

Nonparametric Regression

Statistical Machine Learning, Spring 2014

Ryan Tibshirani (with Larry Wasserman)

1 Introduction, and k -nearest-neighbors

1.1 Basic setup, random inputs

- Given a random pair $(X, Y) \in \mathbb{R}^d \times \mathbb{R}$, the function

$$f_0(x) = \mathbb{E}(Y|X = x)$$

is called the regression function (of Y on X). The basic goal in nonparametric regression is to construct an estimate \hat{f} of f_0 , from i.i.d. samples $(x_1, y_1), \dots, (x_n, y_n) \in \mathbb{R}^d \times \mathbb{R}$ that have the same joint distribution as (X, Y) . We often call X the input, predictor, feature, etc., and Y the output, outcome, response, etc. Importantly, in nonparametric regression we do not assume a certain parametric form for f_0

- Note for i.i.d. samples $(x_1, y_1), \dots, (x_n, y_n)$, we can always write

$$y_i = f_0(x_i) + \epsilon_i, \quad i = 1, \dots, n,$$

where $\epsilon_1, \dots, \epsilon_n$ are i.i.d. random errors, with mean zero. Therefore we can think about the sampling distribution as follows: $(x_1, \epsilon_1), \dots, (x_n, \epsilon_n)$ are i.i.d. draws from some common joint distribution, where $\mathbb{E}(\epsilon_i) = 0$, and then y_1, \dots, y_n are generated from the above model

- In addition, we will assume that each ϵ_i is independent of x_i . This is actually quite a strong assumption, and you should think about it skeptically. (Why?) We make this assumption really for the sake of simplicity, and it should be noted that a lot of the theory that we cover (or at least, similar theory) also holds without the assumption of independence between the errors and the inputs

1.2 Basic setup, fixed inputs

- Another common setup in nonparametric regression is to directly assume a model

$$y_i = f_0(x_i) + \epsilon_i, \quad i = 1, \dots, n,$$

where now x_1, \dots, x_n are *fixed* inputs, and $\epsilon_1, \dots, \epsilon_n$ are still i.i.d. random errors with $\mathbb{E}(\epsilon_i) = 0$

- For arbitrary points x_1, \dots, x_n , this is really just the same as starting with the random input model, and conditioning on the particular values of x_1, \dots, x_n
- Generally speaking, nonparametric regression estimators are not defined with the random or fixed setups specifically in mind, i.e., there is no real distinction made here. A caveat: some estimators (like wavelets) do in fact assume evenly spaced fixed inputs, as in

$$x_i = i/n, \quad i = 1, \dots, n,$$

for evenly spaced inputs in the univariate case

- It is also important to mention that the theory is not completely the same between the random and fixed worlds, and some theoretical statements are sharper when assuming fixed input points, especially evenly spaced input points

1.3 What we cover here

- We won't be very precise about which setup we assume—random or fixed inputs—because, as mentioned before, it doesn't really matter when defining nonparametric regression estimators and discussing basic properties
- When it comes to theory, we will mix and match. The goal is to give you a flavor of some interesting results over a variety of methods, and under different assumptions. A few topics we will cover into more depth than others, but overall, this will not be a complete coverage. There are some excellent texts out there that you can consult for more details, proofs, etc., and some are listed below. There are surely others too, and you can always come ask one of us if you are looking for something in particular
 - Kernel smoothing, local polynomials
 - * Tsybakov (2009)
 - Regression splines, smoothing splines
 - * de Boor (1978)
 - * Green & Silverman (1994)
 - * Wahba (1990)
 - Reproducing kernel Hilbert spaces
 - * Scholkopf & Smola (2002)
 - * Wahba (1990)
 - Wavelets
 - * Johnstone (2011)
 - * Mallat (2008)
 - General references, more theoretical
 - * Györfi, Kohler, Krzyżak & Walk (2002)
 - * Wasserman (2006)
 - General references, more applied
 - * Simonoff (1996)
 - * Hastie, Tibshirani & Friedman (2008)
- After we discuss k -nearest-neighbors, our focus will be on the univariate case, $d = 1$, mainly for simplicity of presentation. At the end we'll briefly discuss the multivariate case

1.4 k -nearest-neighbors regression

- Here's a basic method to start us off: k -nearest-neighbors regression. We fix an integer $k \geq 1$ and define

$$\hat{f}(x) = \frac{1}{k} \sum_{i \in N_k(x)} y_i, \quad (1)$$

where $N_k(x)$ contains the indices of the k closest points of x_1, \dots, x_n to x

- This is not at all a bad estimator, and you will find it used in lots of applications, in many cases probably because of its simplicity. By varying the number of neighbors k , we can achieve a wide range of flexibility in the estimated function \hat{f} , with small k corresponding to a more flexible fit, and large k less flexible
- But it does have its limitations, an apparent one being that the fitted function \hat{f} essentially always looks jagged, especially for small or moderate k . Why is this? It helps to write

$$\hat{f}(x) = \sum_{i=1}^n w_i(x) y_i, \quad (2)$$

where the weights $w_i(x)$, $i = 1, \dots, n$ are defined as

$$w_i(x) = \begin{cases} 1/k & \text{if } x_i \text{ is one of the } k \text{ nearest points to } x \\ 0 & \text{else.} \end{cases}$$

Note that $w_i(x)$ is discontinuous as a function of x , and therefore so if $\hat{f}(x)$

- The representation (2) also reveals that the k -nearest-neighbors estimate is in a class of estimates we call *linear smoothers*, i.e., writing $y = (y_1, \dots, y_n) \in \mathbb{R}^n$, the vector of fitted values

$$\hat{y} = (\hat{f}(x_1), \dots, \hat{f}(x_n)) \in \mathbb{R}^n$$

can simply be expressed as $\hat{y} = Sy$. (To be clear, this means that for fixed inputs x_1, \dots, x_n , the vector of fitted values \hat{y} is a linear function of y ; it does not mean that $\hat{f}(x)$ need behave linearly as a function of x !) This class is quite large, and contains many popular estimators, as we'll see in the coming sections

- The k -nearest-neighbors estimator is consistent, under the random input model, provided we take $k = k_n$ such that $k_n \rightarrow \infty$ and $k_n/n \rightarrow 0$; e.g., $k = \sqrt{n}$ will do. See Section 6.2 of Györfi et al. (2002)
- Furthermore, assuming that the true regression function f_0 is Lipschitz continuous, the k -nearest-neighbors estimate further satisfies

$$\mathbb{E}(\hat{f}(X) - f(X))^2 = O(n^{-2/(2+d)}).$$

See Section 6.3 of Györfi et al. (2002)

2 Kernel smoothing, local polynomials

2.1 Kernel smoothing

- From here on, we assume $d = 1$ for simplicity, but we consider $d > 1$ in a separate section at the end. As in kernel density estimation, *kernel regression* or *kernel smoothing* begins with a kernel function $K : \mathbb{R} \rightarrow \mathbb{R}$, satisfying

$$\int K(x) dx = 1, \quad \int xK(x) dx = 0, \quad 0 < \int x^2 K(x) dx < \infty.$$

Two common examples are the Gaussian kernel:

$$K(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2),$$

and the Epanechnikov kernel

$$K(x) = \begin{cases} 3/4(1 - x^2) & \text{if } |x| \leq 1 \\ 0 & \text{else} \end{cases}$$

