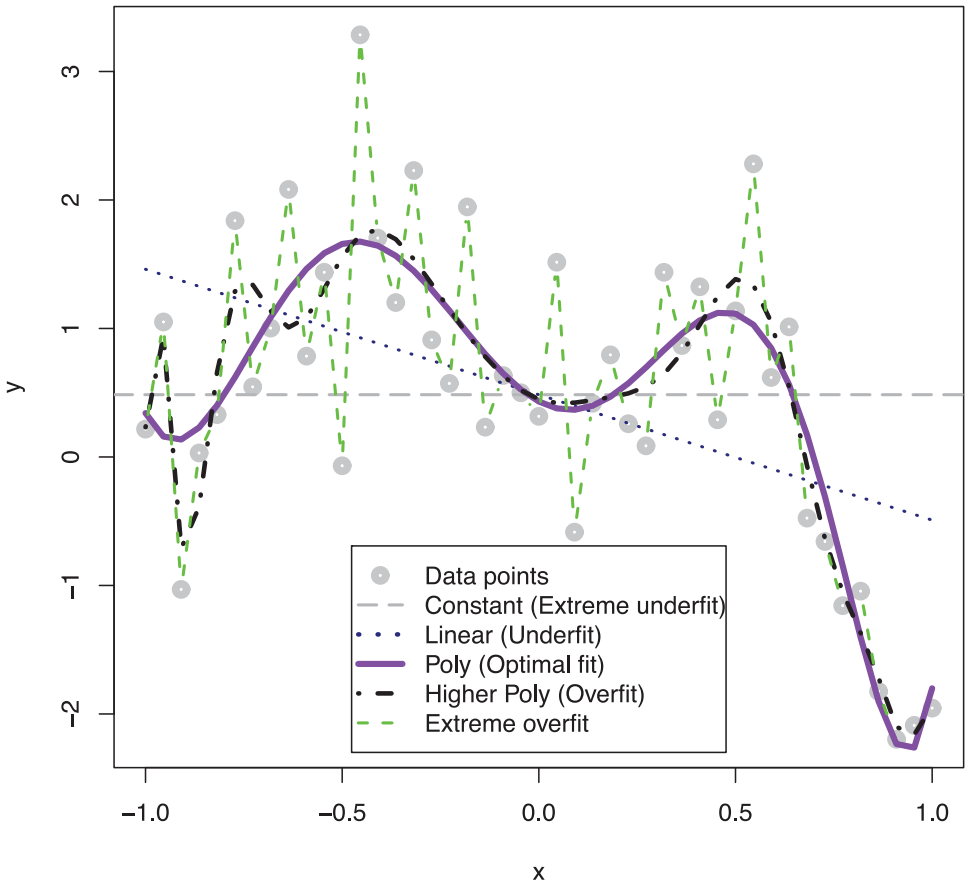
Model Selection for Optimal Prediction in Statistical Machine Learning



Ernest Fokoué

Introduction

At the core of all our modern-day advances in artificial intelligence is the emerging field of statistical machine learning (SML). From a very general perspective, SML can be thought of as a field of mathematical sciences that combines mathematics, probability, statistics, and computer science with several ideas from cognitive neuroscience and psychology to inspire the creation, invention, and discovery of abstract models that attempt to learn and extract patterns from the data. One could think of SML as a field of science dedicated to building models endowed with the ability to learn from the data in ways similar to the ways humans learn, with the ultimate goal of understanding and then mastering our complex world well enough to predict its unfolding as accurately as possible. One of the earliest applications of statistical machine learning centered around the now ubiquitous MNIST benchmark task, which consists of building statistical models (also

Ernest Fokoué is a professor of statistics at Rochester Institute of Technology. His email address is epfeqa@rit.edu.

Communicated by Notices *Associate Editor Emilie Purvine.*

DOI: https://doi.org/10.1090/noti2014

known as learning machines) that automatically learn and accurately recognize handwritten digits from the United States Postal Service (USPS). A typical deployment of an artificial intelligence solution to a real-life problem would have several components touching several aspects of the taxonomy of statistical machine learning. For instance, when artificial intelligence is used for the task of automated sorting of USPS letters, at least one component of the whole system deals with recognizing the recipient of a given letter as accurately as (or even better than) a human operator. *This would mean that the statistical machine learning model can ideally recognize handwritten digits regardless of the various ways in which those digits are written*. How does one go about formulating, defining, designing, building, refining, and ultimately deploying such statistical machine learning models for the intended use? In the case of the MNIST data, for instance, the digits to be potentially recognized are captured as a matrix, which is then transformed and represented as a high-dimensional vector fed as the input to a statistical learning model, along with the true label of the digit at hand. *Conceptually, the task of building the statistical learning machine is mathematically formulated as the construction of a function mapping the elements of the input space* (*space in which the digits are represented*) *to the output space* (*space of the true labels for the digits*). Over the years, different methods have been created and developed by statisticians and computer scientists from all around the world to help build statistical learning machines for a wide variety of problems like those mentioned earlier. F. Rosenblatt's [13] groundbreaking and thought-provoking publication of the seminal paper featuring the so-called *Perceptron* ushered in the era of brain-inspired statistical and computational learning, and can rightly be thought of as the catalyst of the field of artificial neural networks, and even arguably the ancestor of our modern-day hot topic of deep neural networks. A couple of decades after Rosenblatt's seminal paper, the Multilayer Perceptron (MLP) was introduced as one of the solutions to the limitations of the Perceptron. MLPs extended, strengthened, and empowered artificial neural networks by allowing potentially many hidden layers, a tremendous improvement over the Perceptron that brought a much needed new spring to artificial intelligence. MLPs turned out to be extremely successful, namely, on tasks like the MNIST USPS digit recognition task mentioned earlier, but also on several other tasks including credit scoring, stock price prediction, automatic translation, and medical diagnosis, just to name a few. MLPs triggered a veritable scientific revolution, inspiring the flourishing of creativity among researchers, many of whom invented or discovered entirely new learning methods and paradigms, or revived or adapted existing ones.

Theoretical Foundations

It is typical in statistical machine learning that a given problem will be solved in a wide variety of different ways. As a result, it is a central element in SML, both within each paradigm and among paradigms, to come up with good criteria for deciding and determining which learning machine or statistical model is the best for the given task at hand. To better explain this quintessential task of model selection, we consider a typical statistical machine learning setting, with two sets $\mathscr{X}$ and $\mathscr{Y}$, along with their Cartesian product $\mathscr{Z} \equiv \mathscr{X} \times \mathscr{Y}$. We further define $\mathscr{Z}^n \equiv \mathscr{Z} \times \mathscr{Z} \times \cdots \times \mathscr{Z}$ to be the n-fold Cartesian product of $\mathscr{Z}$. We assume that $\mathscr{Z}$ is equipped with a probability measure ψ, albeit assumed unknown throughout this paper. Let $\mathbf{Z} \in \mathscr{Z}^n$, with $\mathbf{Z} = ((\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n))$, denote a realization of a random sample of n examples, where each example $\mathbf{z}_i = (\mathbf{x}_i, y_i)$ is independently drawn according to the above probability measure ψ on the product space $\mathscr{Z} \equiv \mathscr{X} \times \mathscr{Y}$. For practical reasons, and in keeping with the data science and artificial intelligence lexicon, we shall quite often refer to the random sample $\mathbf{Z}$ as the *data set*, and will use the compact and comprehensive notation

$$\mathscr{D}_n = \{(\mathbf{x}_i, y_i) \overset{\text{iid}}{\sim} p_{\mathbf{xy}}(\mathbf{x}, y),\ i = 1, \dots, n\}, \tag{1}$$

where all pairs $(\mathbf{x}_i, y_i) \in \mathscr{X} \times \mathscr{Y}$, and $p_{\mathbf{xy}}(\mathbf{x}, y)$ is the probability density function associated with the probability measure ψ on $\mathscr{Z}$. *Given a random sample* $\mathbf{Z} = ((\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n))$, *one of the most pervading goals in both theoretical and applied statistical machine learning is to find the function* $f^\star : \mathscr{X} \to \mathscr{Y}$ *that best captures the dependencies between the* $\mathbf{x}_i$*'s and the* y_i*'s in such a way that, given a new random* (*unseen*) *observation* $\mathbf{z}^{\text{new}} = (\mathbf{x}^{\text{new}}, y^{\text{new}}) \sim p_{\mathbf{xy}}(\mathbf{x}, y)$ *with* $\mathbf{z}^{\text{new}} \notin \mathscr{D}_n$, *the image* $f^\star(\mathbf{x}^{\text{new}})$ *of* $\mathbf{x}^{\text{new}} \sim p_{\mathbf{x}}(\mathbf{x})$ *provides a prediction of* y^{new} *that is as accurate and precise as possible, in the sense of yielding the smallest possible discrepancy between* y^{new} *and* $f^\star(\mathbf{x}^{\text{new}})$.

This setting, where one seeks to build functions of the type $f : \mathscr{X} \to \mathscr{Y}$, is the foundational setting of machine learning in general and statistical machine learning in particular. Throughout this paper, we shall refer to $\mathscr{X}$ as the input space and to $\mathscr{Y}$ as the output space. For simplicity, we shall assume that $\mathscr{X} \subseteq \mathbb{R}^p$ for our methodological and theoretical derivations and explanations, but will allow $\mathscr{X}$ to be more general in practical demonstrations and examples. We will consider both regression learning corresponding to $\mathscr{Y} = \mathbb{R}$ and multicategory classification learning (pattern recognition) corresponding to output spaces of the form $\mathscr{Y} = \{1, 2, \dots, G\}$, where G is the number of categories.

Definition 1. A loss function $\mathcal{L}(\cdot, \cdot)$ is a nonnegative bivariate function $\mathcal{L} : \mathscr{Y} \times \mathscr{Y} \longrightarrow \mathbb{R}_+$, such that given $a, b \in \mathscr{Y}$, the value of $\mathcal{L}(a, b)$ measures the discrepancy between a

and b, or the deviance of a from b, or the loss incurred from using b in place of a. For instance, $\mathcal{L}(y, f(\mathbf{x})) = \mathcal{L}(f(\mathbf{x}), y)$ will be used throughout this paper to quantify the discrepancy between y and $f(\mathbf{x})$ with the finality of choosing the best f, the optimal f, the f that minimizes expected discrepancy over the entire $\mathscr{Z}$. The loss function plays a central role in statistical learning theory as it allows an unambiguous measure and quantification of optimality.

