TAIL RISK BOUNDS FOR ON-LINE ALGORITHMS

Nicolò Cesa-Bianchi
Università di Milano, Italy

Joint work with: Claudio Gentile

STATISTICAL LEARNING THEORY

Examples (X_t, Y_t) are i.i.d. according to fixed and unknown probability distribution on $\mathcal{X} \times \mathcal{Y}$

Learning algorithm

$$(X_1, Y_1), \ldots, (X_n, Y_n) \quad \longrightarrow \boxed{\quad A \quad} \longrightarrow \quad \widehat{H} : \mathcal{X} \to \mathcal{D}$$

Loss $\qquad \ell : \mathcal{D} \times \mathcal{Y} \to \mathbb{R} \qquad$ ($\mathcal{D}$ = decision space)

Risk $\qquad \mathtt{risk}(H) = \mathbb{E}\, \ell(H(X), Y)$

Empirical risk $\qquad \mathtt{risk}_{\mathtt{emp}}(H) = \dfrac{1}{n} \displaystyle\sum_{t=1}^{n} \ell(H(X_t), Y_t)$

EXAMPLES

Regression with square loss:

$$\mathcal{Y} = \mathcal{D} = \mathbb{R} \qquad \ell(H(x), y) = (H(x) - y)^2$$

Binary classification:

$$\mathcal{Y} = \mathcal{D} = \{-1, 1\} \qquad \ell(H(x), y) = \mathbb{I}_{\{H(x) \neq y\}}$$

Classification with absolute loss:

$$\mathcal{Y} = \{-1, 1\} \qquad \mathcal{D} = [-1, 1] \qquad \ell(H(x), y) = |H(x) - y|$$

RISK BOUNDS

$$(X_1, Y_1), \ldots, (X_n, Y_n) \quad \longrightarrow \boxed{\quad A \quad} \longrightarrow \quad \widehat{H} : \mathcal{X} \to \mathcal{D}$$

$\widehat{H}$ is (random) hypothesis output by learner

Goal: Show that $\texttt{risk}(\widehat{H})$ is small with high probability

We assume bounded loss functions

DATA-DEPENDENT VC THEORY

$\mathcal{H}$ = set of functions $H : \mathcal{X} \to \mathcal{D}$ from which $\widehat{H}$ is selected

w.h.p. for all $h \in \mathcal{H}$

$$\texttt{risk}(h) \leq \texttt{risk}_{\texttt{emp}}(h) + c_1 \sqrt{\texttt{risk}_{\texttt{emp}}(h) \, \frac{V_{\mathcal{H}} \ln n}{n}} + c_2 \frac{V_{\mathcal{H}} \ln n}{n}$$

where n is sample size

VC theory of generalization studies properties of $\mathcal{H}$

We study a small subclass of $\mathcal{H}$ generated by the interaction between a learner and the training data

AN ALGORITHM-DEPENDENT THEORY

$$
\begin{array}{rcccl}
\langle \quad \rangle & \longrightarrow & \boxed{\quad A \quad} & \longrightarrow & H_0 \\
\langle (X_1, Y_1) \rangle & \longrightarrow & \boxed{\quad A \quad} & \longrightarrow & H_1 \\
\langle (X_1, Y_1), (X_2, Y_2) \rangle & \longrightarrow & \boxed{\quad A \quad} & \longrightarrow & H_2 \\
 & & \vdots & & \\
\langle (X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n) \rangle & \longrightarrow & \boxed{\quad A \quad} & \longrightarrow & H_n
\end{array}
$$

The ensemble of functions generated by A:

$$H_0, H_1, \ldots, H_{n-1}, H_n$$

GOALS

1. Bound the average risk of the ensemble in terms of the incremental performance of the algorithm on the data.

2. Find an element of the ensemble whose risk is close to the ensemble average

3. Relate to optimal risk in $\mathcal{H}$

STEP 1: BOUND THE AVERAGE RISK

$\texttt{risk}(H_{t-1}) - \ell(H_{t-1}(X_t), Y_t)$ is a martingale difference sequence

$$\mathbb{E}\Big[\texttt{risk}(H_{t-1}) - \ell(H_{t-1}(X_t), Y_t) \,\Big|\, (X_1, Y_1), \ldots, (X_{t-1}, Y_{t-1})\Big] = 0$$