- Given a bandwidth $h > 0$, the (Nadaraya-Watson) kernel regression estimate is defined as

$$\hat{f}(x) = \frac{\sum_{i=1}^n K\left(\frac{x - x_i}{h}\right) y_i}{\sum_{i=1}^n K\left(\frac{x - x_i}{h}\right)}. \quad (3)$$

Hence kernel smoothing is also a linear smoother (2), with choice of weights $w_i(x) = K((x - x_i)/h) / \sum_{j=1}^n K((x - x_j)/h)$

- In comparison to the k -nearest-neighbors estimator in (1), which can be thought of as a raw (discontinuous) moving average of nearby outputs, the kernel estimator in (3) is a smooth moving average of outputs
- A shortcoming: the kernel regression suffers from poor bias at the boundaries of the domain of the inputs x_1, \dots, x_n . This happens because of the asymmetry of the kernel weights in such regions

2.2 Local polynomials

- We can alleviate this boundary bias issue by moving from a local constant fit to a local linear fit, or a local higher-order fit
- To build intuition, another way to view the kernel estimator in (3) is the following: at each input x , it employs the estimate $\hat{f}(x) = \hat{\theta}$, where $\hat{\theta}$ is the minimizer of

$$\sum_{i=1}^n K\left(\frac{x - x_i}{h}\right) (y_i - \theta)^2.$$

Instead we could consider forming the local estimate $\hat{f}(x) = \hat{\alpha} + \hat{\beta}x$, where $\hat{\alpha}, \hat{\beta}$ minimize

$$\sum_{i=1}^n K\left(\frac{x - x_i}{h}\right) (y_i - \alpha - \beta x_i)^2.$$

This is called *local linear regression*

- We can rewrite the local linear regression estimate $\hat{f}(x)$. This is just given by a weighted least squares fit, so

$$\hat{f}(x) = b(x)^T (B^T \Omega B)^{-1} B^T \Omega y,$$

where $b(x) = (1, x)$, B is an $n \times 2$ matrix with i th row $b(x_i) = (1, x_i)$, and Ω is an $n \times n$ diagonal matrix with i th diagonal element $K((x - x_i)/h)$. We can express this more concisely as $\hat{f}(x) = w(x)^T y$ for $w(x) = \Omega B (B^T \Omega B)^{-1} b(x)$, and so local linear regression is a linear smoother too

- The vector of fitted values $\hat{y} = (\hat{f}(x_1), \dots, \hat{f}(x_n))$ can be expressed as

$$\hat{y} = \begin{pmatrix} w_1(x)^T y \\ \vdots \\ w_n(x)^T y \end{pmatrix} = B(B^T \Omega B)^{-1} B^T \Omega y,$$

which should look familiar to you from weighted least squares

- Now we'll sketch how the local linear fit reduces the bias. Compute at a fixed point x ,

$$\mathbb{E}[\hat{f}(x)] = \sum_{i=1}^n w_i(x) f_0(x_i).$$

Using a Taylor expansion about x ,

$$\mathbb{E}[\hat{f}(x)] = f_0(x) \sum_{i=1}^n w_i(x) + f'_0(x) \sum_{i=1}^n (x_i - x) w_i(x) + R,$$

where the remainder term R contains quadratic and higher-order terms, and under regularity conditions, is small. One can check that in fact for the local linear regression estimator \hat{f} ,

$$\sum_{i=1}^n w_i(x) = 1 \quad \text{and} \quad \sum_{i=1}^n (x_i - x) w_i(x) = 0,$$

and so $\mathbb{E}[\hat{f}(x)] = f_0(x) + R$, which means that \hat{f} is unbiased to first order

- We don't have to stop with a local linear fit, we can more generally fit $\hat{f}(x) = \hat{\beta}_0 + \sum_{j=1}^p \hat{\beta}_j x^j$, where $\hat{\beta}_0, \dots, \hat{\beta}_p$ minimize

$$\sum_{i=1}^n K\left(\frac{x - x_i}{h}\right) \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_i^j\right)^2.$$

This is called *local polynomial regression*

- Again we can express

$$\hat{f}(x) = b(x)(B^T \Omega B)^{-1} B^T \Omega y = w(x)^T y,$$

where $b(x) = (1, x, \dots, x^p)$, B is an $n \times (p+1)$ matrix with i th row $b(x_i) = (1, x_i, \dots, x_i^p)$, and Ω is as before. Hence again, local polynomial regression is a linear smoother

- The basic message is that a higher degree in the local polynomial fit can help reduce the bias, even in the interior of the domain of inputs, but (not surprisingly) this comes at the expense of an increase in variance

2.3 Error bounds

- Consider the Holder class of functions $\Sigma(k, L)$, for an integer $k \geq 1$ and constant $L > 0$, which contains the set of all $k-1$ times differentiable functions $f : \mathbb{R} \rightarrow \mathbb{R}$ whose $(k-1)$ st derivative is Lipschitz continuous:

$$|f^{(k-1)}(x) - f^{(k-1)}(z)| \leq L|x - z|, \quad \text{for any } x, z.$$

(Actually, the function class $\Sigma(\gamma, L)$ can also be defined for non-integral γ , but this is not really important.)

- Consider the fixed inputs model, with appropriate conditions on the spacings of the inputs $x_1, \dots, x_n \in [0, 1]$ (and the error distribution and choice of kernel K). Assuming that the true function f_0 is in the Holder class $\Sigma(k, L)$, a Taylor expansion shows that the local polynomial estimator \hat{f} of order $k - 1$ has bounded bias and variance,

$$|\mathbb{E}[\hat{f}(x)] - f_0(x)| \leq C_1 h^k, \quad \text{Var}(\hat{f}(x)) \leq \frac{C_2}{nh},$$

over all $x \in [0, 1]$. Hence taking $h = \Theta(n^{-1/(2k+1)})$, the L_2 error of \hat{f} has convergence rate

$$\mathbb{E}\|\hat{f} - f_0\|_{L_2}^2 = O(n^{-2k/(2k+1)}).$$

(Here $\|f\|_{L_2}^2 = \int_0^1 f(x)^2 dx$.) See Section 1.6.1 of Tsybakov (2009)

- How fast is this convergence rate? In fact, we can't broadly do better over the function class $\Sigma(k, L)$. If we assume fixed inputs evenly spaced over $[0, 1]$, $x_i = i/n$ for $i = 1, \dots, n$, and a mild condition on the error distribution, the minimax risk is

$$\min_{\hat{f}} \max_{f_0 \in \Sigma(k, L)} \mathbb{E}\|\hat{f} - f_0\|_{L_2}^2 = \Omega(n^{-2k/(2k+1)}),$$

where the minimum above is over all estimators \hat{f} . See Section 2.6.2 of Tsybakov (2009)

- Is this the end of the story? Not at all. We'll see that by widening our the scope of functions that we consider, local polynomials are far from optimal
- As an aside, why did we study the Holder class $\Sigma(k, L)$ anyway? Because it was pretty natural to assume that $f_0^{(k)}$ is Lipschitz after performing a k th order Taylor expansion to compute the bias and variance

3 Regression splines, smoothing splines

3.1 Splines

- Regression splines and smoothing splines are motivated from a different perspective than kernels and local polynomials; in the latter, we started off with a special kind of local averaging, and moved our way up to a higher-order local models. With regression splines and smoothing splines, we build up our estimate globally, from a set of select basis functions
- These basis functions, as you might guess, are *splines*. A k th order spline is a piecewise polynomial function of degree k , that is continuous and has continuous derivatives of orders $1, \dots, k - 1$, at its knot points
- Formally, a function $f : \mathbb{R} \rightarrow \mathbb{R}$ is a k th order spline with knot points at $t_1 < \dots < t_m$, if f is a polynomial of degree k on each of the intervals $(-\infty, t_1], [t_1, t_2], \dots, [t_m, \infty)$, and $f^{(j)}$ is continuous at t_1, \dots, t_m , for each $j = 0, 1, \dots, k - 1$
- Splines have some very special properties are have been a topic of interest among statisticians and mathematicians for a long time. See de Boor (1978) for an in-depth coverage
- A bit of statistical folklore: it is said that a cubic spline is so smooth, that one cannot detect the locations of its knots by eye!