Definition 2. The theoretical risk or generalization error or true error of any function $f \in \mathscr{Y}^{\mathscr{X}}$ is given by

$$\begin{aligned} R(f) &= \mathbb{E}[\mathcal{L}(Y, f(X))] \\ &= \int_{\mathscr{X}\times\mathscr{Y}} \mathcal{L}(y, f(\mathbf{x}))\, p_{\mathbf{xy}}(\mathbf{x}, y)\, d\mathbf{x}dy \end{aligned} \tag{2}$$

and can be interpreted as the expected discrepancy between $f(X)$ and Y, and indeed as a measure of the predictive strength of f. Ideally, one seeks to find the minimizer $f^\star$ of $R(f)$ over all measurable functions $f \in \mathscr{Y}^{\mathscr{X}}$, specifically,

$$f^\star = \underset{f\in\mathscr{Y}^{\mathscr{X}}}{\arg\inf}\{R(f)\} = \underset{f\in\mathscr{Y}^{\mathscr{X}}}{\arg\inf}\Big\{\mathbb{E}[\mathcal{L}(Y, f(X))]\Big\}, \tag{3}$$

whose corresponding theoretical risk $R^\star$ serves as the gold standard and is given by

$$R^\star = R(f^\star) = \inf_{f\in\mathscr{Y}^{\mathscr{X}}}\{R(f)\}. \tag{4}$$

If we reconsider our overarching goal stated earlier, then the smallest risk (expected loss) in the prediction of Y^{new} given X^{new} is achieved with the $f^\star$ of (3), and that theoretical optimal risk is $R^\star$ of (4), namely, $\mathbb{E}[\mathcal{L}(Y^{\text{new}}, f^\star(X^{\text{new}}))] = R^\star$. The theoretical optimal predictive model is therefore $f^\star$, although we must recognize that it is of no practical use as it cannot be computed. For instance, when we consider both classification and regression, the theoretical optimal predictive model $f^\star$ can be elicited and derived for some well-known foundational loss functions. For classification, an intuitive and indeed widely studied loss function is the so-called zero-one (0/1) loss function, defined simply with the indicator function as follows:

$$\mathcal{L}(y, f(\mathbf{x})) = \mathbb{1}(y \neq f(\mathbf{x})) = \begin{cases} 0 & \text{if} \quad y = f(\mathbf{x}), \\ 1 & \text{if} \quad y \neq f(\mathbf{x}). \end{cases} \tag{5}$$

When the zero-one loss function is used in classification, it can be shown quite easily that $R(f)$, the corresponding true risk (also known as theoretical risk or generalization error or true error), coincides with the misclassification probability $\texttt{Prob}_{(X,Y)\sim\psi}[Y \neq f(X)]$, namely,

$$\begin{aligned} R(f) &= \int_{\mathscr{X}\times\mathscr{Y}} \mathcal{L}(y, f(\mathbf{x}))\, p_{\mathbf{xy}}(\mathbf{x}, y)\, d\mathbf{x}dy \\ &= \mathbb{E}\left[\mathbb{1}(Y \neq f(X))\right] \\ &= \texttt{Prob}_{(X,Y)\sim\psi}[Y \neq f(X)]. \end{aligned} \tag{6}$$

This intuitive result is of paramount importance for practical aspects of statistical machine learning, because it provides an understandable frame of reference for the interpretation of the predictive performance of learning machines. Indeed, *the true error $R(f)$ of a classifier f therefore defines the probability that f misclassifies any arbitrary observation randomly drawn from the population of interest according to the distribution ψ.* $R(f)$ can also be interpreted as the *expected* disagreement between classifier $f(X)$ and the true label Y.

Definition 3 (The Bayes classifier)**.** Consider a pattern $\mathbf{x}$ from the input space and a class label y. Let $p(\mathbf{x}|y)$ denote the class conditional density of $\mathbf{x}$ in class y, and let $\texttt{Prob}[Y = y]$ denote the prior probability of class membership. The posterior probability of class membership is

$$\texttt{Prob}[Y = y|\mathbf{x}] = \frac{\texttt{Prob}[Y = y]p(\mathbf{x}|y)}{p(\mathbf{x})}. \tag{7}$$

Given $\mathbf{x} \in \mathscr{X}$ to be classified, the Bayes classification strategy consists of assigning $\mathbf{x}$ to the class with maximum posterior probability. With $h : \mathscr{X} \longrightarrow \mathscr{Y}$ denoting the Bayes classifier, we have, $\forall \mathbf{x} \in \mathscr{X}$,

$$h(\mathbf{x}) = \underset{c\in\mathscr{Y}}{\texttt{argmax}}\{\texttt{Prob}(Y = c|\mathbf{x})\}. \tag{8}$$

Theorem 1. *The minimizer of the zero-one risk over all possible classifiers is the Bayes classifier h defined in* (8)*:*

$$\begin{aligned} f^\star &= \underset{f}{\texttt{arginf}}\{R(f)\} = \underset{f}{\texttt{arginf}}\{\mathbb{E}\left[\mathbb{1}(Y \neq f(X))\right]\} \\ &= \underset{f}{\texttt{arginf}}\left\{\texttt{Prob}_{(X,Y)\sim\psi}[Y \neq f(X)]\right\} = h. \end{aligned} \tag{9}$$

Therefore, the Bayes classifier h defined in (8) *is the universal best classifier, such that $\forall \mathbf{x} \in \mathscr{X}$,*

$$\begin{aligned} f^\star(\mathbf{x}) &= h(\mathbf{x}) = \underset{c\in\mathscr{Y}}{\texttt{argmax}}\{\texttt{Prob}(Y = c|\mathbf{x})\} \\ &= \underset{c\in\mathscr{Y}}{\texttt{argmax}}\left\{\frac{\texttt{Prob}[Y = c]p(\mathbf{x}|c)}{p(\mathbf{x})}\right\}. \end{aligned} \tag{10}$$

The risk $R^\star$ corresponding to $f^\star$ is the smallest possible error that any classifier can achieve, i.e.,

$$R^\star = R(f^\star) = R(h) = \inf_{f}\{R(f)\}.$$

The fact that the *Bayes classifier* achieves the universal infimum error over all measurable classifiers is a fundamental result in pattern recognition and statistical learning. The probability theory for pattern recognition is made up of multiple results featuring learning machines whose performance is compared to the performance of the Bayes classifier [7] and [19]. Although this result is of more theoretical than practical importance, it turns out to provide a framework of reference for building more practical classification learning machines. Although we do not know the true density $p_{\mathbf{xy}}(\cdot, \cdot)$, we can assume a wide variety of possible densities in special cases, and then attempt the

construction of the Bayes classifier under those distributional assumptions. It is found in practice that when the assumptions are met (or almost met), the ensuing learning machine tends to exhibit superior predictive performances. For instance, under the assumption of multivariate Gaussian class conditional densities with equal covariance matrices in binary classification, one can derive the population Bayes Gaussian linear discriminant analysis classifier, whose estimator from the corresponding data yields the best predictive performance over all other learning machines. It bears repeating that this superior performance presupposes that the assumed multivariate Gaussianity is plausible. Every single aspect of optimal predictive model selection we have mentioned so far is strongly tied to the distributional characteristics of the space under consideration. In the case of superior predictive performances inherited from the correct assumption of the generator of the data, it must be said that practical data sets often arise from rather complex distributions that are often far too difficult to estimate. One could even consider estimating the density and then estimating the corresponding classifier. Unfortunately, the task of probability density estimation in complex high-dimensional spaces turns out to be a treacherous task, often more complex (statistically and computationally) than the classification task one would be intending to use density estimation for. Some researchers have resorted to semiparametric solutions like the use of mixtures of Gaussians (or mixtures of other parametric densities) to model their class conditional densities, and have done so with great success, although the analysis of mixtures is fraught with challenges, to the point that having to deal with those along with the main task of classification may render their use unattractive and not viable in this context. For this reason, practitioners and methodological and theoretical researchers tend to focus on more realizable goals than the hunt for the universal best learning machine. The approach consists of assuming that the function underlying the data (the decision boundary in the context of classification) is a member of a class of functions with some specific (sometimes desirable) properties. Of course, the very fact of choosing a specific function space automatically comes at the potential price of incurring an approximation error. In the example given earlier, assuming Gaussian class conditional probability densities with equal covariance matrices led to the derivation of a classifier belonging to the space of linear learning machines. In this case, the ensuing function space was implicit in the distributional choice. We will see later that the choice of the function space is often quite explicit and typically motivated by experience or pure convenience. Before we delve into the search for optimal predictive models in specific function spaces, it is useful to point out that fundamental statistical learning results exist in regression analysis that are similar to the ones presented earlier in the context of classification learning.

Theorem 2. *Consider functions $f : \mathbb{R}^p \to \mathbb{R}$ and the squared theoretical risk functional*

$$\begin{aligned} R(f) &= \mathbb{E}[(Y - f(X))^2] \\ &= \int_{\mathscr{X}\times\mathscr{Y}} (y - f(\mathbf{x}))^2 p_{\mathbf{xy}}(\mathbf{x}, y) d\mathbf{x}dy. \end{aligned} \tag{11}$$

Then the best function $f^\star = \underset{f}{\operatorname{arginf}}\{R(f)\}$ is given by the conditional expectation of Y given X; i.e., $\forall \mathbf{x} \in \mathscr{X}$,

$$f^\star(\mathbf{x}) = \mathbb{E}[Y|X = \mathbf{x}] = \int_{\mathscr{Y}} y\, p(y|\mathbf{x}) dy. \tag{12}$$

Theorem 2 provides the basic foundation of all regression analysis under the squared error loss. Clearly, the conditional expectation of Y given that $X = \mathbf{x}$ given in equation (12) is the theoretical optimal predictive function in regression, with a corresponding theoretical risk that is the baseline.

Theorem 3. *For every $f : \mathscr{X} \to \mathscr{Y}$, $R(f) = \int_{\mathscr{X}} (f(\mathbf{x}) - f^\star(\mathbf{x}))^2 d\,\psi(\mathbf{x}) + \sigma_\star^2$, where*

$$\begin{aligned} \sigma_\star^2 &= R^\star = R(f^\star) \\ &= \int_{\mathscr{X}\times\mathscr{Y}} (y - \mathbb{E}[Y|\mathbf{x}])^2 p_{\mathbf{xy}}(\mathbf{x}, y) d\mathbf{x}dy. \end{aligned} \tag{13}$$

Since the conditional density $p(y|\mathbf{x})$ of Y given $\mathbf{x}$, which is the main ingredient of $f^\star$, is not known in practice, the optimum remains a theoretical one and serves as a gold standard and reference when the squared error loss is used, as is often the case. In an effort to realize an estimator of the optimum with the data, one can consider the traditional nonparametric regression machinery. In one dimension, nonparametric regression works very well, but it unfortunately suffers from the curse of dimensionality. Just as with classification learning, one could relax the generality of $p(y|\mathbf{x})$ by assuming, for instance, a specific distribution. An example of this is the ubiquitous assumption of Gaussianity by which $p(y|\mathbf{x}) = \phi(y; h(\mathbf{x}), \sigma^2)$, where $h \in \mathscr{H}$ is a function with certain properties, taken from a function space $\mathscr{H}$. The function space $\mathscr{H}$ could be anything from the space of linear functions in the p-dimensional Euclidean space $\mathbb{R}^p$ to a space of certain nonlinear functions to reproducing kernel Hilbert spaces (RKHS) anchored by a suitably chosen kernel (similarity measure). We will seek to solve the more reasonable problem of choosing from a function space $\mathscr{H} \subset \mathscr{Y}^{\mathscr{X}}$ the function $f^\diamond \in \mathscr{H}$ that best estimates the dependencies between $\mathbf{x}$ and y. As stated earlier, trying to find $f^\star$ is hopeless. One needs to select a function space $\mathscr{H} \subset \mathscr{Y}^{\mathscr{X}}$, then choose the best function $f^\diamond_{\mathscr{H}}$ from $\mathscr{H}$, i.e.,

$$f^\diamond_{\mathscr{H}} = \arg \inf_{f\in\mathscr{H}} \Big\{\mathbb{E}[\mathcal{L}(Y, f(X))]\Big\}, \tag{14}$$

so that

$$R(f^\circ_{\mathscr{H}}) = R^\circ_{\mathscr{H}} = \inf_{f\in\mathscr{H}} R(f).$$

For notational simplicity, we will simply use f° and R° in place of $f^\circ_{\mathscr{H}}$ and $R^\circ_{\mathscr{H}}$, respectively. For the regression learning task, for instance, one could assume that the input space $\mathscr{X}$ is a closed and bounded interval of $\mathbb{R}$, i.e., $\mathscr{X} = [a,b]$, and then consider estimating the dependencies between $\mathbf{x}$ and y from within the space $\mathscr{H}$ of all bounded functions on $\mathscr{X} = [a,b]$, i.e.,

$$\mathscr{H} = \{f : \mathscr{X} \to \mathbb{R} |\ \exists B \geq 0 \text{ such that } |f(\mathbf{x})| \leq B\}.$$