Associated martingale

$$\sum_{t=1}^{n} \Big(\texttt{risk}(H_{t-1}) - \ell(H_{t-1}(X_t), Y_t) \Big)$$

$$\iff \quad \underbrace{\frac{1}{n}\sum_{t=1}^{n} \texttt{risk}(H_{t-1})}_{\text{average risk}} - \underbrace{\frac{1}{n}\sum_{t=1}^{n} \ell(H_{t-1}(X_t), Y_t)}_{\text{on-line statistic}}$$

BERNSTEIN'S BOUND

If $Z_1, Z_2, \ldots$ is a martingale difference sequence with increments bounded by 1 and

$$V_n = \sum_{t=1}^{n} \mathbb{E}\left[Z_t^2 \mid Z_1, \ldots, Z_{t-1}\right]$$

then for all $S, K > 0$

$$\boxed{\mathbb{P}\left(\sum_{t=1}^{n} Z_n \geq S, \quad V_n \leq K\right) \leq \exp\left(-\frac{S^2}{2(S/3+K)}\right)}$$

APPLICATION OF BERNSTEIN'S BOUND

Since $0 \leq \ell \leq 1$,

$$\begin{aligned} &\text{VAR}\left[\ell(H_{t-1}(X_t), Y_t) \,\middle|\, (X_1, Y_1), \ldots, (X_{t-1}, Y_{t-1})\right] \\ &\leq \quad \mathbb{E}\left[\texttt{risk}(H_{t-1}) \,\middle|\, (X_1, Y_1), \ldots, (X_{t-1}, Y_{t-1})\right] \end{aligned}$$

$$\boxed{\frac{1}{n}\sum_{t=1}^{n} \texttt{risk}(H_{t-1}) \quad \leq \quad \frac{M_n}{n} + \frac{c}{n}\left(\ln M_n + \sqrt{M_n \ln M_n}\right) \quad \text{w.h.p.}}$$

note that $\Omega(1) = M_n = O(n)$

$$\frac{M_n}{n} = \frac{1}{n}\sum_{t=1}^{n} \ell(H_{t-1}(X_t), Y_t) \qquad \text{is the on-line statistic}$$

STEP 2: PICK A GOOD FUNCTION IN THE ENSEMBLE

$H_0, H_1, \ldots, H_n$ ensemble of functions

1. test each H_t on $(X_{t+1}, Y_{t+1}), \ldots, (X_n, Y_n)$
2. pick $\widehat{H} = H_{t^*}$ minimizing a penalized risk estimate

$$\boxed{\quad \texttt{risk}(\widehat{H}) \quad \leq \quad \frac{M_n}{n} + \frac{c}{n}\Big((\ln n)^2 + \sqrt{M_n \ln n}\Big) \qquad \text{w.h.p.} \quad}$$

STEP 3: RELATE TO OPTIMAL RISK IN $\mathcal{H}$

So far, no assumption on learner A

Via specific analyses, we can relate M_n/n to best risk in $\mathcal{H}$

Example: Vovk's aggregating forecaster for regression with square loss ($\mathcal{H} =$ RKHS)

$$h_n^* = \operatorname*{argmin}_{h \in \mathcal{H}} \left(\texttt{risk}(h) + \frac{\|h\|^2}{n} \right)$$

$$\frac{M_n}{n} \le \texttt{risk}(h_n^*) + \frac{\|h_n^*\|^2}{n} + c_1 \sqrt{\frac{\texttt{risk}(h_n^*)}{n}} + c_2 \frac{Y^2}{n} \sum_i \ln\big(1 + \lambda_i\big)$$

w.h.p. where $\lambda_1, \lambda_2, \ldots$ are the eigenvalues of the kernel matrix

MORE EXAMPLES

Pointwise bounds for sequences $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n) \in \mathbb{R}^d \times \{-1, +1\}$ that are linearly separable in a given RKHS $\mathcal{H}$

Kernel Perceptron

$$M_n \leq \frac{1}{\gamma} \sqrt{\sum_i \lambda_i}$$

Kernel 2nd order Perceptron

$$M_n \leq \frac{1}{\gamma} \sqrt{\left(1 + \sum_i f(\boldsymbol{x}_i)^2\right) \sum_i \ln(1 + \lambda_i)}$$

