Machine Learning — Statistical Methods for Machine Learning

Consistency and nonparametric algorithms

Instructor: *Nicolò Cesa-Bianchi* version of April 16, 2024

Consistency is an asymptotical property certifying that the risk of the predictors generated by a learning algorithm converges to the Bayes risk in expectation as the size of the training set increases. Recall that $A(S_m)$ is the predictor generated by a learning algorithm A on a training set S_m of size m. A learning algorithm A is **consistent** with respect to a loss function ℓ if for any data distribution $\mathcal{D}$ it holds that

$$\lim_{m\to\infty} \mathbb{E}\Big[\ell_{\mathcal{D}}\big(A(S_m)\big)\Big] = \ell_{\mathcal{D}}(f^*)$$

where the expectation is with respect to the random draw of the training set S_m of size m from the distribution $\mathcal{D}$, and $\ell_{\mathcal{D}}(f^*)$ is the Bayes risk for $(\mathcal{D}, \ell)$. In some cases, we may define consistency with respect to a restricted class of distributions $\mathcal{D}$. For example, in binary classification we may restrict to all distributions $\mathcal{D}$ such that $\eta(\boldsymbol{x}) = \mathbb{P}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})$ is a Lipschitz function on $\mathcal{X}$. Formally, there exists $0 < c < \infty$ such that

$$\big|\eta(\boldsymbol{x}) - \eta(\boldsymbol{x}')\big| \le c\,\big\|\boldsymbol{x} - \boldsymbol{x}'\big\| \qquad \text{for all } \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X} \ .$$

Technically, this conditions implies that η is Lipschitz in $\mathcal{X}$. This is a restriction on the set of all allowed η as $c < \infty$ implies continuity (but the opposite is not true).

Nonparametric algorithms. Given a learning algorithm A, let $\mathcal{H}_m$ be the set of predictors generated by A on training sets of size m: $h \in \mathcal{H}_m$ if and only if there exists a training set S_m of size m such that $A(S_m) = h$. We say that A is a nonparametric learning algorithm if A's approximation error vanishes as m grows to infinity. Formally,

$$\lim_{m\to\infty} \min_{h\in\mathcal{H}_m} \ell_{\mathcal{D}}(h) = \ell_{\mathcal{D}}(f^*) \ .$$

Two notable examples of nonparametric learning algorithms are k-NN and the greedy algorithm for decision tree classifiers (i.e., the algorithm that always chooses to split a leaf that maximizes the decrease in training error). Nonparametric algorithms are recognizable because:

- the size (memory footprint) of their predictors tends to grow with the training set size
- for all sufficiently large values of m, there are predictors in $\mathcal{H}_m$ with training error arbitrarily close to zero.

The standard k-NN algorithm is nonparametric but not known to be consistent for any fixed value of k. Indeed, one can only show that

$$\lim_{m\to\infty} \mathbb{E}\Big[\ell_{\mathcal{D}}\big(k\text{-NN}(S_m)\big)\Big] \le \ell_{\mathcal{D}}(f^*) + 2\sqrt{\frac{\ell_{\mathcal{D}}(f^*)}{k}} \tag{1}$$

for any data distribution $\mathcal{D}$. However, if we let k be chosen as a function k_m of the training set size, then the algorithm becomes consistent provided $k_m \to \infty$ and $k_m = o(m)$.

Similarly, the greedy algorithm for building tree classifiers is consistent (for $\mathcal{X} \equiv \mathbb{R}^d$) whenever the two following conditions are fulfilled: for any leaf ℓ, let $\mathcal{X}_\ell = \big\{\boldsymbol{x} \in \mathbb{R}^d : \boldsymbol{x} \text{ is routed to } \ell\big\}$ and let N_ℓ be the number of training examples routed to ℓ. Then, as $m \to \infty$, to guarantee consistency we must have that the diameter of $\mathcal{X}_\ell$ goes to zero and $N_\ell \to \infty$ for almost all leaves ℓ. In other words, the tree must grow unboundedly but not too fast.

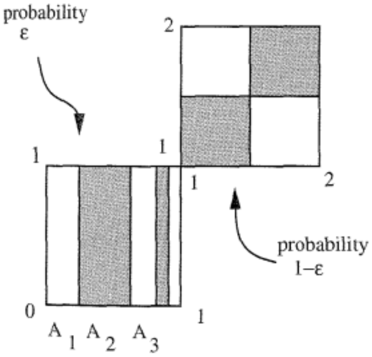

Figure 1: An example of a nonparametric algorithm that is not consistent: splits are always made in the leftmost square. Figure from: A Probabilistic Theory of Pattern Recognition, by Luc Devroye, László Györfi, and Gábor Lugosi. Springer, 1996.

It is possible to construct examples of nonparametric algorithms that are not consistent. We consider one such example: the greedy algorithm for binary tree classifiers that uses the true risk as splitting criterion. In other words, instead of splitting the leaf that causes the largest drop in the training error, we assume the algorithm is allowed to choose the leaf that causes the largest drop in the risk for the zero-one loss. For any $\varepsilon > 0$, consider $\mathcal{X} = [0,1]^2 \cup [1,2]^2$ with uniform distribution ε on $[0,1]^2$ and $1-\varepsilon$ on $[1,2]$. To define the Bayes optimal classifier f^*, introduce

$$\begin{aligned}
A_1 &= \big[0, 1/4\big) \\
A_2 &= \big[1/4, 1/4 + 3/8\big) \\
A_3 &= \big[1/4 + 3/8, 1/4 + 3/8 + 3/16\big) \\
&\;\vdots \\
A_k &= \big[1/4 + 3/8 + \cdots + 3/2^k, 1/4 + 3/8 + \cdots + 3/2^{k+1}\big)
\end{aligned}$$