One could further make the functions of the above $\mathscr{H}$ continuous, so that the space to search becomes

$$\mathscr{H} = \{f : [a,b] \to \mathbb{R} |\quad f \text{ is continuous}\} = C([a,b]),$$

which is the well-known space of all continuous functions on a closed and bounded interval $[a,b]$. This is indeed a very important function space. In fact, polynomial regression consists of searching our learning machine from a function space that is a subspace of $C([a,b])$. In other words, in polynomial regression learning, we are searching the space

$$\mathcal{P}([a,b]) = \{f \in C([a,b]) |\ f \text{ is a polynomial in } \mathbb{R}\}.$$

Interestingly, Weierstrass did prove that $\mathcal{P}([a,b])$ is dense in $C([a,b])$. One considers the space of all polynomials of some degree p, i.e.,

$$\mathscr{H} = \mathcal{P}^p([a,b]) = \Big\{f \in C([a,b]) |\ \exists \theta \in \mathbb{R}^{p+1}| \\ f(\mathbf{x}) = \sum_{j=0}^{p} \theta_j \mathbf{x}^j,\ \forall \mathbf{x} \in [a,b]\Big\}.$$

Similarly, for the classification learning task of binary pattern recognition with $\mathscr{Y} = \{-1,+1\}$, one may consider finding the best linear separating hyperplane, so that the corresponding function space is

$$\mathscr{H} = \Big\{f : \mathscr{X} \to \mathscr{Y} |\ \exists w_0 \in \mathbb{R}, \mathbf{w} \in \mathbb{R}^p :\ \forall \mathbf{x} \in \mathscr{X}, \\ f(\mathbf{x}) = \text{sign}\left(\mathbf{w}^\top \mathbf{x} + w_0\right)\Big\}, \tag{15}$$

or even a more complex function space capable of modelling and representing nonlinear decision boundaries like

$$\mathscr{H}(\Phi) = \Big\{f : \mathscr{X} \to \mathscr{Y} |\ \exists w_0 \in \mathbb{R}, \mathbf{w} \in \mathscr{F} :\ \forall \mathbf{x} \in \mathscr{X}, \\ f(\mathbf{x}) = \text{sign}\left(\langle \mathbf{w}, \Phi(\mathbf{x})\rangle + w_0\right)\Big\}, \tag{16}$$

where $\Phi : \mathscr{X} \longrightarrow \mathscr{F}$ is a mapping that projects each input $\mathbf{x}$ up to a high-dimensional feature space $\mathscr{F}$, thereby allowing the corresponding machine the capacity to capture nonlinear decision boundaries.

Empirical Foundations

Throughout the previous section, we explored some basic aspects of the theoretical foundations of optimal prediction model selection. It turns out that $f^\circ \in \mathscr{H}$, just like $f^\star$, cannot be computed because $p_{\mathbf{xy}}(\mathbf{x}, y)$ is *never* known in practice. What does happen in practice is that, given the data set $\mathscr{D}_n$ along with the chosen loss function $\mathcal{L}(\cdot,\cdot)$, the empirical risk $\widehat{R}(f)$ is defined as an estimator of the theoretical risk $R(f)$. From a practical perspective, given a data set $\mathscr{D}_n$, empirical risk minimization is used in place of theoretical risk minimization to construct estimators of $f^\star$, namely,

$$\widehat{f} = \widehat{f}_{\mathscr{H},n} = \widehat{f}_n = \underset{f\in\mathscr{H}}{\texttt{argmin}}\left\{\widehat{R}_n(f)\right\} \\ = \underset{f\in\mathscr{H}}{\texttt{argmin}}\left\{\frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(y_i, f(\mathbf{x}_i))\right\}. \tag{17}$$

Although the zero-one loss function allows us to theoretically define what constitutes the universal best optimal classifier, it cannot be used in any given function space to construct an estimated learning machine, because its use inherently implies an untenable combinatorial exploration. Fortunately, many other loss functions have been typically used in the search for optimal predictive models in statistical machine learning. With $f : \mathscr{X} \longrightarrow \{-1,+1\}$, and $h \in \mathscr{H}$ such that $f(\mathbf{x}) = \texttt{sign}(h(\mathbf{x}))$, some frequently used loss functions for binary classification include: (a) *Zero-one* (0/1) *loss:* $\mathcal{L}(y, f(\mathbf{x})) = \mathbb{1}(yh(\mathbf{x}) < 0)$, (b) *Hinge loss:* $\mathcal{L}(y, f(\mathbf{x})) = \max(1 - yh(\mathbf{x}), 0)$, (c) *Logistic loss:* $\mathcal{L}(y, f(\mathbf{x})) = \log(1 + \exp(-yh(\mathbf{x})))$, and (d) *Exponential loss:* $\mathcal{L}(y, f(\mathbf{x})) = \exp(-yh(\mathbf{x}))$. With $f : \mathscr{X} \longrightarrow \mathbb{R}$ and $f \in \mathcal{H}$, some loss functions for regression include: (a) $\mathcal{L}_1$ *loss:* $\mathcal{L}(y, f(\mathbf{x})) = |y - f(\mathbf{x})|$, (b) $\mathcal{L}_2$ *loss:* $\mathcal{L}(y, f(\mathbf{x})) = |y - f(\mathbf{x})|^2$, (c) ε*-insensitive* $\mathcal{L}_1$ *loss:* $\mathcal{L}(y, f(\mathbf{x})) = |y - f(\mathbf{x})| - \varepsilon$, and (d) ε*-insensitive* $\mathcal{L}_2$ *loss:* $\mathcal{L}(y, f(\mathbf{x})) = |y - f(\mathbf{x})|^2 - \varepsilon$. Other loss functions exist.

Although the empirical risk minimization principle provides an effective practical framework for learning patterns underlying the data, the estimator $\widehat{f}_{\mathscr{H},n}$ derived from it must be handled with great care and caution for a wide variety of reasons, which we now make clear. With the definitions of $f^\star$, f°, and now $\widehat{f}_{\mathscr{H},n}$ in hand, a natural and almost quintessential yet somewhat audacious question would be to assess the difference between $\widehat{f}_{\mathscr{H},n}$ and $f^\star$, maybe via some suitably defined norm, say $\|\widehat{f}_{\mathscr{H},n} - f^\star\|$, maybe using probabilistic measures like $\Pr[\|\widehat{f}_{\mathscr{H},n} - f^\star\|]$ or even $\mathbb{E}[\|\widehat{f}_{\mathscr{H},n} - f^\star\|]$, though it might not be trivial at all how to properly define such a norm, let alone the corresponding probability distribution. A difference like

$\|\widehat{f}_{\mathscr{H},n} - f^{\diamond}_{\mathscr{H}}\|_{\mathscr{H}}$ might be easier, although itself neither easy nor even practically realizable. The typical approach is to deal with the utility of the function like $R(f)$ rather than the function itself. Now, the relationship between $R(\widehat{f}_n)$ and the other theoretical risks is captured by the following cascade of inequalities, namely,

$$R(f^{\star}) \leq R(f^{\diamond}) \leq R(\widehat{f}_{\mathscr{H},n}). \tag{18}$$

The true risk $R(\widehat{f}_{\mathscr{H},n})$ of the realized estimator $\widehat{f}_{\mathscr{H},n}$ is clearly and unsurprisingly the largest of the three. Since $R^{\star}$ is unrealizable in practice, the natural goal should at least be: *Out of all the functions in $\mathscr{H}$ generated using the data $\mathscr{D}_n$, choose the one that best imitates $f^{\star}$, which means choose $\widehat{f}_{\mathscr{H},n} \in \mathscr{H}$ such that $\mathbb{E}[R(\widehat{f}_{\mathscr{H},n})] - R(f^{\star})$ is smallest.*

If one could directly (or even indirectly) construct $\widehat{f}^{(\mathtt{opt})}_{\mathscr{H},n} \in \mathscr{H}$ such that

$$\widehat{f}^{(\mathtt{opt})}_{\mathscr{H},n} = \underset{\widehat{f}_{\mathscr{H},n} \in \mathscr{H}}{\mathtt{argmin}} \{\mathbb{E}[R(\widehat{f}_{\mathscr{H},n})] - R(f^{\star})\},$$

then $\widehat{f}^{(\mathtt{opt})}_{\mathscr{H},n}$ would be the optimal predictive model. Unfortunately, such a function cannot be directly constructed in practice because its objective function is purely theoretical. The so-called *excess risk*, $\mathbb{E}[R(\widehat{f}_{\mathscr{H},n}) - R^{\star}]$, defined as the expected value of the difference between the true risk $R(\widehat{f}_n)$ associated with $\widehat{f}_n$ and the overall minimum risk $R^{\star}$, can be decomposed to explore in greater detail the source of error in the function estimation process:

$$\mathbb{E}[R(\widehat{f}_n) - R^{\star}] = \underbrace{\mathbb{E}[R(\widehat{f}_n) - R(f^{\diamond})]}_{\text{Estimation error}} + \underbrace{\mathbb{E}[R(f^{\diamond}) - R^{\star}]}_{\text{Approximation error}}. \tag{19}$$

Making the excess risk small is tricky because of the following dilemma: If the approximation error is made small, typically by making the function space $\mathscr{H}$ larger and more complex so that the members of $\mathscr{H}$ approximate $f^{\star}$ very well, then the corresponding estimation error tends to get undesirably larger. Many authors have written excessively on methods for achieving desirable trade-offs with favorable predictive benefits. The empirical risk $\widehat{R}_n(\widehat{f}_n)$ on $\widehat{f}_n$ can be made arbitrarily very small by making $\mathscr{H}$ very complex, leading to a phenomenon known as *overfitting*. It must be emphasized that such a function has very little to do with being optimal predictive, because the theoretical (true) risk $R(\widehat{f}_n)$ of such an $\widehat{f}_n$ is undesirably large. Indeed, when it comes to optimal prediction, it is crucial for the estimator $\widehat{f}_{\mathscr{H},n}$ to have an empirical risk $\widehat{R}_n(\widehat{f}_n)$ that is as close as possible to the true risk $R(\widehat{f}_n)$. Now, it is well known among practitioners that almost all statistical machine learning problems are inherently inverse problems, in the sense that learning methods seek to optimally estimate an unknown generating function using empirical observations assumed to be generated by it. As a result, statistical machine learning problems are inherently *ill-posed*, in the sense that they typically violate at least one of Hadamard's three *well-posedness* conditions. For clarity, according to Hadamard a problem is *well-posed* if it fulfills the following three conditions: (a) *a solution exists;* (b) *the solution is unique; and* (c) *the solution is stable, i.e., does not change drastically under small perturbations.* For many machine learning problems, the first condition of well-posedness, namely, existence, is fulfilled. However, the solution is either not unique or not stable. With large p small n for instance, not only is there a *multiplicity* of solutions but also the instability thereof, due to the singularities resulting from the fact that $n \lll p$. Typically, the regularization framework is used to isolate a feasible and optimal (in some sense) solution. *Tikhonov*'s regularization is the one most commonly resorted to and typically amounts to a Lagrangian formulation of a constrained version of the initial problem, the constraints being the objects used to isolate a unique and stable solution.