$f \in \mathcal{H}$ is a linear separator with margin γ

$\lambda_1, \lambda_2, \ldots$ are the eigenvalues of the kernel matrix

All sums are on mistaken examples

CONCLUSIONS

- Algorithm-based vs. $\mathcal{H}$-based approach to analysis of risk
- Data-dependent bounds for any learner in terms of on-line statistic
- Bounds on $\inf_{h\in\mathcal{H}} \mathtt{risk}(h)$ for specific learners
- Fast rates without fancy statistical tools

EXPERIMENTS ON RCV1 CORPUS

Documents chronologically ordered

Only 50 most frequent categories

Training set: blocks of increasing size (from 5K to 80K)

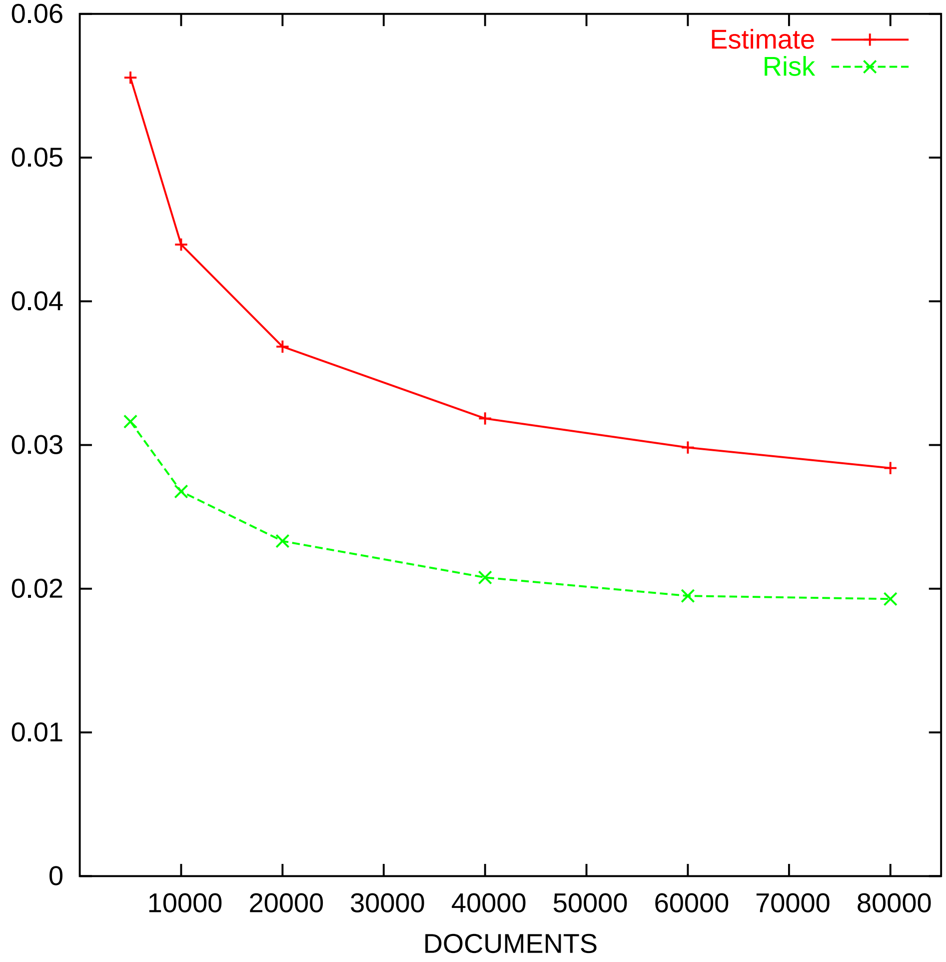
Test set: always a 20K block

Bound on test error:

