobel, Associate Editor for Nonparametric Estimation, Classification, and Neural Networks.

Digital Object Identifier 10.1109/TIT.2004.833339

$\{0,1\}$-valued functions, a sufficient (and also necessary) condition for uniform convergence is that $\mathcal{H}$ has finite VC dimension d—similar conditions apply to multivalued or real-valued function classes and bounded losses. If this condition holds then, for each $0 < \delta < 1$ and sample size n

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}} |\mathtt{risk}_{\mathrm{emp}}(h) - \mathtt{risk}(h)| \geq c\sqrt{\frac{d+\ln(1/\delta)}{n}}\right) \leq \delta \tag{1}$$

holds (for a proof of this result see, e.g., [5]).

Uniform convergence implies that $\mathcal{H}$ can be learned by the empirical risk minimizer, i.e., by the algorithm returning the hypothesis

$$\widehat{H} = \operatorname*{arginf}_{h\in\mathcal{H}} \mathtt{risk}_{\mathrm{emp}}(h).$$

Once we have a uniform convergence result like (1), the risk analysis for $\widehat{H}$ is immediate. Let $h^* = \operatorname{arginf}_{h\in\mathcal{H}} \mathtt{risk}(h)$. Then, with probability at least $1-\delta$

$$\begin{aligned}\mathtt{risk}(\widehat{H}) &\leq \mathtt{risk}_{\mathrm{emp}}(\widehat{H}) + c\sqrt{\frac{d+\ln(2/\delta)}{n}}\\ &\leq \mathtt{risk}_{\mathrm{emp}}(h^*) + c\sqrt{\frac{d+\ln(2/\delta)}{n}}\\ &\leq \mathtt{risk}(h^*) + 2c\sqrt{\frac{d+\ln(2/\delta)}{n}}\end{aligned}$$

where we applied (1) in the first and the last step.

A different approach to uniform convergence, pioneered in [6], replaces the square-root term in (1) with the random quantity

$$S_n(Z^n) + c\sqrt{\frac{\ln(1/\delta)}{n}}$$

where S_n is a sample statistic depending on $\mathcal{H}$ and c is a universal constant. For example, $S_n(Z^n)$ can be the empirical VC entropy [7], [8], the Rademacher complexity [9], or the maximum discrepancy [10] of the class $\mathcal{H}$. In general, this approach is advantageous when the mean of S_n is significantly smaller than $\sqrt{d/n}$ and when large deviations of S_n are unlikely. In these cases, such "data-dependent" uniform convergence bounds are stronger than those based on the VC dimension since, with high probability, we have

$$S_n(Z^n) \approx \mathbb{E}S_n \ll \sqrt{d/n}.$$

In some cases, the statistic S_n directly depends on the empirical behavior of the hypothesis $h \in \mathcal{H}$ under consideration, yielding bounds of the form

$$\mathbb{P}\Bigg((\exists h\in\mathcal{H})\ \mathtt{risk}(h) \geq \mathtt{risk}_{\mathrm{emp}}(h) + S_n(h,Z^n) + c\sqrt{\frac{1}{n}\ln(1/\delta)}\Bigg) \leq \delta. \tag{2}$$

Prominent examples of this kind are the bounds for linear-threshold classifiers, where S_n depends on the margin of h [11]–[14], and the bounds for Bayesian mixtures, where S_n depends on the Kullback–Leibler divergence between the data-dependent mixture coefficients and the *a priori* coefficients [15]. Note that bounds of the form (2) leave open the algorithmic problem of finding the hypothesis $h \in \mathcal{H}$ optimizing the tradeoff between the terms $\mathtt{risk}_{\mathrm{emp}}(h)$ and $S_n(h, Z^n)$.

The techniques based on uniform convergence, which we have seen so far, lead to probabilistic statements which hold simultaneously for all hypotheses in the class. A different approach used to derive data-dependent risk bounds yields statements which only hold for the hypotheses generated by learning algorithms satisfying certain properties. Examples along these lines are the notions of self-bounding learners [16], [17], algorithmic stability [18], and algorithmic luckiness [19].

In this paper, we follow a similar idea and develop a general framework for analyzing the risk of hypotheses generated by on-line learners, a specific class of learning algorithms. Exploiting certain properties of on-line learners, we prove new data-dependent results via elementary large deviation theory (Section II), thus avoiding the sophisticated statistical tools required by risk analyses based on uniform convergence. We borrow results from the literature on prediction of individual sequences (see, e.g., [20]–[24] for early references on the subject, and [25]–[31] for specific work on the pattern classification problem). Based on strong pointwise bounds on the sample statistic governing the risk for a specific on-line learner, the kernel Perceptron algorithm, we derive sharp tail risk bounds for linear hypotheses in terms of the spectrum of the empirical kernel (Gram) matrix (Section III). Though our bounds are not easily comparable with previous spectral bounds based on uniform convergence, there are two reasons for which ours might be preferable: The empirical kernel matrices occurring in our bounds are sparse, as they only contain a subset of "support" examples; moreover, the linear hypotheses found via our approach achieve optimal tradeoffs between empirical hinge loss and margin size over the class of all linear functions, thus solving the algorithmic problem left open by results like (2).

II. Risk Analysis for On-Line Algorithms

Unlike standard learning algorithms, on-line learners take in input a hypothesis $h \in \mathcal{H}$ and an example (x,y) and return a new hypothesis $h' \in \mathcal{H}$ (on empty inputs, a *default hypothesis* h_0 is returned). We now illustrate how an on-line algorithm $\mathtt{A}$ is run on a sample $Z^n = ((X_1,Y_1),\ldots,(X_n,Y_n))$. First, $\mathtt{A}$ uses the first example (X_1,Y_1) and the default hypothesis h_0 to generate the first hypothesis H_1. Note that H_1 is a random element of $\mathcal{H}$ as it depends on the random example (X_1,Y_1). Next, it uses the second example (X_2,Y_2) and the first hypothesis H_1 to generate the second hypothesis H_2, and so on. At the end of this process we obtain a sequence $H_0,H_1,\ldots,H_n$ of (not necessarily distinct) hypotheses, where $H_0 = h_0$ and each H_t is obtained from H_{t-1} and (X_t,Y_t).