Effect of Model Complexity

To gain deeper insights into the properties and challenges inherent in optimal predictive model selection, we now consider a practical exploration of univariate regression learning using the polynomial function space, namely,

$$\begin{aligned}\mathscr{H} \quad = \quad & \Big\{ f \in C([a,b]) |\ \exists \theta_0, \theta_1, \ldots, \theta_p \in \mathbb{R} | \\ & f(\mathbf{x}) = \sum_{j=0}^{p} \theta_j \mathbf{x}^j \ \forall \mathbf{x} \in [a,b] \Big\}.\end{aligned}$$

Having chosen our function space $\mathscr{H}$ along with the squared error loss, our statistical learning task consists of finding the minimizer of the empirical counterpart of the average squared errors (ASE), i.e.,

$$\begin{aligned}\widehat{f}_{\mathscr{H},n} \quad &= \quad \widehat{f}_n = \widehat{f} = \underset{f \in \mathscr{H}}{\arg\min} \left\{ \widehat{\mathrm{ASE}}(f) \right\} \\ &= \quad \underset{f \in \mathscr{H}}{\arg\min} \left\{ \frac{1}{n} \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i; \boldsymbol{\theta}))^2 \right\}. \end{aligned} \tag{20}$$

We are seeking the best member of the function space $\mathscr{H}$ based on the given data set $\mathscr{D}_n$. Since we specifically chose the function space of all univariate real-valued polynomials of degree at most p in some interval $[a,b]$, finding $\widehat{f}$ comes down to estimating the coefficients of the polynomial using the data. Using the $n \times (p+1)$ Vandermonde matrix $\mathbf{X} = (\mathbf{x}_i^j), i = 1, \ldots, n, j = 0, \ldots, p$, and $\mathbf{Y} \in \mathbb{R}^n$, the

solution to problem (20) is given by

$$\begin{aligned}\hat{\theta} &= \underset{\theta\in\mathbb{R}^{p+1}}{\arg\min}\left\{\frac{1}{n}\sum_{i=1}^{n}\left(Y_i-\sum_{j=0}^{p}\theta_j\mathbf{x}_i^j\right)^2\right\}\\ &= \underset{\theta\in\Theta}{\operatorname{argmin}}\{(\mathbf{Y}-\mathbf{X}\theta)^\top(\mathbf{Y}-\mathbf{X}\theta)\}\\ &= (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}. \qquad (21)\end{aligned}$$

The estimator in equation (21) has many quintessential layers that are crucial to the understanding of optimal predictive model selection. It is therefore important to dissect and unpack those key aspects of statistical learning.

(a) *Stochastic nature of the estimator.* First and foremost, the estimate $\hat{\theta} = (\hat{\theta}_0, \hat{\theta}_1, \dots, \hat{\theta}_p)^\top$ of $\theta = (\theta_0, \theta_1, \dots, \theta_p)^\top$ is a random variable, and as a result the estimate $\widehat{f}(\mathbf{x}) = \widehat{f}(\mathbf{x};\hat{\theta}) = \hat{\theta}_0 + \hat{\theta}_1\mathbf{x} + \hat{\theta}_2\mathbf{x}^2 + \cdots + \hat{\theta}_p\mathbf{x}^p$ of $f^\star(\mathbf{x})$ is also a random variable. We therefore have to be mindful whenever $\widehat{f}$ is used, that it is inherently a random entity whose handling is best done with the powerful machineries of probability and statistics.

(b) *Bias and variance.* Since $\widehat{f}(\mathbf{x})$ is a random variable, we must compute important aspects like its bias $\mathbb{B}[\widehat{f}(\mathbf{x})] = \mathbb{E}[\widehat{f}(\mathbf{x})] - f^\star(\mathbf{x})$, which measures how far our chosen class of models is from the true generator of the data, and its variance $\mathbb{V}[\widehat{f}(\mathbf{x})] = \mathbb{E}[(\widehat{f}(\mathbf{x}) - \mathbb{E}[\widehat{f}(\mathbf{x})])^2]$, which, as the name says, tells us relatively how stable the constructed estimator is.

(c) *Model complexity and temptation to overfit.* Since our goal expressed through the objective function is to find the member of the class $\mathscr{H}$ that minimizes the empirical risk, it is very tempting at first to use the data at hand to build the $\hat{f}$ that makes $\widehat{\mathrm{ASE}}(\hat{f})$ the smallest. For instance, the higher the value of $\mathbf{p}$, the smaller $\widehat{\mathrm{ASE}}(\widehat{f}(\cdot))$ will get. In fact, in the most extreme of scenarios, one could simply make $\widehat{\mathrm{ASE}}(\widehat{f}(\cdot)) = 0$ by specifying $\widehat{f}(\mathbf{x}_i) = y_i \forall i = 1, \dots, n$. In a sense, we have a dilemma: If we make $\widehat{f}$ complex (large p), we make the bias small, but the variance is increased. If we make $\widehat{f}$ simple (small p), we make the bias large, but the variance is decreased. In this case, the degree p of the polynomial represents the complexity of the corresponding model. In the end, we will have to come up with various criteria for estimating the optimal complexity, in the sense of the one that leads to low prediction error. To help gain deeper insights into this fundamental statistical machine learning phenomenon, let's consider the synthetic (artificial) task of learning a univariate polynomial regression from the data. We simulate the data using the function $f^\star(\mathbf{x}) = -\mathbf{x} + \sqrt{2}\sin(\pi^{3/2}\mathbf{x}^2)$ for $\mathbf{x} \in [-1, +1]$, with a noise variance $\sigma^2 = 0.3^2$.

Figure 1 helps us gain insights into the basics of bias-variance trade-off. The polynomial of degree 1, which happens to be the model with lowest nonzero complexity,

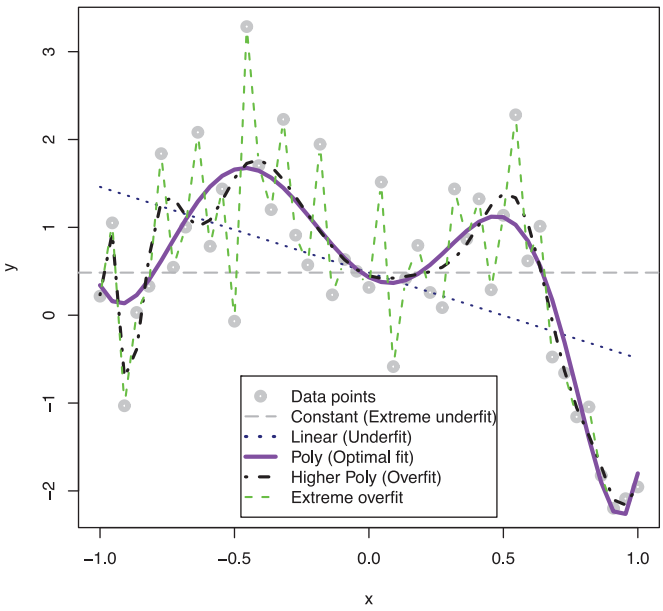


Figure 1. Effect of complexity on estimated function.

performs poorly, as does the perfect memorizer whose complexity is virtually infinite since it simply connects all the points. The solid line model does a great job learning the underlying function. The low complexity models attempt to avoid a large estimation variance but then pay a price in the form of an increased bias, resulting in a large prediction error. The high complexity models attempt to fit too well, literally memorizing the data in the extreme case, and thereby learning both the noise and the signal, resulting in a large variance as the price paid for low bias, ultimately yielding another high prediction error. The optimal fit depicted by the solid line model is achieved by settling for a trade-off between bias and variance. The task dedicated to determining that optimal complexity, which results in the optimal predictive performance, occupies a central place in statistical machine learning and will be further discussed throughout this paper. The phenomenon of bias-variance trade-off is of fundamental importance and can be further explained in the context of regression learning by the so-called bias-variance decomposition of the theoretical risk on $\widehat{f}(\cdot)$ under the squared error loss. Let's consider the data set $\mathscr{D}_n$. Let's also assume that $Y_i = f^\star(\mathbf{x}_i) + \varepsilon_i$, where the ε_i's are i.i.d. from some distribution with $\mathtt{mean}(\varepsilon) = 0$ and $\mathtt{variance}(\varepsilon) = \sigma^2$. Let $\widehat{f}$ be our estimator of $f^\star$ built using the random sample provided. Let $\mathbf{x} \in \mathscr{X}$. The pointwise bias-variance decomposition of the expected squared error is given by $R(\widehat{f}) = \mathbb{E}[(Y - \widehat{f}(\mathbf{x}))^2] = \sigma^2 + \mathtt{Bias}^2(\widehat{f}(\mathbf{x})) + \mathtt{variance}(\widehat{f}(\mathbf{x}))$, where $\sigma^2 = \mathtt{variance}(\varepsilon)$ is the variance of the noise term but essentially represents the irreducible learning error, that is, the error inherent in the structure of the population, one that cannot be changed by any learning machine. It

is easy to verify that this is the smallest possible error, i.e., $R^\star = R(f^\star) = \mathbb{E}[(Y - f^\star(\mathbf{x}))^2] = \texttt{variance}(\varepsilon) = \sigma^2$. Interestingly, the bias-variance phenomenon depicted in Figure 3 and Figure 1 in the context of regression learning is also present in classification learning. A detailed account of the same type of decomposition for the 0/1 loss used in classification can be found in [12] and [8]. The optimal decision boundary seen in Figure 2 is obtained using cross validation on the k-Nearest Neighbors learning machine (22) for various values of k. Clearly, k does indeed control the complexity of the underlying model, namely, the decision boundary. Although the decision boundary in this case cannot be explicitly written or learned as the optimum of some explicit objective function, one can still use cross validation to determine the optimal value of k (optimal neighborhood size). This tremendous flexibility of the cross validation principle is certainly one of its greatest strengths, which makes it very appealing and widely applicable in statistical machine learning. For classification, $\mathscr{Y} = \{1, \dots, G\}$,

$$\widehat{f}^{(\text{kNN})}(\mathbf{x}) = \underset{g \in \mathscr{Y}}{\operatorname{argmax}} \left\{ \frac{1}{k} \sum_{i=1}^{n} \mathbb{1}(y_i = g) \mathbb{1}(\mathbf{x}_i \in \mathcal{V}_k(\mathbf{x})) \right\}. \quad (22)$$

For regression, $\mathscr{Y} = \mathbb{R}$,

$$\widehat{f}^{(\text{kNN})}(\mathbf{x}) = \frac{1}{k} \sum_{i=1}^{n} y_i \mathbb{1}(\mathbf{x}_i \in \mathcal{V}_k(\mathbf{x})). \quad (23)$$

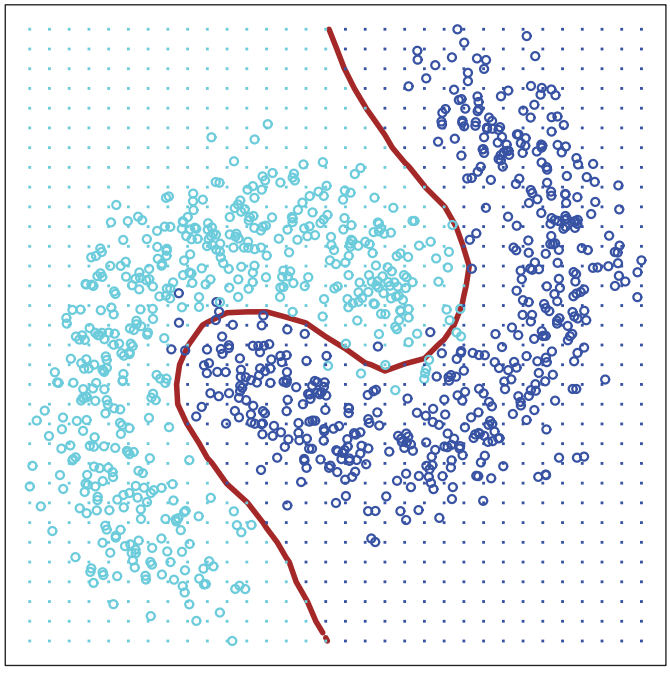


Figure 2. Optimal kNN decision boundary.