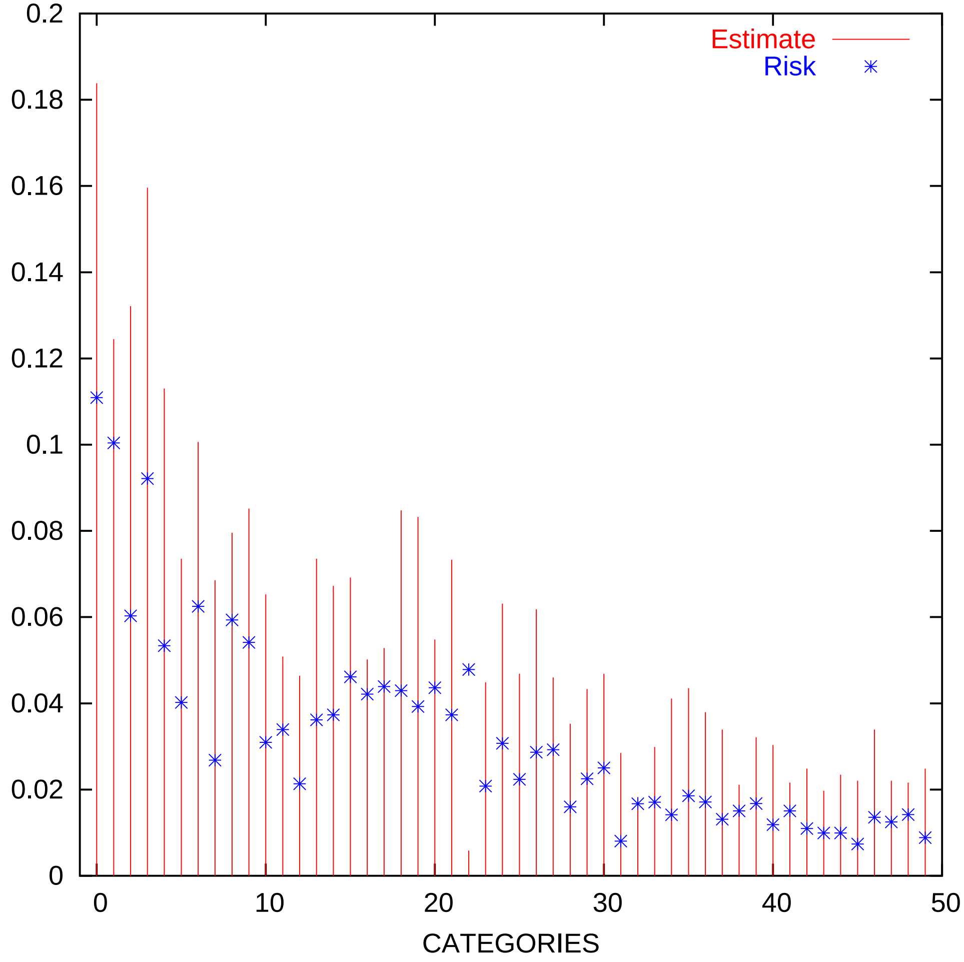
$$M_n + \frac{1}{n}\left(c_1 \ln \frac{M_n}{\delta} + c_2 \sqrt{M_n \ln \frac{M_n}{\delta}}\right)$$