- How can we parametrize the set of a splines with knots at t_1, \dots, t_m ? The most natural way is to use the *truncated power basis*, g_1, \dots, g_{m+k+1} , defined as

$$g_1(x) = 1, \quad g_2(x) = x, \quad \dots \quad g_{k+1}(x) = x^k, \\ g_{k+1+j}(x) = (x - t_j)_+^k, \quad j = 1, \dots, m.$$

(Here x_+ denotes the positive part of x , i.e., $x_+ = \max\{x, 0\}$.)

- While these basis functions are natural, a much better computational choice, both for speed and numerical accuracy, is the *B-spline* basis. This was a major development in spline theory and is now pretty much the standard in software; see de Boor (1978) for details

3.2 Regression splines

- A first idea: let's perform regression on a spline basis. In other words, given inputs x_1, \dots, x_n and outputs y_1, \dots, y_n , we consider fitting functions f that are k th order splines with knots at some chosen locations t_1, \dots, t_m . This means expressing f as

$$f(x) = \sum_{j=1}^m \beta_j g_j(x),$$

where $\beta_1, \dots, \beta_{m+k+1}$ are coefficients and g_1, \dots, g_{m+k+1} , are basis functions for order k splines over the knots t_1, \dots, t_m (e.g., the truncated power basis or B-spline basis)

- Letting $y = (y_1, \dots, y_n) \in \mathbb{R}^n$, and defining the basis matrix $G \in \mathbb{R}^{n \times (m+k+1)}$ by

$$G_{ij} = g_j(x_i), \quad i = 1, \dots, n, \quad j = 1, \dots, m+k+1,$$

we can just use linear regression to determine the optimal coefficients $\hat{\beta} = (\hat{\beta}_1, \dots, \hat{\beta}_{m+k+1})$,

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^{m+k+1}}{\operatorname{argmin}} \quad \|y - G\beta\|_2^2,$$

which then leaves us with the fitted *regression spline* $\hat{f}(x) = \sum_{j=1}^{m+k+1} \hat{\beta}_j g_j(x)$

- Of course we know that $\hat{\beta} = (G^T G)^{-1} G^T y$, so the fitted values $\hat{y} = (\hat{f}(x_1), \dots, \hat{f}(x_n))$ are

$$\hat{y} = G(G^T G)^{-1} G^T y,$$

and regression splines are linear smoothers

- This is a classic method, and can work well provided we choose good knots t_1, \dots, t_m ; but in general choosing knots is a tricky business. There is a large literature on knot selection for regression splines via greedy methods like recursive partitioning

3.3 Natural splines

- A problem with regression splines is that the estimates tend to display erratic behavior, i.e., they have high variance, at the boundaries of the input domain. (This is the opposite problem to that we saw with kernel smoothing, which had poor bias at the boundaries.) This only gets worse with higher order regression splines
- A way to remedy this problem is to force the piecewise function to be of lower degree to the left of the leftmost knot, and to the right of the rightmost knot—this is exactly what *natural splines* do. A natural spline of order k , with knots at $t_1 < \dots < t_m$, is a piecewise polynomial function f such that

- f reduces to a polynomial of degree k on each of $[t_1, t_2], \dots, [t_{m-1}, t_m]$,
- f reduces to a polynomial of degree $(k-1)/2$ on $(-\infty, t_1]$ and $[t_m, \infty)$,
- f is continuous and has continuous derivatives of orders $1, \dots, k-1$ at its knots t_1, \dots, t_m .

It is implicit here that natural splines are only defined for odd orders k

- What is the dimension of the span of k th order natural splines with knots at t_1, \dots, t_m ? Recall for splines, this was $m + k + 1$ (the number of truncated power basis functions). We can compute this dimension by counting:

$$\underbrace{(k+1) \cdot (m-1)}_a + \underbrace{\left(\frac{(k-1)}{2} + 1\right) \cdot 2}_b - \underbrace{k \cdot m}_c = m.$$

Above, a is the number of degrees of freedom in the interior intervals $[t_1, t_2], \dots, [t_{m-1}, t_m]$, b is the number of degrees of freedom in the exterior intervals $(-\infty, t_1], [t_m, \infty)$, and c is the number of constraints at the knots t_1, \dots, t_m . The fact that the total dimension is m is amazing; this is independent of k !

- Note that there is a variant of the truncated power basis for natural splines, and a variant of the B-splines basis for natural splines. Again, B-splines are the preferred parametrization for computational speed and stability
- Natural splines of cubic order is the most common special case: these are smooth piecewise cubic functions, that are simply linear beyond the leftmost and rightmost knots

3.4 Smoothing splines

- Smoothing splines are an interesting creature: at the end of the day, these estimators perform a regularized regression over the natural spline basis, placing knots at all inputs x_1, \dots, x_n . Smoothing splines circumvent the problem of knot selection (as they just use the inputs as knots), and simultaneously, they control for overfitting by shrinking the coefficients of the estimated function (in its basis expansion)
- What makes them even more interesting is that they can be alternatively motivated directly from a functional minimization perspective. With inputs x_1, \dots, x_n contained in an interval $[a, b]$, the *smoothing spline* estimate \hat{f} of a given order k is defined as

$$\hat{f} = \operatorname{argmin}_f \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \int_a^b (f^{((k+1)/2)}(x))^2 dx. \quad (4)$$

This is an infinite-dimensional optimization problem over all functions f for the which the criterion is finite. This criterion trades off the least squares error of f over the observed pairs (x_i, y_i) , $i = 1, \dots, n$, with a penalty term that is large when the $((k+1)/2)$ nd derivative of f is wiggly. The tuning parameter $\lambda \geq 0$ governs the strength of each term in the minimization

- By far the most commonly considered case is $k = 3$, i.e., cubic smoothing splines, which are defined as

$$\hat{f} = \operatorname{argmin}_f \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \int_a^b (f''(x))^2 dx \quad (5)$$

- Remarkably, it so happens that the minimizer in the general k th order smoothing spline problem (4) is unique, and is a natural cubic spline with knots at the input points x_1, \dots, x_n ! Here we follow a proof for the cubic case of Green & Silverman (1994) (see also Exercise 5.7 in Hastie et al. (2008))

The key result can be stated as follows: if \tilde{f} is any twice differentiable function on $[a, b]$, and $x_1, \dots, x_n \in [a, b]$, then there exists a natural cubic spline f with knots at x_1, \dots, x_n such that $f(x_i) = \tilde{f}(x_i)$, $i = 1, \dots, n$ and

$$\int_a^b f''(x)^2 dx \leq \int_a^b \tilde{f}''(x)^2 dx.$$

Note that this would in fact prove that we can restrict our attention in (5) to natural splines with knots at x_1, \dots, x_n