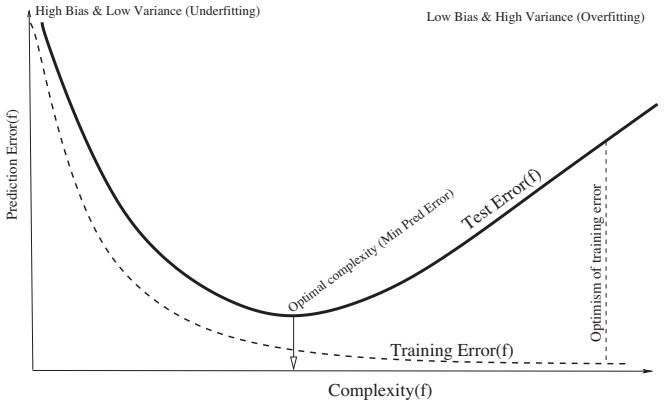


Figure 3. Bias-variance trade-off and model complexity.

Elements of Model Identification

Once a specific function space is chosen for our learning task, like we did earlier with our choice of the space of univariate real-valued polynomials, it is not enough to know the highest polynomial degree for our particular regression learning task. Indeed, we also need to know which of the coefficients are nonzero. In other words, we need a clear and unambiguous way, like an index, to distinguish the members of $\mathscr{H}$ so that we can identify and then select specific ones. To help clarify that, we can think of the function space $\mathscr{H}$ in this case as a vector space with the monomials $\{\mathbf{x}, \mathbf{x}^2, \dots, \mathbf{x}^j, \dots, \mathbf{x}^p\}$ as the basis vectors or atoms of the expansion that help span the space. In general, one considers a basis set $\{\mathrm{B}_1(\mathbf{x}), \mathrm{B}_2(\mathbf{x}), \dots, \mathrm{B}_j(\mathbf{x}), \dots, \mathrm{B}_p(\mathbf{x})\}$, so that for polynomial regression, $\mathrm{B}_j(\mathbf{x}) = \mathbf{x}^j$. Using the basis set, a member $f \in \mathscr{H}$ can then be specified by simply indicating which of the monomials are combined together to form its representation. For our space of univariate real-valued polynomials of degree at most p, we could use one of the key building blocks of the parametric model selection machinery, namely, a vector of indicator variables. With the p original atoms, there are $2^p - 1$ nonempty models, each corresponding to a subset of the provided atoms. We shall use a vector $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \dots, \gamma_p)^\top$ to denote the index of a given model, with each γ_j being an indicator of the atom's presence in the model under consideration, namely, $\gamma_j = \mathbb{1}\left(\text{atom } \mathrm{B}_j(\mathbf{x}) \text{ appears in model } M_{\boldsymbol{\gamma}}\right)$.

For simplicity we shall assume no intercept; i.e., $\theta_0 = 0$. Here, $\boldsymbol{\gamma} = (1, 1, \dots, 1)^\top$ corresponds to the *full model* M_f, while $\boldsymbol{\gamma} = (0, 0, \dots, 0)^\top$ corresponds to the empty model, also referred to as the *null model*, and given by $M_0 : \mathbf{Y} = \boldsymbol{\varepsilon}$ (pure zero-mean noise).

Equipped with this index, $|M_{\boldsymbol{\gamma}}| = |f_{\boldsymbol{\gamma}}| = p_{\boldsymbol{\gamma}} = \sum_{j=1}^{p} \gamma_j$ is the number of atoms in model $M_{\boldsymbol{\gamma}}$, and $\boldsymbol{\theta}_{\boldsymbol{\gamma}} \in \mathbb{R}^{p_{\boldsymbol{\gamma}}}$ is the subset of $\boldsymbol{\theta} \in \mathbb{R}^p$ made up of only the θ_j's picked up by $\boldsymbol{\gamma}$, that is, $\theta_{\gamma_j} = \gamma_j \theta_j$. Finally, $\mathbf{X}_{\boldsymbol{\gamma}}$ is the submatrix of $\mathbf{X}$ whose columns are only those $p_{\boldsymbol{\gamma}}$ columns of $\mathbf{X}$ picked up by $\boldsymbol{\gamma}$, so that $\mathbf{X}_{\boldsymbol{\gamma}}$ is really an $n \times p_{\boldsymbol{\gamma}}$ matrix, and the corresponding

model M_γ is given by

$$M_\gamma : \mathbf{Y} = \mathbf{X}_\gamma \theta_\gamma + \varepsilon. \tag{24}$$

Putting everything together, we define a function space $\mathscr{H}$ as the hypothesis space containing the pattern underlying our data, but in a sense, using the language of models, we are somewhat dealing with a model space $\mathscr{M}$. Having now defined the useful concept of index (indicator vector) of a given model, we can unambiguously specify members $f_\gamma \in \mathscr{H}$ or $M_\gamma \in \mathscr{M}$, using the vector $\gamma \in \mathbf{\Gamma} = \{0,1\}^p$, which represents the indexing of that specific member of the model space $\mathscr{M}$ or, equivalently, the function space $\mathscr{H}$. Clearly, $\mathbf{\Gamma}$ is made up of the 2^p models. For our polynomial regression task, we are in the presence of the so-called parametric family of models, in the sense that the choice of a member M_γ of the model space $\mathscr{M}$ through its index vector γ maps to the corresponding collection of parameters contained in θ_γ. In such a parametric context, the unambiguous specification of a model or the corresponding function thereof typically indicates both the model M_γ and the corresponding parameter vector θ_γ. Let $\tilde{\mathbf{x}} = (\mathrm{B}_1(\mathbf{x}), \mathrm{B}_2(\mathbf{x}), \ldots, \mathrm{B}_p(\mathbf{x}))^\top$ and $\mathbf{V}_\gamma \in \{0,1\}^{p \times p_\gamma}$, such that $\mathbf{V}_\gamma[j,k] = \gamma_j, j = 1, \ldots, p, k = 1, \ldots, p_\gamma$. Any member $f_\gamma = f_\gamma(\mathbf{x}|\theta_\gamma, M_\gamma) \in \mathscr{H}$ can be fully specified as

$$\begin{aligned} f_\gamma(\mathbf{x}|\theta_\gamma, M_\gamma) &= \tilde{\mathbf{x}}^\top \mathbf{V}_\gamma \theta_\gamma \\ &= \sum_{j=1}^{p} \gamma_j \theta_{\gamma_j} \mathrm{B}_j(\mathbf{x}). \end{aligned} \tag{25}$$

For any $M_\gamma \in \mathscr{M}$, the ordinary least squares (OLS) estimate encountered earlier in equation (21) is now given by

$$\hat{\theta}_\gamma^{(\mathrm{OLS})} = \hat{\theta}_\gamma = (\mathbf{X}_\gamma^\top \mathbf{X}_\gamma)^{-1} \mathbf{X}_\gamma^\top \mathbf{Y}. \tag{26}$$

It is easy to see that the ordinary least squares prediction of average response at $\mathbf{x}$ is given by

$$\begin{aligned} \widehat{f}_\gamma^{(\mathrm{OLS})}(\mathbf{x}) &= \widehat{f}_\gamma^{(\mathrm{OLS})}(\mathbf{x}|\hat{\theta}_\gamma^{(\mathrm{OLS})}, M_\gamma) \\ &= \tilde{\mathbf{x}}^\top \mathbf{V}_\gamma \hat{\theta}_\gamma = \sum_{j=1}^{p} \gamma_j \hat{\theta}_{\gamma_j} \mathrm{B}_j(\mathbf{x}). \end{aligned} \tag{27}$$

It is important to note that the identifier of functions need not be a vector as in the above parametric modelling scenario. In nonparametric univariate regression learning, for instance, the identifier of a member of the Naradaya–Watson space of estimators is simply a real scalar, which is the bandwidth of the kernel used in the estimation

$$\widehat{f}_\gamma^{(\mathrm{NW})}(\mathbf{x}) = \sum_{i=1}^{n} y_i K\left(\frac{\mathbf{x} - \mathbf{x}_i}{\gamma}\right) \Big/ \sum_{\ell=1}^{n} K\left(\frac{\mathbf{x} - \mathbf{x}_\ell}{\gamma}\right). \tag{28}$$

For this nonparametric scenario, the model index γ suffices to fully specify the model, as there are no parameters in the traditional sense of a finite collection of model coefficients. Here $\gamma \in \mathbf{\Gamma} \subseteq \mathbb{R}_+^\star$, which means that our model space search is done on an infinite subset of the right-hand side of the real number line. For the k-Nearest Neighbors learning machine, the complexity of the implicit underlying model is measured by k, the size of the neighborhood, which is a discrete number from 1 to n. Therefore, for kNN, $\gamma \in \mathbf{\Gamma} = \{1, 2, \ldots, n\}$. In practice, this is truncated to a reasonable maximum number of neighbors.

Model Selection Criteria

When it comes to model selection for optimal prediction both Bayesian statistics and non-Bayesian statistics have contributed richly. Essentially, one can identify four main ways to address the quest for optimal prediction: namely, (a) Selection, (b) Compression, (c) Regularization, and (d) Aggregation. The first three approaches operate under the strong assumption that a single member of function space $\mathscr{H}$ exists with optimal predictive properties, and all the methods and techniques seek to find that unique member. All the existing criteria are carefully created, designed, and developed to help yield that member of $\mathscr{H}$. On the other hand, aggregation, also known as ensemble learning or model averaging or model combination, takes the view that a single optimum might not exist. Aggregation operates on the assumption that many decent candidate models exist, and instead of needlessly wasting time to seek a unique optimum that one may never find, it is better to combine the good candidates in some fashion to yield an overall lower prediction (generalization) error. Over the years, aggregation techniques like Bayesian Model Averaging (BMA) [2, 11], Bootstrap Aggregating (Bagging) [3], Random Forest [4], Random Subspace Learning, Stacking, and certainly Adaptive Boosting [14] and Gradient Boosting have emerged and continue to be developed. Interestingly, these so-called ensemble learning methods tend to yield the best predictive performances in practical applications.

Likelihood based selection. In the presence of a multiplicity of potential models competing to fit the data, and considering that the estimators of those models are based on random samples with inherently built-in uncertainty, it makes sense to assume that any choice of a model consequently has built-in uncertainty. Before the data is collected and the model built, $p(M_\gamma)$ represents its prior probability. Once the data is collected, the posterior probability $p(M_\gamma|\mathscr{D}_n)$ of model M_γ provides a reasonable mechanism for assessing and measuring the uncertainty attached to its selection. Now, using $m_\gamma(\mathscr{D}_n) = p(\mathscr{D}_n|M_\gamma) = \int_\Theta p(\mathscr{D}_n|\theta_\gamma, M_\gamma) p(\theta_\gamma) d\theta_\gamma$, we can write

$$\begin{aligned} p(M_\gamma|\mathscr{D}_n) &= \frac{p(M_\gamma) m_\gamma(\mathscr{D}_n)}{\sum_{\gamma'} p(M_{\gamma'}) m_{\gamma'}(\mathscr{D}_n)} = \frac{p(\mathscr{D}_n|M_\gamma) p(M_\gamma)}{p(\mathscr{D}_n)} \\ &= \frac{p(\mathscr{D}_n|M_\gamma) p(M_\gamma)}{\sum_{\ell=1}^{2^p} p(\mathscr{D}_n|M_\ell) p(M_\ell)}. \end{aligned} \tag{29}$$

In a parametric context like the one introduced in "Elements of Model Identification," the Bayesian estimator of the parameter vector θ_γ for model $M_\gamma \in \mathscr{M}$ is given by

$$\begin{aligned}\tilde{\theta}_\gamma^{(\text{Bayes})} &= \tilde{\theta}_\gamma = \mathbb{E}[\theta_\gamma | M_\gamma, \mathscr{D}_n] \\ &= \int \theta_\gamma p(\theta_\gamma | M_\gamma, \mathscr{D}_n) d\theta_\gamma. \end{aligned} \tag{30}$$