Our goal is to use the ensemble $H_0,H_1,\ldots,H_n$ of hypotheses generated by an on-line learner $\mathtt{A}$ to obtain a hypothesis with low risk. An obvious choice is to pick H_n, i.e., the hypothesis that depends on the whole training set Z^n. However, without making specific assumptions on the way $\mathtt{A}$

operates, we cannot say about H_n much more than what could be said based on standard uniform convergence arguments. In what follows, we propose a different approach which derives a hypothesis from the ensemble $H_0, H_1, \dots, H_{n-1}$ (we actually *discard* the last hypothesis H_n, even though we do so for purely technical reasons).

Given a sample Z^n and an on-line learner $\mathtt{A}$, we call H_0, $H_1, \dots, H_{n-1}$ the ensemble of hypotheses generated by $\mathtt{A}$. A central role in our analysis is played by the sample statistic

$$M_n = M_n(Z^n) = \frac{1}{n}\sum_{t=1}^{n} \ell(H_{t-1}(X_t), Y_t).$$

The statistic M_n can be easily computed as the on-line algorithm is run on Z^n. M_n measures how good on average each H_{t-1} did on the *next* example (X_t, Y_t). In agreement with this intuition, we now prove that M_n is close to the ensemble's average risk. Since we are working with bounded losses, there is no loss of generality in assuming that the loss function ℓ satisfies $0 \le \ell \le 1$.

Proposition 1: Let $H_0, \dots, H_{n-1}$ be the ensemble of hypotheses generated by an arbitrary on-line algorithm $\mathtt{A}$ working with a loss ℓ satisfying $0 \le \ell \le 1$. Then, for any $0 < \delta \le 1$

$$\mathbb{P}\left(\frac{1}{n}\sum_{t=1}^{n}\mathtt{risk}(H_{t-1}) \ge M_n + \sqrt{\frac{2}{n}\ln\frac{1}{\delta}}\right) \le \delta.$$

Proof: For each $t = 1, \dots, n$ set

$$V_{t-1} = \mathtt{risk}(H_{t-1}) - \ell(H_{t-1}(X_t), Y_t).$$

We have

$$\frac{1}{n}\sum_{t=1}^{n} V_{t-1} = \frac{1}{n}\sum_{t=1}^{n}\mathtt{risk}(H_{t-1}) - M_n.$$

Furthermore, $-1 \le V_{t-1} \le 1$ holds since ℓ takes values in $[0,1]$. Finally, using $\mathbb{E}_t$ to denote the conditional expectation $\mathbb{E}[\cdot \mid Z_1, \dots, Z_{t-1}]$

$$\mathbb{E}_t V_{t-1} = \mathtt{risk}(H_{t-1}) - \mathbb{E}_t \ell(H_{t-1}(X_t), Y_t) = 0.$$

A direct application of the Hoeffding–Azuma inequality (a generalization of Chernoff–Hoeffding bounds to sums of conditionally zero-mean bounded random variables [32]) to the random variables $V_0, \dots, V_{n-1}$ proves the lemma. □

We will be using this simple concentration result several times in the rest of this section.

A. Risk Analysis for Convex Losses

Using Proposition 1, it is now possible to derive a data-dependent risk bound of the same form as the one mentioned in Section I. Suppose that the decision space $\mathcal{D}$ of $\mathtt{A}$ is a convex set and that the loss function ℓ is convex in its first argument. Define the *average hypothesis*

$$\overline{H} = \frac{1}{n}\sum_{t=1}^{n} H_{t-1}.$$

The assumption on $\mathcal{D}$ ensures that $\overline{H}$ is indeed a map from $\mathcal{X}$ to $\mathcal{D}$. It is now fairly easy to prove a data-dependent bound on the risk of $\overline{H}$.

Corollary 2: Let $H_0, \dots, H_{n-1}$ be the ensemble of hypotheses generated by an arbitrary on-line algorithm $\mathtt{A}$ working with a loss function ℓ which is convex in its first argument and satisfying $0 \le \ell \le 1$. Then, for any $0 < \delta \le 1$

$$\mathbb{P}\left(\mathtt{risk}(\overline{H}) \ge M_n + \sqrt{\frac{2}{n}\ln\frac{1}{\delta}}\right) \le \delta.$$

Proof: Using Jensen inequality and the linearity of expectation

$$\begin{aligned}\mathtt{risk}(\overline{H}) &= \mathbb{E}\ell\left(\frac{1}{n}\sum_{t=1}^{n} H_{t-1}(X), Y\right)\\ &\le \frac{1}{n}\sum_{t=1}^{n}\mathbb{E}\ell(H_{t-1}(X), Y)\\ &= \frac{1}{n}\sum_{t=1}^{n}\mathtt{risk}(H_{t-1}).\end{aligned}$$

An application of Proposition 1 concludes the proof. □

As a remark, we note that a version of Corollary 2 restricted to the absolute loss was given in [33].

B. Risk Analysis for General (Bounded) Losses

Proposition 1 tells us that at least one element in the ensemble has risk not larger than M_n with high probability. In this subsection, we show how to identify such a hypothesis with high probability and with no conditions on the loss function other than boundedness. Our approach is related to a result of Littlestone [34]. However, unlike Littlestone's, our technique does not require a cross-validation set.

The main idea for finding a hypothesis in the ensemble whose risk is close to the ensemble's average risk is to compute the empirical risk of each hypothesis H_t on the sequence of remaining examples. Then, to compensate for the fact that the hypotheses have been tested on portions of the sample of different length, a different penalization term is added to the empirical risk of each H_t.