Proof: the natural spline basis with knots at x_1, \dots, x_n is n -dimensional, so given any n points $z_i = \tilde{f}(x_i)$, $i = 1, \dots, n$, we can always find a natural spline f with knots at x_1, \dots, x_n that satisfies $f(x_i) = z_i$, $i = 1, \dots, n$. Now define

$$h(x) = \tilde{f}(x) - f(x).$$

Consider

$$\begin{aligned} \int_a^b f''(x)h''(x) dx &= \sum_a^b f''(x)h'(x) \Big|_a^b - \int_a^b f'''(x)h'(x) dx \\ &= \int_{x_1}^{x_n} f'''(x)h'(x) dx \\ &= \sum_{j=1}^{n-1} f'''(x)h(x) \Big|_{x_j}^{x_{j+1}} + \int_{x_1}^{x_n} f^{(4)}(x)h'(x) dx \\ &= - \sum_{j=1}^{n-1} f'''(x_j^+)(h(x_{j+1}) - h(x_j)), \end{aligned}$$

where in the first line we used integration by parts; in the second we used the that $f''(a) = f''(b) = 0$, and $f'''(x) = 0$ for $x \leq x_1$ and $x \geq x_n$, as f is a natural spline; in the third we used integration by parts again; in the fourth line we used the fact that f''' is constant on any open interval (x_j, x_{j+1}) , $j = 1, \dots, n-1$, and that $f^{(4)} = 0$, again because f is a natural spline. (In the above, we use $f'''(u^+)$ to denote $\lim_{x \downarrow u} f'''(x)$.) Finally, since $h(x_j) = 0$ for all $j = 1, \dots, n$, we have

$$\int_a^b f''(x)h''(x) dx = 0.$$

From this, it follows that

$$\begin{aligned} \int_a^b \tilde{f}''(x)^2 dx &= \int_a^b (f''(x) + h''(x))^2 dx \\ &= \int_a^b f''(x)^2 dx + \int_a^b h''(x)^2 dx + 2 \int_a^b f''(x)h''(x) dx \\ &= \int_a^b f''(x)^2 dx + \int_a^b h''(x)^2 dx, \end{aligned}$$

and therefore

$$\int_a^b f''(x)^2 dx \leq \int_a^b \tilde{f}''(x)^2 dx, \tag{6}$$

with equality if and only if $h''(x) = 0$ for all $x \in [a, b]$. Note that $h'' = 0$ implies that h must be linear, and since we already know that $h(x_j) = 0$ for all $j = 1, \dots, n$, this is equivalent to $h = 0$. In other words, the inequality (6) holds strictly except when $\tilde{f} = f$, so the solution in (5) is uniquely a natural spline with knots at the inputs

3.5 Finite-dimensional form

- The key result presented above tells us that we can choose a basis η_1, \dots, η_n for the set of k th order natural splines with knots over x_1, \dots, x_n , and reparametrize the problem (4) as

$$\hat{\beta} = \operatorname{argmin}_{\beta \in \mathbb{R}^n} \sum_{i=1}^n \left(y_i - \sum_{j=1}^n \beta_j \eta_j(x_i) \right)^2 + \lambda \int_a^b \left(\sum_{j=1}^n \beta_j \eta_j^{((k+1)/2)}(x) \right)^2 dx. \quad (7)$$

This is already a finite-dimensional problem, and once we solve for the coefficients $\hat{\beta} \in \mathbb{R}^n$, we know that the smoothing spline estimate is simply $\hat{f}(x) = \sum_{j=1}^n \hat{\beta}_j \eta_j(x)$

- Defining the basis matrix and penalty matrices $N, \Omega \in \mathbb{R}^{n \times n}$ by

$$N_{ij} = \eta_j(x_i) \quad \text{and} \quad \Omega_{ij} = \int_0^1 \eta_i^{((k+1)/2)}(t) \eta_j^{((k+1)/2)}(t) dt \quad \text{for } i, j = 1, \dots, n,$$

the problem in (7) can be written more succinctly as

$$\hat{\beta} = \operatorname{argmin}_{\beta \in \mathbb{R}^n} \|y - N\beta\|_2^2 + \lambda \beta^T \Omega \beta, \quad (8)$$

which shows the smoothing spline problem to be a type of generalized ridge regression problem. In fact, the solution in (8) has the explicit form

$$\hat{\beta} = (N^T N + \lambda \Omega)^{-1} N^T y,$$

and therefore the fitted values $\hat{y} = (\hat{f}(x_1), \dots, \hat{f}(x_n))$ are

$$\hat{y} = N(N^T N + \lambda \Omega)^{-1} N^T y. \quad (9)$$

Therefore, once again, smoothing splines are a kind of linear smoother

- A special property of smoothing splines: the fitted values in (9) can be computed in $O(n)$ operations. This is achieved by forming N from the B-spline basis (for natural splines), and in this case the matrix $N^T N + \Omega I$ ends up being banded (with a bandwidth that only depends on the polynomial order k). In practice, smoothing spline computations are extremely fast

3.6 Reinsch form

- It is informative to rewrite the fitted values in (9) in what is called Reinsch form,

$$\begin{aligned} \hat{y} &= N(N^T N + \lambda \Omega)^{-1} N^T y \\ &= N \left(N^T (I + \lambda (N^T)^{-1} \Omega N^{-1}) N \right)^{-1} N^T y \\ &= (I + \lambda K)^{-1} y, \end{aligned} \quad (10)$$

where $K = (N^T)^{-1} \Omega N^{-1}$

- Note that this matrix K does not depend on λ . If we compute an eigendecomposition $K = UDU^T$, then the eigendecomposition of $S = N(N^T N + \lambda\Omega)^{-1} = (I + \lambda K)^{-1}$ is

$$S = \sum_{j=1}^n \frac{1}{1 + \lambda d_j} u_j u_j^T,$$

where $D = \text{diag}(d_1, \dots, d_n)$

- Therefore the smoothing spline fitted values are $\hat{y} = Sy$, i.e.,

$$\hat{y} = \sum_{j=1}^n \frac{u_j^T y}{1 + \lambda d_j} u_j.$$

An interpretation: smoothing splines perform a regression on the orthonormal basis $u_1, \dots, u_n \in \mathbb{R}^n$, yet they shrink the coefficients in this regression, with more shrinkage assigned to eigenvectors u_j that correspond to large eigenvalues d_j

- So what exactly are these basis vectors u_1, \dots, u_n ? These are known as the *Demmler-Reinsch basis*, and a lot of their properties can be worked out analytically (Demmler & Reinsch 1975). Basically: the eigenvectors u_j that correspond to smaller eigenvalues d_j are smoother, and so with smoothing splines, we shrink less in their direction. Said differently, by increasing λ in the smoothing spline estimator, we are tuning out the more wiggly components

3.7 Kernel smoothing equivalence

- It turns out that the cubic smoothing spline estimator is in some sense asymptotically equivalent to a kernel regression estimator, with an unusual choice of kernel. Recall that both are linear smoothers; this equivalence is achieved by showing that under some conditions the smoothing spline weights converge to kernel weights, under the kernel

$$K(x) = \frac{1}{2} \exp(-|x|/\sqrt{2}) \sin(|x|/\sqrt{2} + \pi/4).$$

(The asymptotic weights actually also involve the local density of input points.) See Silverman (1984)