From a Bayesian perspective, if model M_γ is selected, then the predictor of the response Y given $\mathbf{x}$ is given by

$$\begin{aligned}\widehat{f}_\gamma^{(\text{Bayes})}(\mathbf{x}) &= \tilde{\mathbf{x}}^\top \mathbf{V}_\gamma \mathbb{E}[\theta_\gamma | M_\gamma, \mathscr{D}_n] = \mathbf{x}^\top \mathbf{V}_\gamma \tilde{\theta}_\gamma \\ &= \sum_{j=1}^{p} \gamma_j \mathrm{B}_j(\mathbf{x}) \tilde{\theta}_{\gamma_j}. \end{aligned} \tag{31}$$

Under the squared error loss, the Bayesian Model Averaging (BMA) predictor provides the optimal predictor [2, 11], whose corresponding prediction function is given by

$$\widehat{f}_\gamma^{(\text{BMA})}(\mathbf{x}) = \sum_{\gamma \in \mathbf{\Gamma}} \sum_{j=1}^{p} \gamma_j p(M_\gamma | \mathscr{D}_n) \mathrm{B}_j(\mathbf{x}) \hat{\theta}_{\gamma_j}. \tag{32}$$

The median probability model introduced and developed in [2] seeks to achieve both optimal prediction and consistent model selection. The quintessential element in the construction of the median probability model is the posterior inclusion probability $\texttt{PIP}_j$ of atom $\mathrm{B}_j(\mathbf{x})$, with

$$\texttt{PIP}_j = \Pr[\gamma_j = 1 | \mathscr{D}_n] = \sum_{\gamma \in \mathbf{\Gamma}} \gamma_j p(M_\gamma | \mathscr{D}_n). \tag{33}$$

The median probability model index vector is given by $\gamma^{(\text{med})} \in \mathbf{\Gamma} = \{0,1\}^p$, where $\gamma^{(\text{med})} = \mathbb{1}\left(\texttt{PIP}_j \geq \frac{1}{2}\right)$. The median probability model is the model made up of atoms appearing in at least half of the models in the model space. The main limitation of the median probability model lies in the fact that the model does not always exist, mainly due to the rigidity of the threshold. In [9] I remedied this limitation by suggesting a flexibility and adaptive approach for optimal predictive atom selection in the general basis function expansion framework. An alternative to the median probability model is the highest posterior model, whose model index vector is given by

$$\gamma^{(\text{HPM})} = \underset{\gamma \in \mathbf{\Gamma}}{\texttt{argmax}} \left\{ p(M_\gamma | \mathscr{D}_n) \right\}.$$

Recall also that given a model $M_\gamma \in \mathscr{M}$, along with the corresponding $\theta_\gamma \in \mathbb{R}^{p_\gamma}$, the likelihood of θ_γ is

$$\begin{aligned} L(\theta_\gamma | M_\gamma, \mathscr{D}_n) &= p(\mathscr{D}_n | f_\gamma(\mathbf{X} | \theta_\gamma, M_\gamma)) \\ &= \prod_{i=1}^{n} p(y_i | f_\gamma(\mathbf{x}_i | \theta_\gamma, M_\gamma)), \end{aligned} \tag{34}$$

and the maximum likelihood estimator of θ_γ is

$$\hat{\theta}_\gamma^{(\text{MLE})} = \underset{\theta_\gamma \in \mathbb{R}^{p_\gamma}}{\texttt{argmax}} \Big\{ \log L(\theta_\gamma | M_\gamma, \mathscr{D}_n) \Big\}. \tag{35}$$

The Schwarz Bayesian Information Criterion (BIC) [15], although very prevalent in non-Bayesian settings, just happens, as its name suggests, to have a Bayesian origin. The model index $\gamma^{(\text{BIC})}$ of a model $M_\gamma \in \mathscr{M}$ is given by

$$\gamma^{(\text{BIC})} = \underset{\gamma \in \mathbf{\Gamma}}{\texttt{argmin}} \left\{ \texttt{BIC}_n(M_\gamma) \right\}, \tag{36}$$

where the score $\texttt{BIC}_n(M_\gamma)$ of model $M_\gamma \in \mathscr{M}$ is

$$\texttt{BIC}_n(M_\gamma) = -2 \log L(\hat{\theta}_\gamma | M_\gamma; \mathscr{D}_n) + |M_\gamma| \log n. \tag{37}$$

The Akaike Information Criterion (AIC) [1], where the score $\texttt{AIC}_n(M_\gamma)$ of model $M_\gamma \in \mathscr{M}$ is defined as

$$\texttt{AIC}_n(M_\gamma) = -2 \log L(\hat{\theta}_\gamma | M_\gamma; \mathscr{D}_n) + 2|M_\gamma|, \tag{38}$$

predates BIC, and while BIC is regarded as the chief selection criterion, AIC has enjoyed the distinct property of yielding typically better predictive performances.

Elements of cross validation. A more universally applicable model selection score is the ubiquitous cross validation score. In its most general formulation, the V-fold cross validation score proceeds by deterministically dividing the data set $\mathscr{D}_n$ into V chunks (folds) of almost equal sizes, such that $\mathscr{D}_n = \bigcup_{\mathrm{v}=1}^{V} \mathscr{D}_\mathrm{v}$ and $n = \sum_{\mathrm{v}=1}^{V} |\mathscr{D}_\mathrm{v}|$. The cross validation score is given by

$$\texttt{CV}(\widehat{f}) = \frac{1}{V} \sum_{\mathrm{v}=1}^{V} \hat{\varepsilon}_\mathrm{v}, \tag{39}$$

where

$$\hat{\varepsilon}_\mathrm{v} = \frac{1}{|\mathscr{D}_\mathrm{v}|} \sum_{i=1}^{n} \mathbb{1}(\mathbf{z}_i \in \mathscr{D}_\mathrm{v}) \mathcal{L}(\mathrm{y}_i, \widehat{f}^{(-\mathscr{D}_\mathrm{v})}(\mathbf{x}_i)),$$

and $\widehat{f}^{(-\mathscr{D}_\mathrm{v})}(\cdot)$ is the estimator of f constructed without the vth chunk $\mathscr{D}_\mathrm{v}$ of $\mathscr{D}_n$. An algorithmic (pseudo-code) description is given below in Algorithm 1 to help build an intuitive understanding of this most general of model selection scores. In practice, the data is often randomly shuffled prior to the deterministic splitting into chunks. The oldest incarnation of the cross validation principle is leave one out cross validation, which corresponds to $V = n$. It is important to mention here that cross validation is one of the most used approaches to model selection for optimal prediction in statistical machine learning. From its earliest days with M. Stone's [17] seminal paper, along with its wide variety of extensions and adaptations, like [16], the cross validation principle has continually played a central role in the selection of various types of model hyperparameters. In virtually all the model spaces considered in this paper, cross validation is the default approach for empirical intraspace model comparison and model selection. When classification and regression trees are used as the function space, their pruning is done via cross validation. Cross validation is also used as one way to estimate

the number of base learners in ensemble learning methods like Bagging [3] or Random Forest [4] or even adaptive boosting [14]. Cross validation also plays a central role in support vector machine classification and support vector regression learning, as well as in ridge regression [10] and the famous LASSO [18] and its extension. In short, cross validation is central to non-Bayesian regularization. One of the greatest appeals of the cross validation principle lies in its generality, its flexibility, and its wide applicability. Cross validation is typically used for determining the optimal complexity in both parametric and nonparametric function spaces, but also crucially for selecting the specific member of the function space that achieves the lowest prediction error, provided such a unique member exists. It is important to know that there are learning machines, and very good ones at that, that are constructed purely algorithmically. While it is difficult or even at times impossible to use some of the other optimal predictive model selection criteria on purely algorithmic machines like k-Nearest Neighbors learning machines of equations (22) and (23), it is straightforward to use cross validation on them, as long as the error is well defined. Cross validation applies nicely to the most interpretable learning machines, namely, classification and regression trees, which are built purely algorithmically but still benefit from the predictive power and flexibility of the cross validation principle.

Algorithm 1: V-fold Cross Validation

Input: Training data $\mathscr{D}_n = \{\mathbf{z}_i = (\mathbf{x}_i^\top, \mathrm{y}_i)^\top,\ i = 1, \ldots, n\}$, where $\mathbf{x}_i^\top \in \mathscr{X}$ and $\mathrm{y}_i \in \mathscr{Y}$, and the function of interest is denoted by f, sample size n, number of folds V

Output: Cross Validation score $\mathtt{CV}(\widehat{f})$

for $\mathrm{v} = 1$ *to* V **do**

Extract the validation set $\mathscr{D}_\mathrm{v} = \{\mathbf{z}_i \in \mathscr{D}_n :\ i \in [1 + (\mathrm{v}-1) \times m, \mathrm{v} \times m]\}$

Extract the training set $\mathscr{D}_\mathrm{v}^c := \mathscr{D}_n \backslash \mathscr{D}_\mathrm{v}$

Build the estimator $\widehat{f}^{(-\mathscr{D}_\mathrm{v})}(\cdot)$ using $\mathscr{D}_\mathrm{v}^c$

Compute predictions $\widehat{f}^{(-\mathscr{D}_\mathrm{v})}(\mathbf{x}_i)$ for $\mathbf{z}_i \in \mathscr{D}_\mathrm{v}$

Compute the validation error for the vth chunk

$$\hat{\varepsilon}_\mathrm{v} = \frac{1}{|\mathscr{D}_\mathrm{v}|} \sum_{i=1}^{n} \mathbb{1}(\mathbf{z}_i \in \mathscr{D}_\mathrm{v}) \mathcal{L}(\mathrm{y}_i, \widehat{f}^{(-\mathscr{D}_\mathrm{v})}(\mathbf{x}_i))$$

Compute the CV score $\mathrm{CV}(\widehat{f}) = \dfrac{1}{V} \displaystyle\sum_{\upsilon=1}^{V} \hat{\varepsilon}_\mathrm{v}$

Regularized risk minimization. One of the fundamental results in statistical learning theory has to do with the fact that the minimizer of the empirical risk could turn out to be overly optimistic and lead to poor generalization performance. It is indeed the case that by making our estimated classifier very complex, it can adapt too well to the data at hand, meaning very low in-sample error rate, but yield very high out-of-sample error rates due to overfitting, the estimated classifier having learned both the signal and the noise. In technical terms, this is referred to as the bias-variance dilemma, in the sense that by increasing the complexity of the estimated learning machine, the bias is reduced (good fit all the way to the point of overfitting) (see Figure 2), but the variance of that estimator is increased. On the other hand, considering much simpler estimators leads to less variance but higher bias (due to underfitting, model not rich enough to fit the data well). This phenomenon of the bias variance dilemma is particularly potent with massive data when the number of predictor variables p is much larger than the sample size n. One of the main tools in the modern machine learning arsenal for dealing with this is the so-called regularization framework, whereby instead of using the empirical risk alone, a constrained version of it, also known as the regularized or penalized version, is used. Indeed, within a selected space $\mathscr{H}$ of potential learning machines, one typically chooses some loss function $\mathcal{L}(\cdot, \cdot)$ with some desirable properties like smoothness or convexity (this is because one needs at least to be able to build the desired classifier), and then finds the minimizer of its regularized version, i.e.,

$$\widehat{f}_{\mathscr{H},\lambda,n} = \underset{f \in \mathscr{H}}{\mathtt{argmin}} \left\{ \widehat{R}_{\mathscr{H},n}(f) + \lambda \Omega_{\mathscr{H}}(f) \right\}, \tag{40}$$

where λ controls the bias-variance trade-off. Typically, $\lambda > 0$ and is determined by cross validation. Cross validation for determining λ proceeds by defining a grid $\Lambda \subset \mathbb{R}_+^\star = (0, +\infty)$ of possible values of λ. Sometimes, based on intuition or experience, it could just be $\Lambda = [\lambda_{\mathtt{min}}, \lambda_{\mathtt{max}}]$; then

$$\hat{\lambda}^{(\mathrm{opt})} = \underset{\lambda \in \Lambda}{\mathtt{argmin}} \left\{ \mathrm{CV}(\widehat{f}_\lambda) \right\}. \tag{41}$$

