Machine Learning — Statistical Methods for Machine Learning

Consistency and nonparametric algorithms

Instructor: *Nicolò Cesa-Bianchi* version of April 29, 2022

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 η is a Lipschitz function.

Nonparametric algorithms. Given a learning algorithm A, let $\mathcal{H}_m$ be the set of predictors $A(S_m)$ obtained as S_m varies over all training sets of size m. Since f^* can be any predictor, if A is consistent then the set $\mathcal{H}_m$ must eventually (as m grows) include predictors whose risk is arbitrarily close to the Bayes risk. This is typically achieved by letting the complexity of the predictors output by A be a function of the training set size. Indeed, a characteristic signature of a nonparametric algorithm, like k-NN, is that the size (memory footprint) of its predictors grows with the size of the training data. Algorithms of this kind are called nonparametric because the predictors they generate cannot be described using a pre-determined number of variables. Algorithms that output tree predictors whose number of nodes does not have a pre-determined upper bound (but is free to grow with the size of the training set) are also nonparametric.

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

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

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. Let $\ell(\boldsymbol{x})$ be the leaf to which $\boldsymbol{x} \in \mathbb{R}^d$ is routed in the current tree and let $N_{\ell(\boldsymbol{x})}$ be the number of training examples routed to $\ell(\boldsymbol{x})$. Then, as $m \to \infty$, to guarantee consistency we must have that $N_{\ell(\boldsymbol{X})} \to \infty$ and the diameter of the leaf region $\big\{\boldsymbol{x} \in \mathbb{R}^d : \ell(\boldsymbol{X}) = \ell(\boldsymbol{x})\big\}$ goes to zero, where both conditions must hold in probability with respect to the random draw of $\boldsymbol{X}$. In other words, the tree must grow unboundedly but not too fast.

In practice, a consistent algorithm may not be preferred over a nonconsistent one, see Figure 1. This 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.

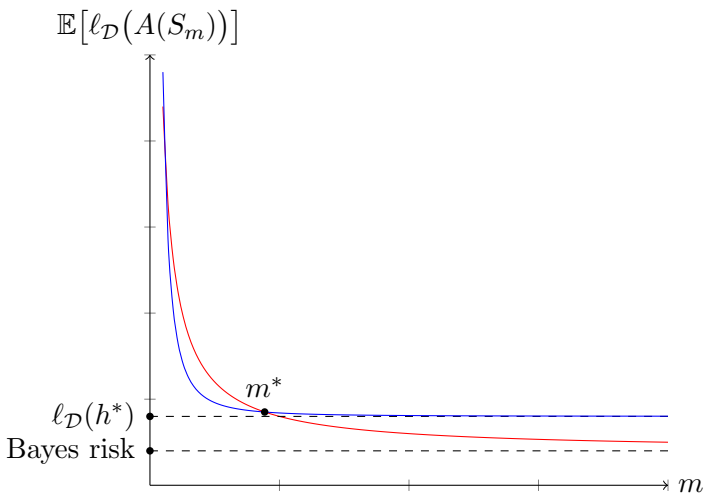

Figure 1: Typical behavior of expected risk $\mathbb{E}\big[\ell_{\mathcal{D}}\big(A(S_m)\big)\big]$ as a function of training set size for a consistent algorithm (red line) and for a nonconsistent algorithm (blue line). For small training set sizes $m < m^*$, the nonconsistent algorithm has a better performance. (Thanks to Edoardo Marangoni for drawing the picture.)

Theorem 1 (No Free Lunch)**.** *For any sequence $a_1, a_2, \ldots$ of positive numbers converging to zero and such that $\frac{1}{16} \geq a_1 \geq a_2 \geq \cdots$ 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 $\mathbb{E}\big[\ell_{\mathcal{D}}\big(A(S_m)\big)\big] \geq a_m$ for all $m \geq 1$.*

Note that $\ell_{\mathcal{D}}(f^*) = 0$ implies $\eta(\boldsymbol{x}) \in \{0, 1\}$ for each $\boldsymbol{x}$, where $\eta(\boldsymbol{x}) = \mathbb{P}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})$. This means that $\eta : \mathcal{X} \to [0, 1]$ is not a Lipschitz function. Also, 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 Lipschitz assumptions on η, the typical convergence rate to Bayes risk is $m^{-1/(d+1)}$ (exponentially slow in d).
- Under no assumption on η, there is no guaranteed convergence rate to Bayes risk.
- 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.