4 Mercer kernels, RKHS

- Smoothing splines are just one example of an estimator of the form

$$\hat{f} = \underset{f \in \mathcal{H}}{\text{argmin}} \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda J(f), \quad (11)$$

where \mathcal{H} is a space of functions, and J is a penalty functional

- Another important subclass of this problem form: we choose the function space $\mathcal{H} = \mathcal{H}_K$ to be what is called a *reproducing kernel Hilbert space*, or RKHS, associated with a particular kernel function $K : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$. To avoid confusion: this is not the same thing as a smoothing kernel! We'll adopt the convention of calling this second kind of kernel, i.e., the kind used in RKHS theory, a *Mercer kernel*, to differentiate the two

- There is an immense literature on the RKHS framework; here we follow the RKHS treatment in Chapter 5 of Hastie et al. (2008). Suppose that K is a positive definite kernel; examples include the polynomial kernel:

$$K(x, y) = (xy + 1)^k,$$

and the Gaussian radial basis kernel:

$$K(x, y) = \exp(-\delta(x - y)^2).$$

For any positive definite kernel function K , we have an eigenexpansion of the form

$$K(x, y) = \sum_{i=1}^{\infty} \gamma_i \phi_i(x) \phi_i(y),$$

for some eigenfunctions $\phi_i(x)$, $i = 1, 2, \dots$ and eigenvalues $\gamma_i \geq 0$, $i = 1, 2, \dots$, satisfying $\sum_{i=1}^{\infty} \gamma_i^2 < \infty$. We then define \mathcal{H}_K , the RKHS, as the space of functions generated by $K(\cdot, y)$, $y \in \mathbb{R}$, i.e., elements in \mathcal{H}_K are of the form

$$f(x) = \sum_{m \in M} \alpha_m K(x, y_m),$$

for a (possibly infinite) set M

- The above eigenexpansion of K implies that elements $f \in \mathcal{H}_K$ can be represented as

$$f(x) = \sum_{i=1}^{\infty} c_i \phi_i(x),$$

subject to the constraint that we must have $\sum_{i=1}^{\infty} c_i^2 / \gamma_i < \infty$. In fact, this representation is used to define a norm $\|\cdot\|_{\mathcal{H}_K}$ on \mathcal{H}_K : we define

$$\|f\|_{\mathcal{H}_K}^2 = \sum_{i=1}^{\infty} c_i^2 / \gamma_i.$$

- The natural choice now is to take the penalty functional in (11) as this squared RKHS norm, $J(f) = \|f\|_{\mathcal{H}_K}^2$. This yields the RKHS problem

$$\hat{f} = \operatorname{argmin}_{f \in \mathcal{H}_K} \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{H}_K}^2. \quad (12)$$

A remarkable achievement of RKHS theory is that the infinite-dimensional problem (12) can be reduced to a finite-dimensional one (as was the case with smoothing splines). This is called the *representer theorem* and is attributed to Kimeldorf & Wahba (1970). In particular, this result tells us that the minimum in (12) is uniquely attained by a function of the form

$$f(x) = \sum_{i=1}^n \alpha_i K(x, x_i),$$

or in other words, a function f lying in the span of the functions $K(\cdot, x_i)$, $i = 1, \dots, n$. Furthermore, we can rewrite the problem (12) in finite-dimensional form, as

$$\hat{\alpha} = \operatorname{argmin}_{\alpha \in \mathbb{R}^n} \|y - K\alpha\|_2^2 + \lambda \alpha^T K \alpha, \quad (13)$$

where $K \in \mathbb{R}^{n \times n}$ is a symmetric matrix defined by $K_{ij} = K(x_i, x_j)$ for $i, j = 1, \dots, n$. Once we have computed the optimal coefficients $\hat{\alpha}$ in (13), the estimated function \hat{f} in (12) is given by

$$\hat{f}(x) = \sum_{i=1}^n \hat{\alpha}_i K(x, x_i)$$

- The solution in (13) is

$$\hat{\alpha} = (K + \lambda I)^{-1} y,$$

so the fitted values $\hat{y} = (\hat{f}(x_1), \dots, \hat{f}(x_n))$ are

$$\hat{y} = K(K + \lambda I)^{-1} y = (I + \lambda K^{-1})^{-1} y,$$

showing that the RKHS estimator is yet again a linear smoother

- Proof of the equivalence between (12) and (13): we follow Wahba (1990) (see also Exercise 5.15 of Hastie et al. (2008)). This should also give you an idea of some of the basic properties of Mercer kernels. We define the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}_K}$ on \mathcal{H}_K , between functions $f = \sum_{i=1}^{\infty} c_i \phi_i$ and $g = \sum_{i=1}^{\infty} d_i \phi_i$, as

$$\langle f, g \rangle_{\mathcal{H}_K} = \sum_{i=1}^{\infty} c_i d_i / \gamma_i.$$

The functions $K(\cdot, x_i)$, $i = 1, \dots, n$ are called the *representer of evaluation*, and two of their important properties are as follows:

- $\langle f, K(\cdot, x_i) \rangle_{\mathcal{H}_K} = f(x_i)$, for any function $f \in \mathcal{H}_K$
- $\|f\|_{\mathcal{H}_K}^2 = \sum_{i,j=1}^n \alpha_i \alpha_j K(x_i, x_j)$ for any function $f = \sum_{i=1}^n \alpha_i K(\cdot, x_i)$

Using these properties we can now show that (12) reduces to (13). Let $f = \sum_{i=1}^n \alpha_i K(\cdot, x_i)$, and consider defining a function $\tilde{f} = f + \rho$, where ρ is any function orthogonal to the subspace spanned by $K(\cdot, x_i)$, $i = 1, \dots, n$. Then

$$\begin{aligned} \tilde{f}(x_i) &= f(x_i) + \rho(x_i) \\ &= \langle f + \rho, K(\cdot, x_i) \rangle \\ &= \langle f, K(\cdot, x_i) \rangle \\ &= f(x_i), \end{aligned}$$

where the third line relies on the fact that ρ is orthogonal to the representer of evaluation. Therefore f and \tilde{f} attain the same squared loss term in (12), but for the penalty term we have

$$\|\tilde{f}\|_{\mathcal{H}_K}^2 = \|f + \rho\|_{\mathcal{H}_K}^2 = \|f\|_{\mathcal{H}_K}^2 + \|\rho\|_{\mathcal{H}_K}^2,$$

where we again used the fact that ρ is orthogonal to the representer of evaluation, and hence f . Therefore f achieves a smaller criterion value in (12) compared to \tilde{f} , with strict inequality unless $\rho = 0$. Plugging $f = \sum_{i=1}^n \alpha_i K(\cdot, x_i)$ into (12) and expanding then gives the result

5 Linear smoothers

5.1 Degrees of freedom

- Literally every estimator we have discussed so far, when trained on (x_i, y_i) , $i = 1, \dots, n$, produces fitted values $\hat{y} = (\hat{f}(x_1), \dots, \hat{f}(x_n))$ of the form

$$\hat{y} = S y$$

for some matrix $S \in \mathbb{R}^{n \times n}$ depending on the inputs x_1, \dots, x_n —and also possibly on a tuning parameter such as h with kernel smoothing, or λ with smoothing splines—but not on y . Such estimators are called *linear smoothers*

- We define the *degrees of freedom* of \hat{y} as

$$\text{df}(\hat{y}) = \sum_{i=1}^n S_{ii} = \text{tr}(S),$$

the trace of the smooth matrix S . Note that degrees of freedom is really associated with the fitting procedure that was used to produce \hat{y} (not the particular realization of \hat{y} itself), and is intuitively seen as the effective number of parameters used by this fitting procedure. I.e., the higher the degrees of freedom, the more complex it is