$\widehat{f}_{\mathscr{H},\hat{\lambda}^{(\mathrm{opt})},n}$ is clearly far better than $\widehat{f}_{\mathscr{H},n}$ from equation (17). By inherent design, the cross validation mechanism endows $\widehat{f}_{\mathscr{H},\hat{\lambda}^{(\mathrm{opt})},n}$ with some predictive power, making it an estimator with the potential for predictive optimality. As long as the loss function $\mathcal{L}(\cdot, \cdot)$ and the penalty function $\Omega_{\mathscr{H}}(\cdot)$ have desirable mathematical and statistical properties like convexity and differentiability and boundedness to allow the search of the function space $\mathscr{H}$ to be performed by optimization, $\widehat{f}_{\mathscr{H},\hat{\lambda}^{(\mathrm{opt})},n}$, thanks to the cross validation mechanism, provides a practical framework for potentially selecting the optimal predictive member of $\mathscr{H}$. It is important to note that finding $\widehat{f}_{\mathscr{H},\hat{\lambda}^{(\mathrm{opt})},n} \in \mathscr{H}$ does not in any way guarantee that the true risk $R(\widehat{f}_{\mathscr{H},\hat{\lambda}^{(\mathrm{opt})},n})$ on $\widehat{f}_{\mathscr{H},\hat{\lambda}^{(\mathrm{opt})},n}$ is close to $R^\star = R(f^\star)$. In other words, $\widehat{f}_{\mathscr{H},\hat{\lambda}^{(\mathrm{opt})},n}$

is the best in $\mathscr{H}$, but there is no guarantee that it is anywhere near $f^\star$. $\widehat{f}_{\mathscr{H},\widehat{\lambda}^{(\text{opt})},n}$ is what we refer to here as the *intraspace* optimal predictive model, since it is the cross validated best estimator within the function space $\mathscr{H}$.

Logistic regression is arguably one of the most widely used statistical learning machines, even enjoying a direct and strong relationship with artificial neural networks. Using the traditional $\{0,1\}$ labelling on the response variable Y, we have $\mathbb{P}[Y_i = 1|\mathbf{x}_i, \boldsymbol{\theta}_\gamma, M_\gamma] = \pi(\mathbf{x}_i;\boldsymbol{\theta}_\gamma, M_\gamma) = \pi(\mathbf{x}_i;\boldsymbol{\theta}_\gamma) = \pi_i(\boldsymbol{\theta}_\gamma) = \frac{1}{1+e^{-\mathbf{x}_i^\top \boldsymbol{\theta}_\gamma}}$. The likelihood function is

$$L(\boldsymbol{\theta}_\gamma; M_\gamma, \mathscr{D}_n) = \prod_{i=1}^{n} \Big\{ [\pi_i(\boldsymbol{\theta}_\gamma)]^{y_i} [1-\pi_i(\boldsymbol{\theta}_\gamma)]^{1-y_i} \Big\}. \quad (42)$$

The corresponding regularized empirical risk for the binary multiple linear logistic regression model is given by

$$\widehat{R}_\lambda(\boldsymbol{\theta}_\gamma, M_\gamma) = -\log L(\boldsymbol{\theta}_\gamma; M_\gamma, \mathscr{D}_n) + \lambda \|\boldsymbol{\theta}_\gamma\|_{\mathscr{H}}. \quad (43)$$

Now, the celebrated support vector machine [19] for binary classification with response variable taking values in $\{-1,+1\}$ is a solution to the regularized empirical hinge risk functional, namely,

$$\widehat{\mathbf{w}} = \underset{\mathbf{w}\in\mathscr{F}}{\operatorname{argmin}} \left\{ \frac{1}{n}\sum_{i=1}^{n} (1 - y_i \langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle)_+ + \frac{\lambda}{2}\|\mathbf{w}\|_{\mathscr{H}} \right\}.$$

Using quadratic programming on the dual formulation of this problem with α_i as the Lagrangian multipliers, we get $\widehat{\mathbf{w}} = \sum_{i=1}^{n} \widehat{\alpha}_i y_i \Phi(\mathbf{x}_i)$, and the corresponding estimated prediction function is

$$\widehat{f}_{\text{SVM}}(\mathbf{x}) = \texttt{sign}\left(\sum_{i=1}^{n} y_i \widehat{\alpha}_i \mathcal{K}(\mathbf{x}, \mathbf{x}_i) \right),$$

where the nonzero $\widehat{\alpha}_i$'s correspond to the so-called support vectors, and $\mathcal{K}(\mathbf{x}, \mathbf{x}_i) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}_i)\rangle$ is an incarnation of the so-called kernel trick that makes SVM immensely practical. Here, $\mathcal{K}(\cdot,\cdot)$ is a bivariate function called *kernel* defined on $\mathscr{X}\times\mathscr{X}$, used to measure the similarity between two points in an observation space. One of the most commonly used kernels in statistical machine learning is the Gaussian radial basis function kernel given by

$$\mathcal{K}(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\frac{1}{2}\frac{\|\mathbf{x}-\mathbf{x}_i\|_2^2}{\tau^2} \right).$$

There are many other kernels and kernel methods like Gaussian processes [5, 6].

Computational Model Selection

Before $\widehat{f}_{\mathscr{H},n}$ can be deemed good from a predictive perspective, its complexity must be controlled in order to endow it with good generalization properties, i.e., small prediction error on out-of-sample observations. This focus on the "generalizability" of $\widehat{f}_{\mathscr{H},n}$ is incredibly central to statistical learning when optimal prediction is the primary goal. Let $\mathscr{D}_n = \left\{ Z_1, Z_2, \ldots, Z_n \overset{\text{iid}}{\sim} p_Z(\mathbf{z}) \right\}$ where $Z_i = (X_i, Y_i) \in \mathscr{X}\times\mathscr{Y}$. Consider random splits of $\mathscr{D}_n$ into a training and a test set such that $\mathscr{D}_n = \mathscr{D}_{\text{tr}} \cup \mathscr{D}_{\text{te}}$ such that $n = |\mathscr{D}_{\text{tr}}| + |\mathscr{D}_{\text{te}}|$. Consider mappings $f : \mathscr{X} \longrightarrow \mathscr{Y}$ and a loss function $\mathcal{L}(\cdot,\cdot)$. Then the training and test errors are given by

$$\widehat{R}_{\text{tr}}(f) = \frac{1}{|\mathscr{D}_{\text{tr}}|}\sum_{i=1}^{n} \mathcal{L}(Y_i, f(X_i)) \mathbb{1}(Z_i \in \mathscr{D}_{\text{tr}}) \quad (44)$$

and

$$\widehat{R}_{\text{te}}(f) = \frac{1}{|\mathscr{D}_{\text{te}}|}\sum_{j=1}^{n} \mathcal{L}(Y_j, f(X_j)) \mathbb{1}(Z_j \in \mathscr{D}_{\text{te}}). \quad (45)$$

If $\widehat{f} = \underset{f\in\mathcal{H}}{\text{arginf}}\big\{\widehat{R}_{\text{tr}}(f)\big\}$, then $\mathbb{E}(\widehat{R}_{\text{tr}}(\widehat{f})) \leq \mathbb{E}(\widehat{R}_{\text{te}}(\widehat{f}))$. The so-called *optimism of the training error* is given by $\texttt{Optimism}(\widehat{R}_{\text{te}}(\widehat{f})) = \mathbb{E}(\widehat{R}_{\text{te}}(\widehat{f})) - \mathbb{E}(\widehat{R}_{\text{tr}}(\widehat{f}))$ and represents the amount by which the training error (empirical risk) underestimates (hence the term optimism) the test error (generalization error). Indeed, when the function is made more and more complex, the empirical risk gets lower and lower and farther from the true error, as seen in Figure 3. This is an instance of bias-variance dilemma that happens to be at the heart of methodological, theoretical, practical, computational, and epistemological aspects of statistical machine learning. The result of (3) highlights the reason why (17), the minimizer of the empirical risk, does not possess the predictive power needed, in the sense that it does not generalize well. In our quest for optimal predictive models, we will therefore not rely on the empirical risk alone, but instead will resort to score functions with inherent built-in mechanisms for selecting models that generalize well, i.e., produce lower prediction errors. Practically speaking, if the data $\mathscr{D}_n$ is randomly split S times, so that for each replication $\mathtt{s}$, the randomly shuffled (permuted) version $\mathscr{D}_n^{(\mathtt{s})}$ admits the decomposition $\mathscr{D}_n^{(\mathtt{s})} = \mathscr{D}_{\text{tr}}^{(\mathtt{s})} \cup \mathscr{D}_{\text{te}}^{(\mathtt{s})}$, then the $\mathtt{s}$th replication of the test error is given by

$$\begin{aligned} \mathrm{e}_{\text{te}}^{(\mathtt{s})} &= \texttt{te}(\widehat{f}^{(\mathscr{D}_{\text{tr}}^{(\mathtt{s})})}) \\ &= \frac{1}{|\mathscr{D}_{\text{te}}^{(\mathtt{s})}|}\sum_{i=1}^{n} \mathbb{1}(\mathbf{z}_i^{(\mathtt{s})} \in \mathscr{D}_{\text{te}}^{(\mathtt{s})}) \mathcal{L}(\mathrm{y}_i^{(\mathtt{s})}, \widehat{f}^{(\mathscr{D}_{\text{tr}}^{(\mathtt{s})})}(\mathbf{x}_i^{(\mathtt{s})})), \end{aligned} \quad (46)$$

where $\widehat{f}^{(\mathscr{D}_{\text{tr}}^{(\mathtt{s})})}(\cdot)$ is the instance of $\widehat{f}$ obtained using the $\mathtt{s}$th random replication of the training set. Clearly, one has S realizations of the test error, and $\{\mathrm{e}_{\text{te}}^{(1)}, \ldots, \mathrm{e}_{\text{te}}^{(\mathtt{s})}, \ldots, \mathrm{e}_{\text{te}}^{(\mathtt{S})}\}$ can be regarded as a sample of size S from the distribution of the true test error. One of the quantities often computed from the S realizations of the test error is the corresponding average test error

$$\texttt{AVTE}(\widehat{f}) = \frac{1}{S}\sum_{\mathtt{s}=1}^{S} \texttt{te}(\widehat{f}^{(\mathscr{D}_{\text{tr}}^{(\mathtt{s})})}). \quad (47)$$

It is important to note that $\widehat{f}^{(\mathscr{D}_{\mathtt{tr}}^{(s)})}(\cdot)$ should be internally optimized using its own internal intraspace optimality search criterion (like cross validation). This assumption is made with the finality of making sure that the interspace model comparison operates on the best of each considered model space. Let $\mathcal{C}$ be a collection of models, ideally with each from a different function space or a different method of estimation (learning). For instance, $\mathcal{C} = \{\widehat{f}_{\mathtt{LDA}}, \widehat{f}_{\mathtt{SVM}}, \widehat{f}_{\mathtt{CART}}, \widehat{f}_{\mathtt{RF}}, \widehat{f}_{\mathtt{GPR}}, \widehat{f}_{\mathtt{kNN}}, \widehat{f}_{\mathtt{Boost}}, \widehat{f}_{\mathtt{Logit}}, \widehat{f}_{\mathtt{RDA}}\}$.