Now let $f^*(x) = 1$ if and only if $x \in [1, 3/2]^2 \cup [3/2, 2]^2 \cup \big(A_2 \cup A_4 \cup A_6 \cdots\big) \times [0,1]$ and assume $\mathbb{P}(Y = 1 \mid X = x) = f^*(x)$ so that the Bayes risk is zero, see Figure 1. Hence,

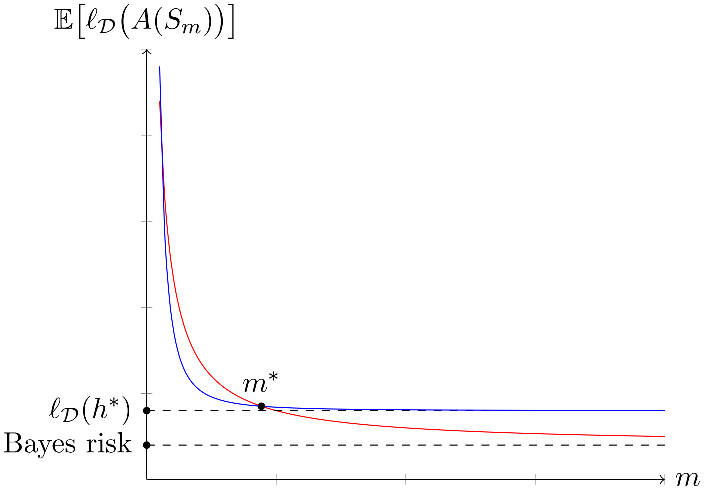

Figure 2: Typical behavior of expected risk $\mathbb{E}\big[\ell_{\mathcal{D}}\big(A(S_m)\big)\big]$ as a function of training set size for a consistent algorithm (red line) and for a nonconsistent algorithm (blue line). For small training set sizes $m < m^*$, the nonconsistent algorithm has a better performance. (Thanks to Edoardo Marangoni for drawing the picture.)

$$\begin{aligned}
\mathbb{P}(Y=1) &= (1-\varepsilon)\mathbb{P}\big(X \in [1,3/2]^2\big) + (1-\varepsilon)\mathbb{P}\big(X \in [3/2,2]^2\big) + \varepsilon \sum_{k=1}^{\infty} \mathbb{P}\big(X \in A_{2k}\big) \\
&= (1-\varepsilon)\left(\frac{1}{4}+\frac{1}{4}\right) + \varepsilon \sum_{k=1}^{\infty} \frac{3}{2^{2k+1}} \\
&= \frac{1-\varepsilon}{2} + \frac{3\varepsilon}{2} \sum_{k=1}^{\infty} \frac{1}{4^k} \\
&= \frac{1-\varepsilon}{2} + \frac{3\varepsilon}{2} \times \frac{1}{3} = \frac{1}{2}
\end{aligned}$$

So, initially the tree has a sigle leaf with risk equal to $\mathbb{P}(Y=1) = \frac{1}{2}$. Any split along x_2 does not decrease the risk. Any split of the form $x_1 \geq a$ with $a \in [1,2]$ does not decrease the risk. Hence the greedy algorithm will choose the split $x_1 \leq \frac{1}{4}$ creating a pure leaf corresponding to A_1 and decreasing the risk by $\varepsilon \times \mathbb{P}(A_1) = \frac{\varepsilon}{4}$. The second greedy split is $x_1 \leq \frac{1}{4} + \frac{3}{8}$ creading a pure leaf corresponding to A_2 and decreasing the risk by $\frac{\varepsilon}{8}$. After k such greedy splits, the risk of the tree is

$$\varepsilon\left(\frac{1}{2} - \sum_{i=1}^{k} \frac{1}{2^{k+1}}\right) + \frac{1-\varepsilon}{2}$$

Hence, after any number of splits the risk of the tree is always larger than $\frac{1-\varepsilon}{2}$.

In practice, a nonconsistent algorithm may be preferred over a consistent one, see Figure 2. This is due to the fact that the rate of convergence to the Bayes risk of a consistent algorithm can be arbitrarily slow, as shown by the following result.

Theorem 1 (No Free Lunch)**.** *For any sequence $a_1, a_2, \dots$ of positive numbers converging to zero and such that $\frac{1}{16} \ge a_1 \ge a_2 \ge \cdots > 0$ and for any consistent learning algorithm A for binary classification with zero-one loss, there exists a data distribution $\mathcal{D}$ such that $\ell_{\mathcal{D}}(f^*) = 0$ and $\mathbb{E}\big[\ell_{\mathcal{D}}\big(A(S_m)\big)\big] \ge a_m$ for all $m \ge 1$.*

Theorem 1 does not prevent a consistent algorithm from converging fast to the Bayes risk for specific distributions $\mathcal{D}$. What the theorem shows is that if A converges to the Bayes risk for any data distribution, then it will converge arbitrarily slow for some of these distributions.

For binary classification, we can summarize the situation as follows.

- Under no assumption on η, there is no guaranteed convergence rate to Bayes risk.
- Under Lipschitz assumptions on η, the typical convergence rate to Bayes risk is $m^{-1/(d+1)}$.
- Under no assumptions on η, the typical convergence rate to the risk of the best predictor in a parametric (or finite) class $\mathcal{H}$ is $m^{-1/2}$, exponentially better than the nonparametric rate.

Note that the convergence rate $m^{-1/(d+1)}$ implies that to get ε-close to Bayes risk, we need a training set size m of order $\varepsilon^{-(d+1)}$. This exponential dependence on the number of features of the training set size is known as **curse of dimensionality** and refers to the difficulty of learning in a nonparametric setting when datapoints live in a high-dimensional space.