- The above definition for linear smoothers is a special case of the broader definition

$$\text{df}(\hat{y}) = \frac{1}{\sigma^2} \sum_{i=1}^n \text{Cov}(\hat{y}_i, y_i),$$

where the covariance above is taken with respect to the model for fixed inputs x_1, \dots, x_n , and $\sigma^2 = \mathbb{E}(\epsilon_i^2)$ denotes the common noise variance. In words: degrees of freedom sums the covariances of each fitted value with the corresponding observation

- Example: for a regression spline estimator, of polynomial order k , with knots at the locations t_1, \dots, t_m , recall that $\hat{y} = G(G^T G^{-1})G^T y$ for $G \in \mathbb{R}^{n \times (m+k+1)}$ the order k spline basis matrix over the knots t_1, \dots, t_m . Therefore

$$\text{df}(\hat{y}) = \text{tr}(G(G^T G)^{-1}G^T) = \text{tr}(G^T G(G^T G)^{-1}) = m + k + 1,$$

i.e., the degrees of freedom of a regression spline estimator is the number of knots + the polynomial order + 1

- Example: for a smoothing spline estimator, recall that we were able to express the fitted values as $\hat{y} = (I + \lambda K)^{-1}$, i.e., as

$$\hat{y} = U(1 + \lambda D)^{-1}U^T y,$$

where UDU^T is the eigendecomposition of the Reinsch matrix $K = (N^T)^{-1}\Omega N^{-1}$ (which depended only on the input points x_1, \dots, x_n and the polynomial order k). A smoothing spline hence has degrees of freedom

$$\text{df}(\hat{y}) = \text{tr}(U(1 + \lambda D)^{-1}U^T) = \sum_{i=1}^n \frac{1}{1 + \lambda d_j},$$

where $D = \text{diag}(d_1, \dots, d_n)$. This is a monotone decreasing function in λ , with $\text{df}(\hat{y}) = 0$ when $\lambda = 0$, and $\text{df}(\hat{y}) \rightarrow (k - 1)/2$ when $\lambda \rightarrow \infty$, the number of zero eigenvalues among d_1, \dots, d_n

5.2 Optimism and unbiased risk estimation

- Why is degrees of freedom a useful concept? For one, it allows us to put two different procedures on equal footing. E.g., suppose we wanted to compare kernel smoothing versus smoothing splines; we could tune them to match their degrees of freedom, and then compare their performances

- A more mathematical motivation for degrees of freedom comes from looking at an expansion for the risk of \hat{y} . Again, assuming the model with fixed inputs and common error variance σ^2 , we consider the risk $\mathbb{E}\|\mu_0 - \hat{y}\|_2^2$, where $\mu_0 = (f_0(x_1), \dots, f_0(x_n))$ is the true underlying mean vector. We compute

$$\begin{aligned}\mathbb{E}\|\mu_0 - \hat{y}\|_2^2 &= \mathbb{E}\|\mu_0 - y + y - \hat{y}\|_2^2 \\ &= n\sigma^2 + \mathbb{E}\|y - \hat{y}\|_2^2 + 2\text{tr}(\text{Cov}(\mu_0 - y, y - \hat{y})) \\ &= n\sigma^2 + \mathbb{E}\|y - \hat{y}\|_2^2 + 2\sigma^2 \text{df}(\hat{y})\end{aligned}\tag{14}$$

- The formula (14) can be interpreted in two ways. The first: it relates the *optimism* of \hat{y} to its degrees of freedom, i.e.,

$$\mathbb{E}\|\mu_0 - \hat{y}\|_2^2 - \mathbb{E}\|y - \hat{y}\|_2^2 = n\sigma^2 + 2\sigma^2 \text{df}(\hat{y}),$$

where the left-hand side is the difference between the risk and the expected training error, or as we will call it, the optimism. Therefore, the more complex the fitting procedure, the higher its degrees of freedom, and the greater its optimism

- The second interpretation for the decomposition (14) that is provides an avenue for *unbiased risk estimation*. Note simply that

$$\hat{R} = n\sigma^2 + \|y - \hat{y}\|_2^2 + 2\sigma^2 \text{df}(\hat{y})\tag{15}$$

provides an unbiased estimate of the risk $R = \mathbb{E}\|\mu_0 - \hat{y}\|_2^2$. In the above, $\|y - \hat{y}\|_2^2$ is the achieved training error, and $\text{df}(\hat{y})$ is the degrees of freedom of our fitting procedure; the first is always observed, and the second is known exactly for linear smoothers. (However, even when $\text{df}(\hat{y})$ is not known exactly, we could replace it by its own unbiased estimate $\hat{\text{df}}(\hat{y})$, and this would still give us an unbiased estimate for the risk.)

- Aside from providing an absolute estimate of the risk, the estimate in (15) can also be used for the purposes of model selection. E.g., suppose that our fitting procedure depends on a tuning parameter θ , which we will write as $\hat{y}_\theta = S_\theta y$ for a linear smoother. Then we could choose the tuning parameter θ to minimize the estimated risk, as in

$$\hat{\theta} = \underset{\theta}{\text{argmin}} \|y - S_\theta y\|_2^2 + 2\sigma^2 \text{tr}(S_\theta).$$

This is just like the C_p criterion, or AIC, in regression; we could replace the factor of 2 above with $\log n$ to obtain something like BIC

5.3 Leave-one-out and generalized cross-validation

- A ubiquitous tool for error estimation and model selection is cross-validation. We could certainly use K -fold or leave-one-out cross-validation to obtain an estimate of the expected test error (which only differs from the risk by a constant amount, the irreducible error), or to perform model selection
- For linear smoothers $\hat{y} = (\hat{f}(x_1), \dots, \hat{f}(x_n)) = S y$, leave-one-out cross-validation is particularly appealing because in many cases we have the seemingly magical reduction

$$\frac{1}{n} \sum_{i=1}^n (y_i - \hat{f}^{-i}(x_i))^2 = \frac{1}{n} \sum_{i=1}^n \frac{(y_i - \hat{f}(x_i))^2}{1 - S_{ii}},\tag{16}$$

where \hat{f}^{-i} denotes the estimated function trained on all but the i th pair (x_i, y_i) . This provides an enormous computational savings because it shows that to compute leave-one-out cross-validation error, we don't have to actually ever compute \hat{f}^{-i} , $i = 1, \dots, n$

- Why does (16) hold, and for which linear smoothers $\hat{y} = Sy$? Just rearranging (16) perhaps demystifies this seemingly magical relationship and helps to answer these questions. Suppose we knew that \hat{f} had the property

$$\hat{f}^{-i}(x_i) = \frac{1}{1 - S_{ii}}(\hat{f}(x_i) - S_{ii}y_i). \quad (17)$$

That is, to obtain the estimate at x_i under the function \hat{f}^{-i} fit on all but (x_i, y_i) , we take the sum of the linear weights across all but the i th point, $\hat{f}(x_i) - S_{ii}y_i = \sum_{i \neq j} S_{ij}y_j$, and then renormalize so that these weights sum to 1

- This is not an unreasonable property; e.g., we can immediately convince ourselves that it holds for kernel smoothing. A little calculation shows that it also holds for smoothing splines (using the Sherman-Morrison update formula). How about for k -nearest-neighbors?
- From the special property (17), it is easy to show the leave-one-out formula (16). We have

$$y_i - \hat{f}^{-i}(x_i) = y_i - \frac{1}{1 - S_{ii}}(\hat{f}(x_i) - S_{ii}y_i) = \frac{y_i - \hat{f}(x_i)}{1 - S_{ii}},$$

and then squaring both sides and summing over n gives (16)