We ran a voted Perceptron with a linear kernel

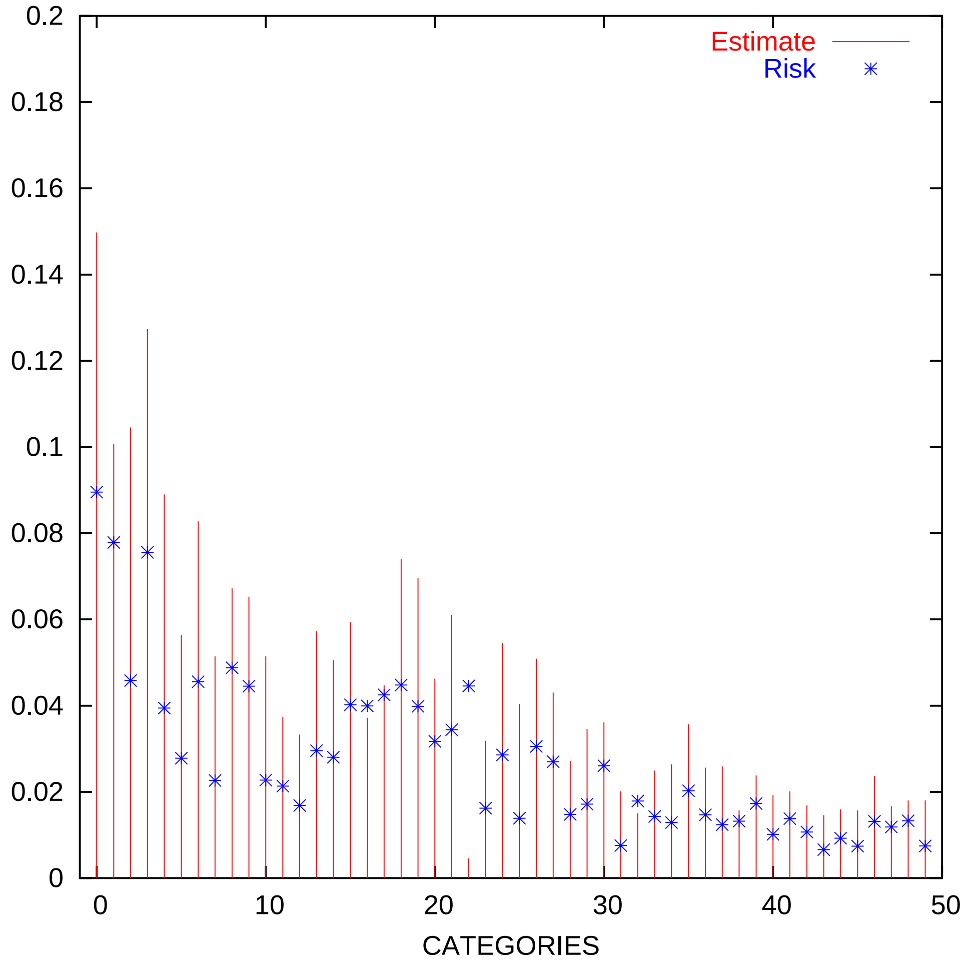
AVERAGES OVER ALL CATEGORIES



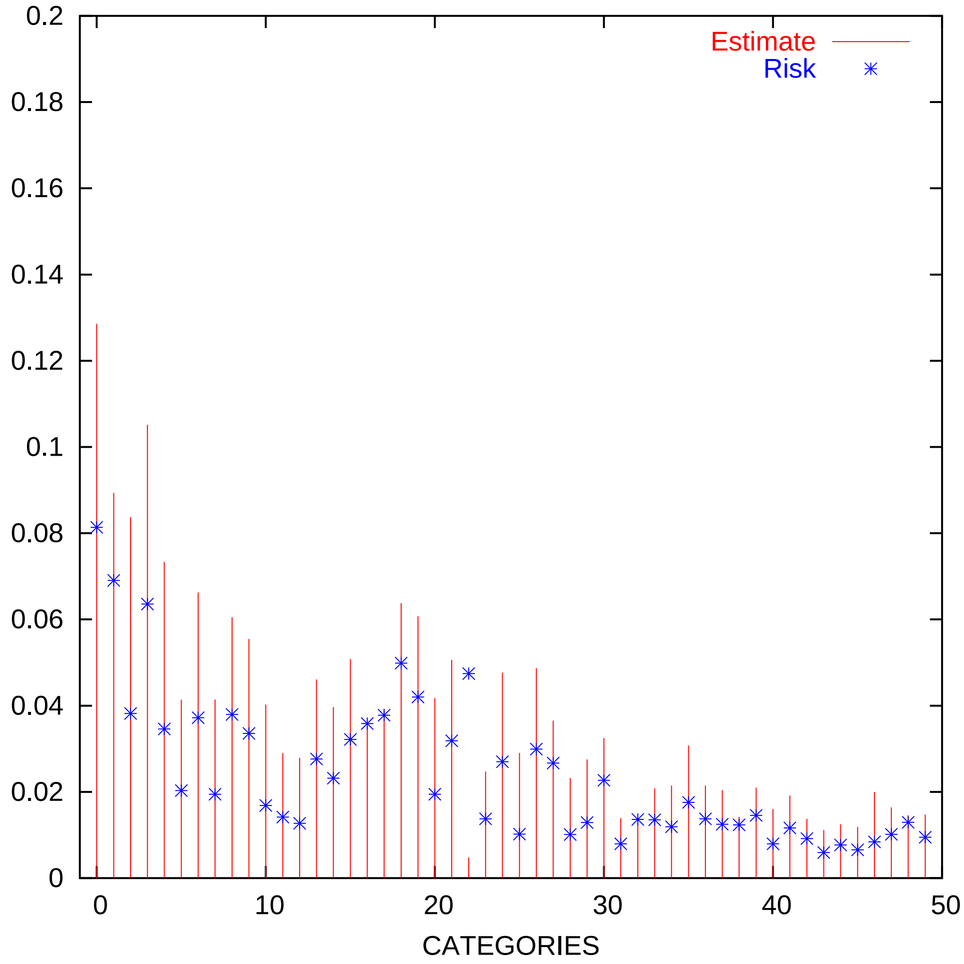
ESTIMATES AFTER 5K DOCUMENTS