Formally, define the penalized empirical risk of hypothesis H_t by $\mathtt{risk}_{\text{emp}}(H_t, t+1) + c_\delta(n-t)$ where

$$\mathtt{risk}_{\text{emp}}(H_t, t+1) = \frac{1}{n-t}\sum_{i=t+1}^{n}\ell(H_t(X_i), Y_i)$$

is the empirical risk of H_t on the remaining sample $Z_{t+1}, \dots, Z_n$ and

$$c_\delta(x) = \sqrt{\frac{1}{2x}\ln\frac{n(n+1)}{\delta}}, \qquad x = 1, \dots, n$$

is a penalization function. Note that the penalized empirical risk is a sample statistic. Note also that the penalization function explicitly depends on the confidence parameter δ. This somewhat unusual dependence comes from the need of matching the size of the confidence interval, at confidence level $1-\delta$, provided by the Chernoff–Hoeffding bound.

Our learner returns the hypothesis $\widehat{H}$ minimizing the penalized risk estimate over all hypotheses in the ensemble, i.e.,

$$\widehat{H} = \operatorname*{argmin}_{0\le t<n}\left(\texttt{risk}_{\mathrm{emp}}(H_t, t+1) + c_\delta(n-t)\right). \tag{3}$$

Our proof technique builds on the concentration result contained in Proposition 1, and proceeds essentially in two steps. We first prove that the true risk of $\widehat{H}$ is close to the minimal penalized risk over the random hypotheses in the ensemble (Lemma 3); then we exploit the fact that, simultaneously for all these random hypotheses, the true risk is close to the corresponding empirical risk (Theorems 4 and 5).

The proof of the next lemma is given in the appendix.

Lemma 3: Let $H_0,\dots,H_{n-1}$ be the ensemble of hypotheses generated by an arbitrary on-line algorithm $\mathtt{A}$ working with a loss ℓ satisfying $0\le\ell\le 1$. Then, for any $0<\delta\le 1$, the hypothesis $\widehat{H}$ satisfies

$$\mathbb{P}\left(\texttt{risk}(\widehat{H}) > \min_{0\le t<n}\left(\texttt{risk}(H_t) + 2c_\delta(n-t)\right)\right) \le \delta.$$

The following theorem is our main result.

Theorem 4: Let $H_0,\dots,H_{n-1}$ be the ensemble of hypotheses generated by an arbitrary on-line algorithm $\mathtt{A}$ working with a loss ℓ satisfying $0\le\ell\le 1$. Then, for any $0<\delta\le 1$, the hypothesis $\widehat{H}$ minimizing the penalized empirical risk based on $c_{\delta/2}$ (i.e., the one obtained by replacing c_δ with $c_{\delta/2}$ in (3)) satisfies

$$\mathbb{P}\left(\texttt{risk}(\widehat{H}) \ge M_n + 6\sqrt{\frac{1}{n}\ln\frac{2(n+1)}{\delta}}\right) \le \delta.$$

As a matter of fact, we will be proving the tighter statement contained in the following Theorem 5. The bound in this theorem is formally similar to model selection inequalities, such as those in [10]. Theorem 4 is obtained by replacing the minimum over $0\le t<n$ in the statement of Theorem 5 with $t=0$.

Theorem 5: Under the assumptions of Theorem 4, we have that

$$\mathbb{P}\left(\texttt{risk}(\widehat{H}) \ge \min_{0\le t<n}\left(M_{t,n} + 6\sqrt{\frac{1}{n-t}\ln\frac{2(n+1)}{\delta}}\right)\right)$$

is at most δ, where

$$M_{t,n} = \frac{1}{n-t}\sum_{i=t+1}^{n}\ell(H_{i-1}(X_i), Y_i).$$

Proof: Applying Lemma 3 with $c_{\delta/2}$ we obtain

$$\mathbb{P}\left(\texttt{risk}(\widehat{H}) > \min_{0\le t<n}\left(\texttt{risk}(H_t) + 2c_{\delta/2}(n-t)\right)\right) \le \frac{\delta}{2}.$$

We then observe that

$$\begin{aligned}
&\min_{0\le t<n}\left(\texttt{risk}(H_t) + 2c_{\delta/2}(n-t)\right)\\
&\quad= \min_{0\le t<n}\min_{t\le i<n}\left(\texttt{risk}(H_i) + 2c_{\delta/2}(n-i)\right)\\
&\quad\le \min_{0\le t<n}\frac{1}{n-t}\sum_{i=t}^{n-1}\left(\texttt{risk}(H_i) + 2c_{\delta/2}(n-i)\right)\\
&\quad= \min_{0\le t<n}\left(\frac{1}{n-t}\sum_{i=t}^{n-1}\texttt{risk}(H_i)\right.\\
&\qquad\qquad\left. + \frac{2}{n-t}\sum_{i=t}^{n-1}\sqrt{\frac{1}{2(n-i)}\ln\frac{2n(n+1)}{\delta}}\right)\\
&\quad< \min_{0\le t<n}\left(\frac{1}{n-t}\sum_{i=t}^{n-1}\texttt{risk}(H_i)\right.\\
&\qquad\qquad\left. + \frac{2}{n-t}\sum_{i=t}^{n-1}\sqrt{\frac{1}{n-i}\ln\frac{2(n+1)}{\delta}}\right)\\
&\quad\le \min_{0\le t<n}\left(\frac{1}{n-t}\sum_{i=t}^{n-1}\texttt{risk}(H_i)\right.\\
&\qquad\qquad\left. + 4\sqrt{\frac{1}{n-t}\ln\frac{2(n+1)}{\delta}}\right),
\end{aligned}$$

where the last inequality follows from $\sum_{i=1}^{n-t}\sqrt{1/i}\le 2\sqrt{n-t}$.