- Finally, *generalized cross-validation* is a small twist on the right-hand side in (16) that gives an approximation to leave-one-out cross-validation error. It is defined as by replacing the appearances of diagonal terms S_{ii} with the average diagonal term $\text{tr}(S)/n$,

$$\text{GCV}(\hat{f}) = \frac{1}{n} \sum_{i=1}^n \frac{(y_i - \hat{f}(x_i))^2}{1 - \text{tr}(S)/n}.$$

This can be of computational advantage in some cases where $\text{tr}(S)$ is easier to compute than individual elements S_{ii} , and is also closely tied to the unbiased risk estimate in (15), seen by making the approximation $1/(1 - x)^2 \approx 1 + 2x$

6 Wavelets and local adaptivity

6.1 Wavelet smoothing

- Not every nonparametric regression estimate needs to be a linear smoother (though this does seem to be incredibly common), and *wavelet smoothing* is one of the leading nonlinear tools for nonparametric estimation. The theory of wavelets is quite elegant and we only give a very terse introduction here; see Mallat (2008) for an excellent reference
- You can think of wavelets as defining an orthonormal function basis, with the basis functions exhibiting a highly varied level of smoothness. Importantly, these basis functions also display spatially localized smoothness at different locations in the input domain. There are actually many different choices for wavelets bases (Haar wavelets, symmlets, etc.), but these are details that we will not go into
- Consider these basis functions, ϕ_1, \dots, ϕ_n , evaluated over n equally spaced inputs in $[0, 1]$:

$$x_i = i/n, \quad i = 1, \dots, n.$$

(The assumption of evenly spaced inputs is crucial for fast computations; we also typically assume with wavelets that n is a power of 2.) We now form a wavelet basis matrix $W \in \mathbb{R}^{n \times n}$, defined by

$$W_{ij} = \phi_j(x_i), \quad i = 1, \dots, n$$

- The goal, given outputs $y = (y_1, \dots, y_n)$ over the evenly spaced input points, is to represent y as a sparse combination of the wavelet basis functions. To do so, we first perform a wavelet transform (multiply by W^T):

$$\theta = W^T y,$$

we threshold the coefficients θ :

$$\hat{\theta} = \mathcal{T}_\lambda(\theta),$$

and then perform an inverse wavelet transform (multiply by W):

$$\hat{y} = W\hat{\theta}$$

- The wavelet and inverse wavelet transforms (multiplication by W^T and W) each require $O(n)$ operations, and are practically extremely fast due to clever pyramidal multiplication schemes that exploit the special structure of wavelets
- The threshold function \mathcal{T}_λ is usually taken to be hard-thresholding, i.e.,

$$[\mathcal{T}_\lambda(z)]_i = z_i \cdot 1\{|z| \geq \lambda\}, \quad i = 1, \dots, n,$$

or soft-thresholding, i.e.,

$$[\mathcal{T}_\lambda(z)]_i = (z_i - \text{sign}(z_i)\lambda) \cdot 1\{|z| \geq \lambda\}, \quad i = 1, \dots, n.$$

These thresholding functions are both also $O(n)$, and computationally trivial, making wavelet smoothing very fast overall. It should be emphasized that wavelet smoothing is not a linear smoother, i.e., there is no matrix S here such that $\hat{y} = Sy$ for all y

- We can write the wavelet smoothing estimate in a more familiar form, following our previous discussions on basis functions and regularization. For hard-thresholding, we solve

$$\hat{\theta} = \underset{\theta \in \mathbb{R}^n}{\text{argmin}} \|y - W\theta\|_2^2 + \lambda^2 \|\theta\|_0,$$

and then the wavelet smoothing fitted values are $\hat{y} = W\hat{\theta}$. Here $\|\theta\|_0 = \sum_{i=1}^n 1\{\theta_i \neq 0\}$, the number of nonzero components of θ . For soft-thresholding, the corresponding optimization problem is

$$\hat{\theta} = \underset{\theta \in \mathbb{R}^n}{\text{argmin}} \|y - W\theta\|_2^2 + 2\lambda \|\theta\|_1,$$

and then the wavelet smoothing fitted values are again $\hat{y} = W\hat{\theta}$. Here $\|\theta\|_1 = \sum_{i=1}^n |\theta_i|$, the ℓ_1 norm

6.2 The strengths of wavelets, the limitations of linear smoothers

- Apart from its sheer computational speed, an important strength of wavelet smoothing is that it can represent a signal that has a spatially heterogeneous degree of smoothness, i.e., it can be both smooth and wiggly at different regions of the input domain. The reason that wavelet smoothing can achieve such local adaptivity is because it selects a sparse number of wavelet basis functions, by thresholding the coefficients from a basis regression
- Can a linear smoother do the same? We can appeal to convergence theory to give a precise answer to this question. We consider the function class $F(k, L)$ of all k times (weakly) differentiable functions, whose k th derivative satisfies $\text{TV}(f^{(k)}) \leq L$, with $\text{TV}(\cdot)$ being the total variation operator. A result of Donoho & Johnstone (1998) shows that, assuming $f_0 \in F(k, L)$

(and further conditions on the problem setup), the wavelet smoothing estimator with an appropriately tuned parameter λ converges at the rate $n^{-(2k+2)/(2k+3)}$. These authors also show that this is the minimax rate of convergence over $F(k, L)$

Now, it can be shown that $F(k, L) \supseteq \Sigma(k+1, L-1)$, where $\Sigma(k+1, L-1)$ is the order $k+1$ Holder class that we considered previously. From what we know about kernel smoothing, this estimator converges at the rate $n^{-(2(k+1))/(2(k+1)+1)} = n^{-(2k+2)/(2k+3)}$ over $\Sigma(k+1, L-1)$. But how about the larger class $F(k, L)$? Can kernel smoothing achieve the same (minimax) rate over this larger class? How about linear smoothers in general?

A remarkable result of Donoho & Johnstone (1998) shows that no linear smoother can attain the minimax rate over $F(k, L)$, and that a lower bound on the convergence rate achieved by linear smoothers is $n^{-(2k+1)/(2k+2)}$. There is actually a big difference in these rates:

$$\frac{n^{-(2k+1)/(2k+2)}}{n^{-(2k+2)/(2k+3)}} = n^{(k+1)/((2k+2)(2k+3))} \rightarrow \infty \quad \text{as } n \rightarrow \infty.$$

Practically, the performance of wavelets and linear smoothers in problems with spatially heterogeneous smoothness can be striking as well

- However, you should keep in mind that wavelets are not perfect: a major shortcoming is that they require a highly restrictive setup: recall that they require evenly spaced inputs, and n to be power of 2, and there are often further assumptions made about the behavior of the fitted function at the boundaries of the input domain
- Also, wavelets are not the end of the story when it comes to local adaptivity. So what are the alternatives? Both kernel smoothing and smoothing splines can be made to be more locally adaptive by allowing for a local bandwidth parameter or a local penalty parameter (but this can be difficult to implement in practice). Also, locally adaptive regression splines and trend filtering are two other nonparametric estimators that use only a single smoothing parameter, and still achieve the minimax rate of convergence over $F(k, L)$. See Mammen & van de Geer (1997) and Tibshirani (2013)