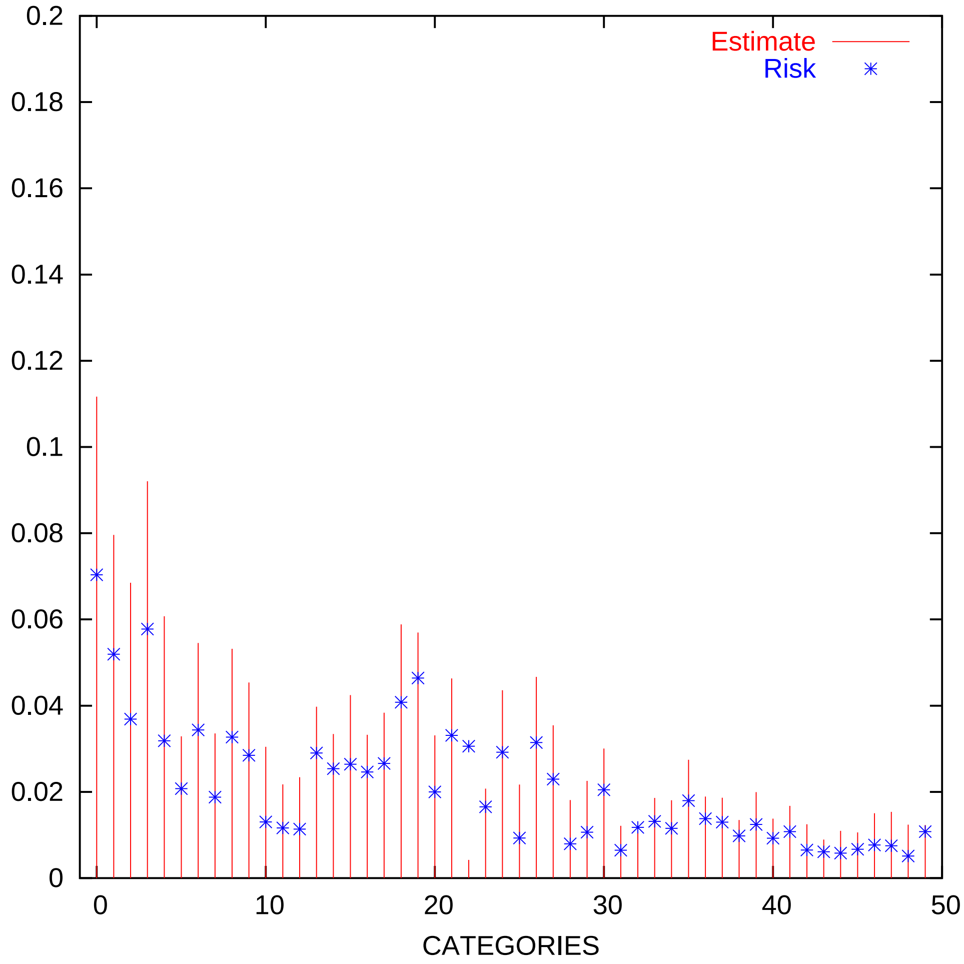
ESTIMATES AFTER 10K DOCUMENTS



ESTIMATES AFTER 20K DOCUMENTS



ESTIMATES AFTER 40K DOCUMENTS



ESTIMATES AFTER 80K DOCUMENTS