Now, it is clear that Proposition 1 can be immediately generalized to the following set of inequalities, one for each $t=0,\dots,n-1$:

$$\mathbb{P}\left(\frac{1}{n-t}\sum_{i=t+1}^{n}\texttt{risk}(H_{i-1}) \ge M_{t,n} + \sqrt{\frac{2}{n-t}\ln\frac{1}{\delta}}\right) \le \delta. \tag{4}$$

Hence, setting for brevity

$$K_t = M_{t,n} + \sqrt{\frac{2}{n-t}\ln\frac{2n}{\delta}} + 4\sqrt{\frac{1}{n-t}\ln\frac{2(n+1)}{\delta}}$$

we can write

$$\begin{aligned}
&\mathbb{P}\left(\min_{0\le t<n}\left(\texttt{risk}(H_t) + 2c_{\delta/2}(n-t)\right) \ge \min_{0\le t<n}K_t\right)\\
&\quad\le \mathbb{P}\left(\min_{0\le t<n}\left(\frac{1}{n-t}\sum_{i=t}^{n-1}\texttt{risk}(H_i)\right.\right.\\
&\qquad\qquad\left.\left. + 4\sqrt{\frac{1}{n-t}\ln\frac{2(n+1)}{\delta}}\right) \ge \min_{0\le t<n}K_t\right)\\
&\quad\le \sum_{t=0}^{n-1}\mathbb{P}\left(\frac{1}{n-t}\sum_{i=t}^{n-1}\texttt{risk}(H_i)\right.\\
&\qquad\qquad\left. + 4\sqrt{\frac{1}{n-t}\ln\frac{2(n+1)}{\delta}} \ge K_t\right)\\
&\quad= \sum_{t=0}^{n-1}\mathbb{P}\left(\frac{1}{n-t}\sum_{i=t}^{n-1}\texttt{risk}(H_i) \ge M_{t,n}\right.\\
&\qquad\qquad\left. + \sqrt{\frac{2}{n-t}\ln\frac{2n}{\delta}}\right) \le \frac{\delta}{2}
\end{aligned}$$

where in the last inequality we used (4).

Combining with Lemma 3, we have that with probability at least $1-\delta$

$$\begin{aligned}
\texttt{risk}(\widehat{H}) &\le \min_{0\le t<n}K_t\\
&< \min_{0\le t<n}\left(M_{t,n} + 6\sqrt{\frac{1}{n-t}\ln\frac{2(n+1)}{\delta}}\right)
\end{aligned}$$

thereby concluding the proof. □

It should be clear that both Theorems 4 and 5 could be used to prove generalization bounds for specific algorithms. In fact, for the sake of comparison to the existing literature, in the rest of this paper we will be essentially employing Theorem 4. Needless to say, one could recast our results (Theorems 6 and 7 in Section III) in the style of Theorem 5, instead.

III. Applications

As shown in Section II-B, we know how to extract, from the ensemble $H_0, H_1, \dots, H_{n-1}$ of hypotheses generated by an arbitrary on-line learner $\mathtt{A}$, a hypothesis $\widehat{H}$ whose risk satisfies

$$\mathtt{risk}(\widehat{H}) \le M_n + c\sqrt{\frac{\ln(n/\delta)}{n}}$$

with high probability, where

$$M_n = M_n(Z^n) = \frac{1}{n}\sum_{t=1}^{n} \ell(H_{t-1}(X_t), Y_t)$$

and c is a universal constant. We now show applications of this bound to kernel-based linear algorithms Using pointwise bounds on M_n (i.e., bounds on $M_n(z^n)$ holding for every possible realization z^n of the sample Z^n) we obtain sharp tail risk bounds in terms of the spectral properties of the empirical kernel Gram matrix generated during the algorithm's run.

In the sequel we will be focusing on binary classification tasks with linear-threshold learners. A loss function naturally associated to such tasks is the 0–1 loss, which we will be handling through Theorem 4. As a matter of fact, analogs of our results exist, e.g., for linear regression problems with square loss. Clearly, for such convex losses we could use Corollary 2, instead.

For binary classification tasks we take $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{D} = \mathcal{Y} = \{-1, 1\}$. The loss function is the standard 0–1 loss: $\ell(\widehat{y}, y) = 1$ if $\widehat{y} \ne y$ and $\ell(\widehat{y}, y) = 0$ otherwise. On-line linear-threshold learners generate hypotheses of the form[1] $h(x) = \textsc{sgn}(w^\top x)$, where $w \in \mathbb{R}^d$ is a so-called *weight vector* associated with hypothesis h.

We begin by considering the classical Perceptron algorithm [35]–[37] in its dual kernel form, as investigated in, e.g., [27], [38]. For an introduction to kernels in learning theory the reader is referred to [39], or to [40] for a more in-depth monography. Here, we just recall the following basic definitions.

A *kernel* is a nonnegative function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ satisfying

$$\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j K(x_i, x_j) \ge 0$$

for all $\alpha_1, \dots, \alpha_m \in \mathbb{R}$, $x_1, \dots, x_m \in \mathbb{R}^d$, and $m \in \mathbb{N}$ (such functions are also called positive definite). Given a kernel, we can define the linear space

$$\mathcal{V}_K = \Bigg\{ f(\cdot) = \sum_{i=1}^{m} \alpha_i K(x_i, \cdot) : \\ \alpha_i \in \mathbb{R}, x_i \in \mathbb{R}^d, i = 1, \dots, m, m \in \mathbb{N} \Bigg\}$$

with norm defined by

$$\|f\| = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j K(x_i, x_j)}.$$

If this space is completed, by adding all limit points of sequences $f_1, f_2, \dots \in \mathcal{V}_K$ that are convergent in the norm $\|f\|$, the resulting space $\mathcal{H}_K$ is called reproducing kernel Hilbert space.

The hypothesis space for the kernel Perceptron algorithm, run with kernel K, is the space of functions $\textsc{sgn}(f)$ for $f \in \mathcal{H}_K$. The algorithm's initial hypothesis H_0 is the constant -1 function. The tth hypothesis H_t is defined by

$$H_t(x) = \textsc{sgn}\left(\sum_{i \in \mathcal{M}_t} y_i K(x_i, x)\right)$$

where $\mathcal{M}_t$ is the set of indices of previously mistaken examples, i.e., the set of all s such that $1 \le s < t$ and $H_{s-1}(x_s) \ne y_s$.