7 The multivariate case

7.1 The curse of dimensionality

- So far, we've looked exclusively at the univariate case, $d = 1$ (with d being the dimension of the inputs). In fact, nearly everything we've discussed so far has a multivariate counterpart: kernel smoothing very naturally extends to higher dimensions, using, e.g., $K(\|x - x_i\|_2/h)$ as the kernel weight between points $x, x_i \in \mathbb{R}^d$; smoothing splines can be generalized to multiple dimensions using *thin-plate splines*; reproducing kernel Hilbert spaces automatically extend just by starting with a Mercer kernel defined on $\mathbb{R}^d \times \mathbb{R}^d$, as in $K(x, y) = (x^T y + 1)^k$ for the multivariate polynomial kernel, and $K(x, y) = \exp(-\delta\|x - y\|_2^2)$ for the multivariate Gaussian radial basis kernel
- All of these multivariate extensions are interesting and can be very useful for producing rich nonparametric fits in low to moderate dimensions
- But in high dimensions the story is very different; if d is large compared to n , then “true” multivariate extensions such as these ones are problematic and suffer from poor variance. In rough terms, estimation gets exponentially harder as the number of dimensions increases, a phenomenon called the *curse of dimensionality*. (This term usually is attributed to Bellman (1962), who encountered an analogous issue but in a separate context—dynamic programming)

- This curse of dimensionality is echoed by the role of d in the minimax rates of convergence of nonparametric regression estimators, across various setups. E.g., recall the Holder class of functions $\Sigma(k, L)$ that we defined, in the univariate case, of functions whose $(k-1)$ st derivative is L -Lipschitz. In higher dimensions, the natural extension of this is the space $\Sigma_d(k, L)$ of functions on \mathbb{R}^d whose k th order partial derivatives are all L -Lipschitz. It can be shown that

$$\min_{\hat{f}} \max_{f_0 \in \Sigma_d(k, L)} \mathbb{E} \|\hat{f}(X) - f_0(X)\|_2^2 = \Omega(n^{-2k/(2k+d)}),$$

for the random input model, under mild conditions. See Section 3.2. of Györfi et al. (2002). What does this rate mean? For a small fixed ϵ , think about how large we need to make n so that $n^{-2k/(2k+d)} \leq \epsilon$; this is $n \geq \epsilon^{2k/(2k+d)}$. Now, as we increase d , we require exponentially more observations n .

7.2 Additive models

- *Additive models* finesse the curse of dimensionality by fitting an estimate that decomposes as a sum of univariate functions across dimensions. Instead of considering a full d -dimensional function of the form

$$f(x) = f(x_1, \dots, x_d), \quad (18)$$

we restrict our attention to functions of the form

$$f(x) = f_1(x_1) + \dots + f_d(x_d). \quad (19)$$

(Here the notation x_j denotes the j th component of $x \in \mathbb{R}^d$, slightly unusual notation, but used so as not to confuse with the labeling of the d -dimensional inputs x_1, \dots, x_n). As each function f_j , $j = 1, \dots, d$ is univariate, fitting an estimate of the additive form (19) is certainly less ambitious than fitting one of the form (18), but the additive model is still flexible enough to capture interesting (marginal) behavior in high dimensions

- The choice of modeler (19) need not be regarded as an assumption we make about the true function f_0 , just like we don't always assume that the true model is linear when using linear regression. In many cases, we fit an additive model because we think it may provide a useful approximation to the truth, and is able to scale well with the number of dimensions d
- Estimation with the additive model (19) is actually very simple; we just cycle through estimating each function f_j , $j = 1, \dots, d$ individually (like a block coordinate descent algorithm). For this, we need to choose a univariate smoother (i.e., nonparametric regression procedure), which we will write as \mathcal{S} , so that the fitted function from smoothing $y = (y_1, \dots, y_n) \in \mathbb{R}^n$ over the inputs $z = (z_1, \dots, z_n) \in \mathbb{R}^n$ can be written as

$$\hat{f} = \mathcal{S}(z, y).$$

E.g., it is common to choose \mathcal{S} to be the cubic smoothing spline smoother, where the tuning parameter λ selected by generalized cross-validation.

Once \mathcal{S} has been chosen, we initialize $\hat{f}_1, \dots, \hat{f}_d$, and repeat for $j = 1, \dots, d, 1, \dots, d, \dots$:

- define $z = (x_{1j}, \dots, x_{nj})$, and $\tilde{y}_i = y_i - \sum_{\ell \neq j} \hat{f}_\ell(x_{i\ell})$, $i = 1, \dots, n$;
- smooth $\hat{f}_j = \mathcal{S}(z, \tilde{y})$;
- center $\hat{f}_j = \hat{f}_j - \frac{1}{n} \sum_{i=1}^n \hat{f}_j(x_{ij})$.

This algorithm is known as *backfitting*. The second line in the update above is used to remove the mean from each function \hat{f}_j , $j = 1, \dots, d$, otherwise the model would not be identifiable. Our final estimate therefore takes the form

$$\hat{f}(x) = \hat{\alpha} + \hat{f}_1(x_{.1}) + \dots + \hat{f}_d(x_{.d}),$$

where $\hat{\alpha} = \frac{1}{n} \sum_{i=1}^n y_i$. See Hastie & Tibshirani (1990) for details on backfitting and additive models

References

- Bellman, R. (1962), *Adaptive Control Processes*, Princeton University Press, Princeton.
- de Boor, C. (1978), *A Practical Guide to Splines*, Springer, New York.
- Demmler, A. & Reinsch, C. (1975), ‘Oscillation matrices with spline smoothing’, *Numerische Mathematik* **24**(5), 375–382.
- Donoho, D. L. & Johnstone, I. (1998), ‘Minimax estimation via wavelet shrinkage’, *Annals of Statistics* **26**(8), 879–921.
- Green, P. & Silverman, B. (1994), *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach*, Chapman & Hall/CRC Press, Boca Raton.
- Gyorfi, L., Kohler, M., Krzyzak, A. & Walk, H. (2002), *A Distribution-Free Theory of Nonparametric Regression*, Springer, New York.
- Hastie, T. & Tibshirani, R. (1990), *Generalized Additive Models*, Chapman and Hall, London.
- Hastie, T., Tibshirani, R. & Friedman, J. (2008), *The Elements of Statistical Learning; Data Mining, Inference and Prediction*, Springer, New York. Second edition.
- Johnstone, I. (2011), *Gaussian estimation: Sequence and wavelet models*, Under contract to Cambridge University Press. Online version at <http://www-stat.stanford.edu/~imj>.
- Kimeldorf, G. & Wahba, G. (1970), ‘A correspondence between bayesian estimation on stochastic processes and smoothing by splines’, *Annals of Mathematical Statistics* **41**(2), 495–502.
- Mallat, S. (2008), *A wavelet tour of signal processing*, Academic Press, San Diego. Third edition.
- Mammen, E. & van de Geer, S. (1997), ‘Locally adaptive regression splines’, *Annals of Statistics* **25**(1), 387–413.
- Scholkopf, B. & Smola, A. (2002), ‘Learning with kernels’.
- Silverman, B. (1984), ‘Spline smoothing: the equivalent variable kernel method’, **12**(3), 898–916.
- Simonoff, J. (1996), *Smoothing Methods in Statistics*, Springer, New York.
- Tibshirani, R. J. (2013), Adaptive piecewise polynomial estimation via trend filtering. arXiv: 1304.2986.
- Tsybakov, A. (2009), *Introduction to Nonparametric Estimation*, Springer, New York.
- Wahba, G. (1990), *Spline Models for Observational Data*, Society for Industrial and Applied Mathematics, Philadelphia.
- Wasserman, L. (2006), *All of Nonparametric Statistics*, Springer, New York.