Algorithm 2: Stochastic Hold-Out for Generalization

Input: Training data $\mathscr{D}_n = \{\mathbf{z}_i = (\mathbf{x}_i, \mathrm{y}_i),\ i = 1, \ldots, n\}$, where $\mathbf{x}_i \in \mathscr{X}$ and $\mathrm{y}_i \in \mathscr{Y}$, and list of learning machines to be evaluated, sample size n, number of random splits S, number of learning machines M, Proportion $\tau \in (1/2, 1)$ of observations in training set

Output: Matrix $E = (E_{sm}) = \widehat{R}_{\mathtt{te}}(\hat{f}_m^{(s)})$ of test error values for several learning machines

for $s = 1$ *to* S **do**

Generate the sth random split of the data set $\mathscr{D}_n$ into training set $\mathscr{D}_{\mathtt{tr}}^{(s)}$ and test set $\mathscr{D}_{\mathtt{te}}^{(s)}$

Such that $\mathscr{D}_n = \mathscr{D}_{\mathtt{tr}}^{(s)} \cup \mathscr{D}_{\mathtt{te}}^{(s)}$ and $n = |\mathscr{D}| = \tau|\mathscr{D}_{\mathtt{tr}}^{(s)}| + (1-\tau)|\mathscr{D}_{\mathtt{te}}^{(s)}|$

for $m = 1$ *to* M **do**

Build and refine the mth learning machine $\hat{f}_m^{(\mathscr{D}_{\mathtt{tr}}^{(s)})}(\cdot)$ using $\mathscr{D}_{\mathtt{tr}}^{(s)}$

Compute predictions $\hat{f}_m^{(\mathscr{D}_{\mathtt{tr}}^{(s)})}(\mathbf{x}_i)$ for $\mathbf{z}_i \in \mathscr{D}_{\mathtt{te}}^{(s)}$

Compute the test error for the mth learning machine

$\hat{\varepsilon}_{sm} = \widehat{R}_{\mathtt{te}}(\hat{f}_m^{(s)})$

$= \frac{1}{|\mathscr{D}_{\mathtt{te}}^{(s)}|} \sum_{i=1}^{n} \mathbb{1}(\mathbf{z}_i \in \mathscr{D}_{\mathtt{te}}) \mathcal{L}(\mathrm{y}_i, \hat{f}_m^{(\mathscr{D}_{\mathtt{te}}^{(s)})}(\mathbf{x}_i))$

Given a data set $\mathscr{D}_n$ and a collection of potential function spaces like $\mathcal{C}$, one defines

$$\begin{aligned} E = (E_{sm}) &= \widehat{R}_{\mathtt{te}}(\hat{f}_m^{(s)}) = \mathtt{te}(\hat{f}_m^{(\mathscr{D}_{\mathtt{tr}}^{(s)})}) \\ &= \mathtt{Error\ of}\ \ \hat{f}_m^{(\mathscr{D}_{\mathtt{tr}}^{(s)})}(\cdot)\ \ \mathtt{on}\ \mathscr{D}_{\mathtt{te}}^{(s)}. \end{aligned}$$

Then one proceeds to generate the matrix $E_{\mathtt{te}}$ containing S realized values of the test error for each hypothesis space. For classification, $E_{\mathtt{te}} \in [0,1]^{S \times M}$, and for regression $E_{\mathtt{te}} \in \mathbb{R}_+^{S \times M}$. Once $E_{\mathtt{te}}$ is generated, an interspace predictive model comparison is performed. The practical empirical optimal predictive model is given by

$$\hat{f}^{(\mathtt{opt})} = \underset{\hat{f} \in \mathcal{C}}{\mathtt{argmin}} \left\{ \mathtt{AVTE}(\hat{f}) \right\}.$$

It important to note that the median can also be used in place of the mean. Besides, the replications allow various statistical analyses on the predictive performances of each function space. A typical way to explore empirical interspace model comparison is to generate comparative boxplots of the replicated test errors, which can be done using the stochastic hold-out scheme described in Algorithm 2. Figure 4 depicts the results for the famous Crabs leptograpsus benchmark data set, and Figure 5 does the same for the ionosphere data set, which is another benchmark data set. Both data sets can be obtained from R.

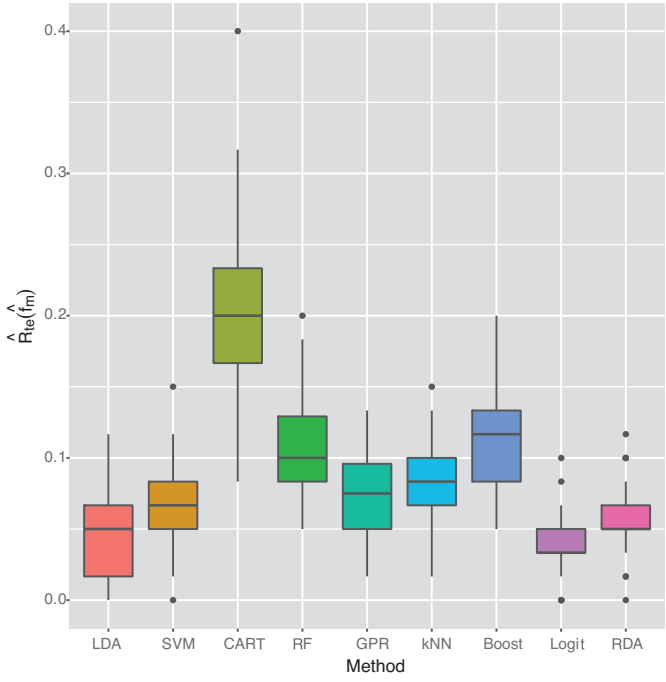


Figure 4. Predictive performances on the Crabs data.

As a matter of fact, each optimal classifier from a given space $\mathscr{H}$ will typically perform well if the data at hand and the generator from which it came somewhat accord with the properties of the space $\mathscr{H}$. This remark is probably what prompted the famous so-called *no free lunch theorem*, herein stated informally. (No Free Lunch). *There is no learning method that is universally superior to all other methods on all data sets. In other words, if a learning method is presented with a data set whose inherent patterns violate its assumptions, then that learning method will underperform.* Indeed, it is very humbling to see that some of the methods deemed somewhat simple sometimes hugely outperform the most sophisticated ones when compared on the basis of average out-of-sample (test) error.

Discussion and Conclusion

Modern data science and artificial intelligence greatly value the creation and construction of statistical learning

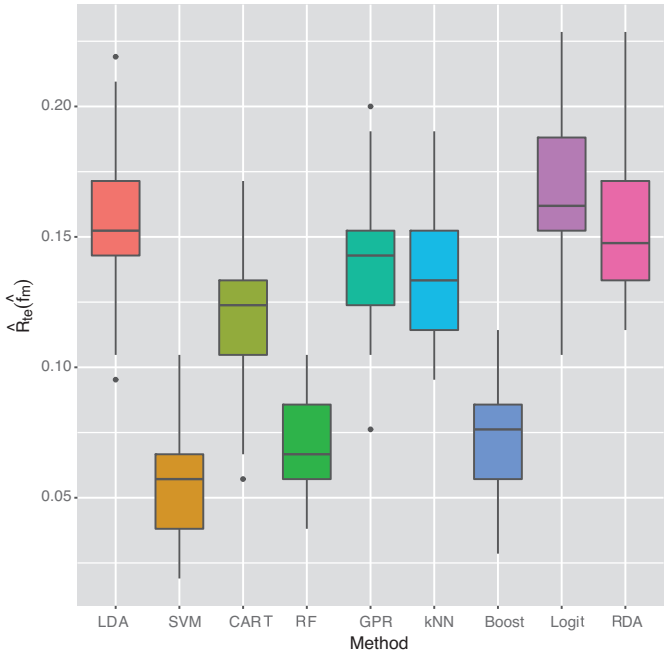


Figure 5. Predictive performances on the ionosphere data.

machines endowed with an inherent capability to predict accurately and precisely. In this paper, we have explored the niceties and subtleties of such a goal and have demonstrated that it requires a hefty dose of care and caution and definitely calls upon a solid theoretical understanding of learnability along with a lot of artlike practical common sense. Anyone who has done practical data science knows beyond a shadow of a doubt that data has a mind of its own, and tends to resist the temptation to seek a holy grail or a unified field, or any paradigm that works perfectly all the time. Practical data science almost always forces the practitioner to solve the problem at hand as thoroughly and as idiosyncratically as possible rather than seek a one-size-fits-all method that works well everywhere. At the heart of what we suggested throughout this paper is the theoretical result known as the no free lunch theorem, which reveals, both implicitly and explicitly, that the theoretical bounds studied extensively by experts do not really help much when it comes to practically selecting the optimal predictive model. Optimal predictive modelling is, and may always be, both a science and an art, requiring both mathematical and statistical rigor along with practical computational common sense.

References

[1] Akaike H. Information theory and an extension of the maximum likelihood principle. In: *Selected Papers of Hirotugu Akaike*. Springer New York, New York, NY; 1973:199–213. MR0483125

[2] Barbieri M and Berger JO. Optimal predictive model selection, *Ann. Statist.*, (32):870–897, 2004. MR2065192

[3] Breiman L. Bagging predictors, *Machine Learning*, (24):123–140, 1996.

[4] Breiman L. Random forests, *Machine Learning*, (45):5–32, 2001. MR3874153

[5] Clarke B, Fokoue E, Zhang H. *Principles and Theory for Data Mining and Machine Learning*, first edition, Springer Texts in Statistics, Springer-Verlag, 2009. MR2839778

[6] Csató L, Fokoué E, Opper M, Schottky B, Winther O. Efficient approaches to Gaussian process classification. In: Leen TK, Solla SA, Müller K-R, eds. *Advances in Neural Information Processing Systems*, number 12 of 12. MIT Press; 2000.

[7] Devroye L, Györfi L, Lugosi G. *A Probabilistic Theory of Pattern Recognition*, Stochastic Modelling and Applied Probability, Springer New York, 1997.

[8] Domingos P. A unified bias-variance decomposition for zero-one and squared loss, *AAAI/IAAI*, AAAI Press, 564–569, 2000.

[9] Fokoué E. Estimation of atom prevalence for optimal prediction. In: *Prediction and Discovery*, Contemporary Mathematics, vol. 443. American Mathematical Society; 2007:103–129. MR2433288

[10] Hoerl A and Kennard R. Ridge regression: biased estimation for non-orthogonal problems, *Technometrics*, (12):55–67, 1970.

[11] Hoeting JA, Madigan D, Raftery AE, Volinsky CT. Bayesian model averaging: A tutorial, *Statist. Sci.*, 14(4):382–417, 1999. MR1765176

[12] Kohavi R and Wolpert DH. Bias plus variance decomposition for zero-one loss functions. In: *Machine Learning, Proceedings of the Thirteenth International Conference (ICML '96), Bari, Italy, July 3–6, 1996*. 1996:275–283.

[13] Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain, *Psychological Review*, 65–386, 1958. MR0122606

[14] Schapire RE and Freund Y. *Boosting: Foundations and Algorithms*, The MIT Press, 2012. MR2920188

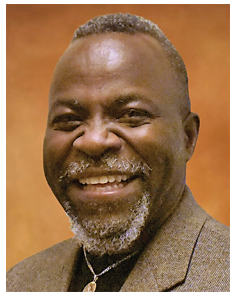
[15] Schwarz G. Estimating the dimension of a model, *The Ann. Statist.*, (6):461–464, 1978. MR0468014

[16] Stone M. An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion, *J. Roy. Statist. Soc. Ser. B (Methodological)*, 39(1):44–47, 1977. MR501454

[17] Stone M. Cross-validatory choice and assessment of statistical predictions, *J. Roy. Statist. Soc. Ser. B (Methodological)*, 111–147, 1974. MR356377

[18] Tibshirani R. Regression shrinkage and selection via the lasso, *J. Roy. Statist. Soc. Ser. B (Methodological)*, 58(1):267–288, 1996. MR1379242

[19] Vapnik VN. *The Nature of Statistical Learning Theory*, Springer, 2000. MR1719582



Ernest Fokoué

Credits

All figures are courtesy of the author. Author photo is courtesy of Rick Scoggins.