We will now prove a specialization of Theorem 4 to the kernel Perceptron algorithm by using a known bound on M_n for the ensemble of hypotheses generated by this algorithm. The bound on M_n uses a loss function different from the 0–1 loss. Define the *hinge loss* $\ell_\gamma : \mathbb{R} \times \{-1, 1\} \to \mathbb{R}$ *at margin* $\gamma > 0$ (see, e.g., [27], [28]) by

$$\ell_\gamma(\widehat{y}, y) = \max\left\{0, 1 - \frac{y\widehat{y}}{\gamma}\right\}. \tag{5}$$

Note that when using linear threshold functions the hinge loss is an upper bound on the 0–1 loss; i.e., $\ell(\textsc{sgn}(\widehat{y}), y) \le \ell_\gamma(\widehat{y}, y)$, for any $y \in \{-1, +1\}$, $\widehat{y} \in \mathbb{R}$, and $\gamma > 0$. For any function $f : \mathcal{X} \to \mathbb{R}$ and sample $Z^n = ((X_1, Y_1), \dots, (X_n, Y_n))$ define

$$D_{\gamma,n}(f, Z^n) = \frac{1}{n}\sum_{t=1}^{n} \ell_\gamma(f(X_t), Y_t).$$

We are now ready to state and prove the risk bound for the kernel Perceptron algorithm.

Theorem 6: Let $H_0, \dots, H_{n-1}$ be the ensemble of hypotheses generated by the kernel Perceptron algorithm using kernel K. Then, for any $0 < \delta \le 1$, the hypothesis $\widehat{H}$ minimizing the penalized empirical risk based on $c_{\delta/2}$ (i.e., the one obtained by replacing c_δ with $c_{\delta/2}$ in (3)) satisfies

$$\mathtt{risk}(\widehat{H}) \le \inf_{f \in \mathcal{H}_K : \|f\| \le 1} \inf_{\gamma > 0} \Bigg(D_{\gamma,n}(f, Z^n) \\ + \frac{1}{\gamma n}\sqrt{\sum_{t \in \mathcal{M}} K(X_t, X_t)} \Bigg) + 6\sqrt{\frac{1}{n}\ln\frac{2(n+1)}{\delta}} \tag{6}$$

with probability at least $1 - \delta$, where[2]

$$\mathcal{M} = \{1 \le t \le n : H_{t-1}(X_t) \ne Y_t\}.$$

Proof: For each sample realization z^n, we upper-bound $M_n(z^n)$ using known generalizations of the Perceptron convergence theorem (see, e.g., [26], [29]). An application of Theorem 4 concludes the proof. □

It is interesting to compare the above bound with the result obtained in [9, Theorem 21] by Bartlett and Mendelson using

[1] Here and in what follows we use $(\cdot)^\top$ to denote vector transposition and $\textsc{sgn}(\cdot)$ to denote the signum function.

[2] From an algorithmic standpoint, it should be observed that the minimum of (3) can only be achieved on indices $t \in \mathcal{M}$. The hypothesis $\widehat{H}$ obtained from the kernel Perceptron algorithm can then be computed by counting only the number of mistaken examples of the hypotheses H_t such that $t \in \mathcal{M} \cup \{0\}$.

uniform convergence techniques. They prove that, for all fixed $\gamma, B > 0$

$$\mathtt{risk}(\mathrm{SGN}(F)) \le D'_{\gamma,n}(F, Z^n) + \frac{4B}{\gamma n}\sqrt{\sum_{i=1}^{n} K(X_i, X_i)} + \left(\frac{8}{\gamma} + 1\right)\sqrt{\frac{1}{2n}\ln\frac{4}{\delta}} \quad (7)$$

holds with probability at least $1 - \delta$ simultaneously for all functions F in the class of all (random) functions of the form

$$F(x) = \sum_{t=1}^{n} \alpha_t K(X_t, x), \qquad \alpha_1, \dots, \alpha_n \in \mathbb{R}$$

and such that $\sum_{i,j} \alpha_i \alpha_j K(X_i, X_j) \le B^2$. The quantity $D'_{\gamma,n}$ is similar to (5), but is based on a loss ℓ'_γ defined by

$$\ell'_\gamma(\widehat{y}, y) = \begin{cases} \ell_\gamma(\widehat{y}, y), & \text{if } \ell_\gamma(\widehat{y}, y) \le 1 \\ 1, & \text{otherwise.} \end{cases}$$

While comparing (6) with (7), one should observe that bound (7) holds for all functions F of the above form; moreover, the hinge loss terms obey the inequality $D'_{\gamma,n}(f, Z^n) \le D_{\gamma,n}(f, Z^n)$. On the other hand, the fact that (6) holds for a specific function bears some crucial advantages over (7). In fact, it solves the algorithmical issue of finding the function $f \in \mathcal{H}_K$ optimizing the choice of γ. As a consequence, our bound does not have a size parameter B (which has to be traded off against $D'_{\gamma,n}$). Moreover, the main square-root term of our bound only contains the trace (i.e., the sum of the eigenvalues) of a small submatrix of the kernel Gram matrix.

The bound of Theorem 6 can be improved by using a recently proposed variant [26] of the kernel Perceptron algorithm. This variant, called (kernel) second-order Perceptron algorithm, generates ensembles of hypotheses H_t of the following form. Fix a sample realization $(x_1, y_1), \dots, (x_n, y_n)$ and, as before, let $\mathcal{M}_t = \{1 \le s < t : H_{s-1}(x_s) \ne y_s\}$, the set of indices of previously mistaken examples. Then

$$H_t(x) = \mathrm{SGN}\left(\boldsymbol{y}_t^\top (aI + G_t)^{-1} \boldsymbol{\kappa}_t(x)\right)$$

where

- G_t is the $|\mathcal{M}_t| \times |\mathcal{M}_t|$ sparse kernel Gram matrix with entries $K(x_i, x_j)$ for $i, j \in \mathcal{M}_t$;
- I is the $|\mathcal{M}_t| \times |\mathcal{M}_t|$ identity matrix;
- $\boldsymbol{y}_t$ is the vector of elements y_i, for $i \in \mathcal{M}_t$;
- $\boldsymbol{\kappa}_t(x)$ is the vector of elements $K(x_i, x)$, for $i \in \mathcal{M}_t$;
- a is a nonnegative parameter.

We remark that the inverse matrix $(aI + G_t)^{-1}$ does not have to be recomputed at each step. Indeed, standard techniques allow to incrementally update the inverse in time linear in the size of the matrix.

For a detailed analysis of this algorithm we refer the reader to [26]. Here we just note that for large values of a, the second-order Perceptron algorithm approximates the behavior of the standard (first-order) Perceptron algorithm. Using a pointwise bound on M_n proven in [26], we can show the following result, which uses the same notation as the statement of Theorem 6. This bound allows us to replace the trace of the kernel Gram matrix occurring in both (6) and (7) with more detailed spectral information.

Theorem 7: Let $H_0, \dots, H_{n-1}$ be the ensemble of hypotheses generated by the kernel second-order Perceptron algorithm using kernel K and run with input parameter $a > 0$. Then, for any $0 < \delta \le 1$, the hypothesis $\widehat{H}$ minimizing the penalized empirical risk based on $c_{\delta/2}$ satisfies (8) at the bottom of the page, with probability at least $1 - \delta$, where $\Lambda_1, \dots, \Lambda_{|\mathcal{M}|}$ are the (random) eigenvalues of the kernel Gram matrix including only those instances X_t such that $t \in \mathcal{M}$.

$$\mathtt{risk}(\widehat{H}) \le \inf_{f \in \mathcal{H}_K : \|f\| \le 1} \inf_{\gamma > 0} \left(D_\gamma(f, Z^n) + \frac{1}{\gamma n}\sqrt{\left(a + \sum_{t \in \mathcal{M}} f(X_t)^2\right) \sum_{t \in \mathcal{M}} \ln(1 + \Lambda_t / a)} \right) + 6\sqrt{\frac{1}{n}\ln\frac{2(n+1)}{\delta}} \quad (8)$$

A simple comparison between (6) and (8) might go as follows (the reader is referred to [26] for a more detailed argument). First, notice that the two bounds only differ in the kernel/eigenvalue terms under the square root sign. Now, the sum $\sum_{t \in \mathcal{M}} K(X_t, X_t)$ in (6) is actually the trace of the kernel Gram matrix including only the instances X_t with $t \in \mathcal{M}$. In turn, this trace equals the sum of the eigenvalues $\sum_{t \in \mathcal{M}} \Lambda_t$. Therefore, (6) has a *linear* dependence on the (random) eigenvalues Λ_t, while (8) has only a *logarithmic* dependence. The price we pay for this logarithmic dependence is the further factor $a + \sum_{t \in \mathcal{M}} f(X_t)^2$. Now, if a function f achieving a small value of D_γ tends to be aligned to an eigenvector of the kernel Gram matrix with a small eigenvalue, then this factor tends to be small, too. In fact, if f is perfectly aligned with an eigenvector with eigenvalue Λ, then $\sum_{t \in \mathcal{M}} f(X_t)^2$ is exactly equal to Λ. Hence, if parameter a is suitably chosen, then the product

$$\left(a + \sum_{t \in \mathcal{M}} f(X_t)^2\right) \sum_{t \in \mathcal{M}} \ln(1 + \Lambda_t / a)$$

occurring in (8) can be much smaller then the trace $\sum_{t \in \mathcal{M}} \Lambda_t$ occurring in (6).

Other examples of risk bounds involving the spectrum of the kernel Gram matrix have been proven in [8 (see Theorem 5.2 therein)] via uniform convergence techniques. These bounds are not readily comparable to ours. In fact, the results in [8] are proven through covering numbers arguments which consider the kernel Gram matrix G of the *whole* sample, whereas our Theorem 7 considers instead a submatrix of G including only instances X_i with $i \in \mathcal{M}$. On the other hand, the results in [8] are expressed in terms of the "large" eigenvalues of G only—where "large" is defined in terms of the margin of the data, whereas our bounds are in terms of all the eigenvalues of this submatrix (note, however, that the eigenvalues of this submatrix cannot be larger than the corresponding eigenvalues of G; rather, they are usually quite smaller). Furthermore, unlike our bounds, the bounds proven in [8] appear to rely on the assumption that there exists some $F \in \mathcal{F}$ achieving zero risk.

Further data-dependent bounds for kernel-based linear algorithms are derived in [11], [18], [19]. These bounds are not expressed in terms of the spectrum of the kernel matrix, and thus are not easily comparable to ours.

IV. Conclusion and Open Problems

In this paper, we have shown how to obtain sharp data-dependent tail bounds on the risk of hypotheses generated by on-line learning algorithms. The key analytical tools used in the proofs are borrowed from the model of prediction of individual sequences, an on-line learning model making no stochastic assumptions on the way the sequence of examples is generated. Surprisingly, these tools turn out to be so powerful that we only need very elementary statistical inequalities to complete the argument and obtain our final risk bounds.

It is interesting to note that *any* learning algorithm can be turned into an on-line algorithm (possibly at the expense of computational efficiency) by rerunning it on the entire sample after each arrival of a new example. Hence, our results of Section II can be used to obtain data-dependent risk bounds, expressed in terms of the sample statistic M_n, for any learning algorithm. In this respect, the advantage of genuinely on-line algorithms is the existence of accurate bounds on M_n, derived by exploiting the structure of the on-line learning process. No such bounds are known for batch algorithms.

All the results we have presented have convergence rates of the form $1/\sqrt{n}$. One might wonder whether it is possible, in our framework, to achieve rates $1/n$ when M_n happens to be small. Indeed, a more careful use of large-deviation tools (along with a more involved analysis) allows to sharpen the results we have given here. As a simple example, the bound of Corollary 2 could be replaced by

$$\mathbb{P}\left(\mathtt{risk}(\overline{H}) \geq M_n + \frac{c_1}{n}\ln\frac{n}{\delta} + c_2\sqrt{\frac{M_n}{n}\ln\frac{n}{\delta}}\right) \leq \delta.$$

This bound has asymptotical rate $1/n$ whenever the cumulative loss $nM_n = \sum_{t=1}^{n} \ell(H_{t-1}(X_t), Y_t)$ of the underlying on-line algorithm remains bounded. Though such sharper bounds are very interesting in their own right, we believe they do not add much to the overall approach of this paper in terms of simplicity and conciseness; for this reason, these bounds will be investigated in a future paper.

We close by mentioning an important open question: Is it possible to extend the results of Section II to the case when the examples are not independent; e.g., when they are generated by a stationary process?

Appendix
Proof of Lemma 3

Let

$$T^* = \operatorname*{argmin}_{0 \leq t < n} \left(\mathtt{risk}(H_t) + 2c_\delta(n-t)\right)$$

and let $H^* = H_{T^*}$. Set, for brevity

$$R_t = \mathtt{risk}_{\mathrm{emp}}(H_t, t+1)$$

$$R^* = \mathtt{risk}_{\mathrm{emp}}(H_{T^*}, T^*+1),$$

and let

$$\widehat{T} = \operatorname*{argmin}_{0 \leq t < n} \left(R_t + 2c_\delta(n-t)\right).$$

Note that $\widehat{H}$ defined in (3) coincides with $H_{\widehat{T}}$. With this notation, and since $R_{\widehat{T}} + c_\delta(n - \widehat{T}) \leq R^* + c_\delta(n - T^*)$ holds with certainty, we can write

$$\begin{aligned}
&\left(\mathtt{risk}(\widehat{H}) > \mathtt{risk}(H^*) + \mathcal{E}\right)\\
&\quad= \mathbb{P}\Big(\mathtt{risk}(H_{\widehat{T}}) > \mathtt{risk}(H^*) + \mathcal{E},\\
&\qquad\qquad R_{\widehat{T}} + c_\delta(n-\widehat{T}) \leq R^* + c_\delta(n-T^*)\Big)\\
&\quad\leq \sum_{t=0}^{n-1} \mathbb{P}\Big(R_t + c_\delta(n-t) \leq R^* + c_\delta(n-T^*),\\
&\qquad\qquad \mathtt{risk}(H_t) > \mathtt{risk}(H^*) + \mathcal{E}\Big)
\end{aligned} \tag{9}$$

where $\mathcal{E}$ is a positive-valued random variable whose value will be specified later. Now, if

$$R_t + c_\delta(n-t) \leq R^* + c_\delta(n-T^*)$$

holds, then at least one of the following three conditions:

$$\begin{aligned}
&R_t \leq \mathtt{risk}(H_t) - c_\delta(n-t)\\
&R^* > \mathtt{risk}(H^*) + c_\delta(n-T^*)\\
&\mathtt{risk}(H_t) - \mathtt{risk}(H^*) < 2c_\delta(n-T^*)
\end{aligned}$$

must hold. Hence, for any fixed t we can write

$$\begin{aligned}
&\mathbb{P}\Big(R_t + c_\delta(n-t) \leq R^* + c_\delta(n-T^*),\\
&\qquad \mathtt{risk}(H_t) > \mathtt{risk}(H^*) + \mathcal{E}\Big)\\
&\quad\leq \mathbb{P}\Big(R_t \leq \mathtt{risk}(H_t) - c_\delta(n-t)\Big) && (10)\\
&\qquad + \mathbb{P}\Big(R^* > \mathtt{risk}(H^*) + c_\delta(n-T^*)\Big) && (11)\\
&\qquad + \mathbb{P}\Big(\mathtt{risk}(H_t) - \mathtt{risk}(H^*) < 2c_\delta(n-T^*),\\
&\qquad\qquad \mathtt{risk}(H_t) > \mathtt{risk}(H^*) + \mathcal{E}\Big). && (12)
\end{aligned}$$

Probability (12) is zero if $\mathcal{E} = 2c_\delta(n-T^*)$. Hence, plugging (10) and (11) into (9) we can write

$$\begin{aligned}
&\mathbb{P}\Big(\mathtt{risk}(\widehat{H}) > \mathtt{risk}(H^*) + 2c_\delta(n-T^*)\Big)\\
&\quad\leq \sum_{t=0}^{n-1} \mathbb{P}\Big(R_t \leq \mathtt{risk}(H_t) - c_\delta(n-t)\Big)\\
&\qquad + n\mathbb{P}\Big(R^* > \mathtt{risk}(H^*) + c_\delta(n-T^*)\Big)\\
&\quad\leq \frac{\delta}{n+1} + n\sum_{t=0}^{n-1} \mathbb{P}\Big(R_t \geq \mathtt{risk}(H_t) + c_\delta(n-t)\Big)\\
&\quad\leq \frac{\delta}{n+1} + \frac{\delta n}{n+1} = \delta
\end{aligned}$$

where in the last two inequalities we applied Chernoff–Hoeffding bounds (see, e.g., [1, Ch. 8]) to the random variables R_t with mean $\texttt{risk}(H_t)$. □

Acknowledgment

The authors wish to thank the anonymous reviewers, and also the Associate Editor, for their many useful comments.

References

[1] L. Devroye, L. Győrfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*. Berlin, Germany: Springer-Verlag, 1996.

[2] V. Vapnik and A. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," *Theory Probab. Its Applic.*, vol. 16, no. 2, pp. 264–280, 1971.

[3] V. Vapnik, "Inductive principles of the search for empirical dependencies," in *Proc. 2nd Annu. Workshop on Computational Learning Theory*, 1989, pp. 3–21.

[4] ——, *The Nature of Statistical Learning Theory*, 2nd ed. Berlin, Germany: Springer-Verlag, 1999.

[5] P. Long, "The complexity of learning according to two models of a drifting environment," *Machine Learning*, vol. 37, no. 3, pp. 337–354, 1999.

[6] P. Bartlett, "The sample complexity of pattern classification with neural networks," *IEEE Trans. Inform. Theory*, vol. 44, pp. 525–536, Mar. 1998.

[7] S. Boucheron, G. Lugosi, and P. Massart, "A sharp concentration inequality with applications," *Random Structures and Algorithms*, vol. 16, pp. 277–292, 2000.

[8] R. Williamson, J. Shawe-Taylor, B. Schölkopf, and A. Smola, "Sample based generalization bounds," NeuroCOLT, Tech. Rep. NC-TR-99-055, 1999.

[9] P. Bartlett and S. Mendelson, "Rademacher and Gaussian complexities: Risk bounds and structural results," *J. Machine Learning Res.*, vol. 3, pp. 463–482, 2002.

[10] P. Bartlett, S. Boucheron, and G. Lugosi, "Model selection and error estimation," *Machine Learning*, vol. 48, pp. 85–113, 2001.

[11] A. Antos, B. Kégl, T. Linder, and G. Lugosi, "Data-dependent margin-based generalization bounds for classification," *J. Machine Learning Res.*, vol. 3, pp. 73–98, 2002.

[12] V. Koltchinskii and D. Panchenko, "Empirical margin distributions and bounding the generalization error of combined classifiers," *Ann. Statist.*, vol. 30, no. 1, pp. 1–50, 2002.

[13] J. Langford, M. Seeger, and N. Megiddo, "An improved predictive accuracy bound for averaging classifiers," in *Proc. 18th Int. Conf. Machine Learning*, 2001, pp. 290–297.

[14] R. Schapire, Y. Freund, P. Bartlett, and W. Lee, "Boosting the margin: A new explanation for the effectiveness of voting methods," *Ann. Statist.*, vol. 26, no. 5, pp. 1651–1686, 1998.

[15] R. Meir and T. Zhang, "Generalization error bounds for Bayesian mixture algorithms," *J. Machine Learning Res.*, vol. 4, pp. 839–860, 2003.

[16] A. Blum and J. Langford, "Microchoice bounds and self bounding learning algorithms," *Machine Learning*, vol. 51, no. 2, pp. 165–179, 2003.

[17] Y. Freund, "Self bounding learning algorithms," in *Proc. 11th Annu. Conf. Computational Learning Theory*, 1998, pp. 127–135.

[18] O. Bousquet and A. Elisseff, "Stability and generalization," *J. Machine Learning Res.*, vol. 2, pp. 499–526, 2002.

[19] R. Herbrich and R. Williamson, "Algorithmic luckiness," *J. Machine Learning Res.*, vol. 3, pp. 175–212, 2002.

[20] N. Cesa-Bianchi, Y. Freund, D. Helmbold, D. Haussler, R. Schapire, and M. Warmuth, "How to use expert advice," *J. Assoc. Comput. Mach.*, vol. 44, no. 3, pp. 427–485, 1997.

[21] M. Feder, N. Merhav, and M. Gutman, "Universal prediction of individual sequences," *IEEE Trans. Inform. Theory*, vol. 38, pp. 1258–1270, July 1992.

[22] D. Haussler, J. Kivinen, and M. Warmuth, "Sequential prediction of individual sequences under general loss functions," *IEEE Trans. Inform. Theory*, vol. 44, pp. 1906–1925, Sept. 1998.

[23] N. Littlestone and M. Warmuth, "The weighted majority algorithm," *Inform. Comput.*, vol. 108, pp. 212–261, 1994.

[24] V. Vovk, "A game of prediction with expert advice," *J. Comp. Syst. Sci.*, vol. 56, no. 2, pp. 153–173, 1998.

[25] P. Auer and M. Warmuth, "Tracking the best disjunction," *Machine Learning*, vol. 32, no. 2, pp. 127–150, 1998.

[26] N. Cesa-Bianchi, A. Conconi, and C. Gentile, "A second-order perceptron algorithm," in *Proc. 5th Annu. Conf. Computational Learning Theory (Lecture Notes in Artificial Intelligence).* Berlin, Germany, 2002, vol. 2375, pp. 121–137.

[27] Y. Freund and R. Schapire, "Large margin classification using the perceptron algorithm," *Machine Learning*, pp. 277–296, 1999.

[28] C. Gentile and M. Warmuth, "Linear hinge loss and average margin," in *Advances in Neural Information Processing Systems 10.* Cambridge, MA: MIT Press, 1999, pp. 225–231.

[29] C. Gentile, "The robustness of the p-norm algorithms," *Machine Learning*, vol. 53, no. 3, pp. 265–299, 2003.

[30] A. Grove, N. Littlestone, and D. Schuurmans, "General convergence results for linear discriminant updates," *Machine Learning*, vol. 43, no. 3, pp. 173–210, 2001.

[31] N. Littlestone, "Redundant noisy attributes, attribute errors, and linear threshold learning using winnow," in *Proc. 4th Annu. Workshop on Computational Learning Theory*, 1991, pp. 147–156.

[32] K. Azuma, "Weighted sums of certain dependent random variables," *Tohoku Math. J.*, vol. 68, pp. 357–367, 1967.

[33] A. Blum, A. Kalai, and J. Langford, "Beating the hold-out: Bounds for k-fold and progressive cross-validation," in *Proc. 12th Annu. Conf. Computational Learning Theory*, 1999, pp. 203–208.

[34] N. Littlestone, "From on-line to batch learning," in *Proc. 2nd Annu. Workshop on Computational Learning Theory*, 1989, pp. 269–284.

[35] H. Block, "The perceptron: A model for brain functioning," *Rev. Mod. Phys.*, vol. 34, pp. 123–135, 1962.

[36] A. Novikoff, "On convergence proofs of perceptrons," in *Proc. Symp. Mathematical Theory of Automata*, vol. XII, 1962, pp. 615–622.

[37] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Rev.*, vol. 65, pp. 386–408, 1958.

[38] M. Aizerman, E. Braverman, and L. Rozonoer, "Theoretical foundations of the potential function method in pattern recognition learning, automation and remote control," *Automat. Remote Contr.*, vol. 25, pp. 821–837, 1964.

[39] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge, U.K.: Cambridge Univ. Press, 2001.

[40] B. Schölkopf and A. Smola, *Learning With Kernels*. Cambridge, MA: MIT Press, 2002.