Nonparametric Regression

Statistical Machine Learning, Spring 2019
Ryan Tibshirani and Larry Wasserman

1 Introduction

1.1 Basic setup

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

$$m_0(x) = \mathbb{E}(Y|X = x)$$

is called the regression function (of Y on X). The basic goal in nonparametric regression: to construct a predictor of Y given X. This is basically the same as constructing an estimate $\widehat{m}$ of m_0, from i.i.d. samples $(X_i, Y_i) \in \mathbb{R}^d \times \mathbb{R}$, $i = 1, \ldots, n$. Given a new X, our prediction of Y is $\widehat{m}(X)$. We often call X the input, predictor, feature, etc., and Y the output, outcome, response, etc.

Note for i.i.d. samples $(X_i, Y_i) \in \mathbb{R}^d \times \mathbb{R}$, $i = 1, \ldots, n$, we can always write

$$Y_i = m_0(X_i) + \epsilon_i, \quad i = 1, \ldots, n,$$

where ϵ_i, $i = 1, \ldots, n$ are i.i.d. random errors, with mean zero. Therefore we can think about the sampling distribution as follows: (X_i, ϵ_i), $i = 1, \ldots, n$ are i.i.d. draws from some common joint distribution, where $\mathbb{E}(\epsilon_i) = 0$, and Y_i, $i = 1, \ldots, n$ are generated from the above model.

It is common to assume that each ϵ_i is independent of X_i. This is a very strong assumption, and you should think about it skeptically. We too will sometimes make this assumption, for simplicity. It should be noted that a good portion of theoretical results that we cover (or at least, similar theory) also holds without this assumption.

1.2 Fixed or random inputs?

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

$$Y_i = m_0(X_i) + \epsilon_i, \quad i = 1, \ldots, n,$$

where now X_i, $i = 1, \ldots, n$ are *fixed* inputs, and ϵ_i, $i = 1, \ldots, n$ are i.i.d. with $\mathbb{E}(\epsilon_i) = 0$.

For arbitrary X_i, $i = 1, \ldots, n$, this is really just the same as starting with the random input model, and conditioning on the particular values of X_i, $i = 1, \ldots, n$. (But note: after conditioning on the inputs, the errors are only i.i.d. if we assumed that the errors and inputs were independent in the first place.)

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

$$X_i = i/n, \quad i = 1, \ldots, n,$$

for evenly spaced inputs in the univariate case.

Theory is not completely the same between the random and fixed input worlds (some theory is sharper when we assume fixed input points, especially evenly spaced input points), but for the most part the theory is quite similar.

Therefore, in what follows, we won't be very precise about which setup we assume—random or fixed inputs—because it mostly doesn't matter when introducing nonparametric regression estimators and discussing basic properties.

1.3 Notation

We will define an empirical norm $\|\cdot\|_n$ in terms of the training points X_i, $i = 1, \ldots, n$, acting on functions $m : \mathbb{R}^d \to \mathbb{R}$, by

$$\|m\|_n^2 = \frac{1}{n} \sum_{i=1}^n m^2(X_i).$$

This makes sense no matter if the inputs are fixed or random (but in the latter case, it is a random norm)

When the inputs are considered random, we will write P_X for the distribution of X, and we will define the L_2 norm $\|\cdot\|_2$ in terms of P_X, acting on functions $m : \mathbb{R}^d \to \mathbb{R}$, by

$$\|m\|_2^2 = \mathbb{E}[m^2(X)] = \int m^2(x)\, dP_X(x).$$

So when you see $\|\cdot\|_2$ in use, it is a hint that the inputs are being treated as random

A quantity of interest will be the (squared) error associated with an estimator $\widehat{m}$ of m_0, which can be measured in either norm:

$$\|\widehat{m} - m_0\|_n^2 \quad \text{or} \quad \|\widehat{m} - m_0\|_2^2.$$

In either case, this is a random quantity (since $\widehat{m}$ is itself random). We will study bounds in probability or in expectation. The expectation of the errors defined above, in terms of either norm (but more typically the L_2 norm) is most properly called the risk; but we will often be a bit loose in terms of our terminology and just call this the error.

1.4 Bias-Variance Tradeoff

If (X, Y) is a new pair then

$$\mathbb{E}(Y - \widehat{m}(X))^2 = \int b_n^2(x) dP(x) + \int v(x) dP(x) + \tau^2 = ||\widehat{m} - m_0||_2^2 + \tau^2$$

where $b_n(x) = \mathbb{E}[\widehat{m}(x)] - m(x)$ is the bias, $v(x) = \mathrm{Var}(\widehat{m}(x))$ is the variance and $\tau^2 = \mathbb{E}(Y - m(X))^2$ is the un-avoidable error. Generally, we have to choose tuning parameters carefully to balance the bias and variance.

1.5 What does "nonparametric" mean?

Importantly, in nonparametric regression we don't assume a particular parametric form for m_0. This doesn't mean, however, that we can't estimate m_0 using (say) a linear combination of spline basis functions, written as $\widehat{m}(x) = \sum_{j=1}^p \widehat{\beta}_j g_j(x)$. A common question: the coefficients on the spline basis functions $\beta_1, \ldots, \beta_p$ are parameters, so how can this be nonparametric? Again, the point is that *we don't assume a parametric form for* m_0, i.e., we don't assume that m_0 itself is an exact linear combination of splines basis functions $g_1, \ldots, g_p$.

1.6 What we cover here

The goal is to expose you to a variety of methods, and give you a flavor of some interesting results, under different assumptions. A few topics we will cover into more depth than others, but overall, this will be far from a complete treatment of nonparametric regression. Below are some excellent texts out there that you can consult for more details, proofs, etc.

Nearest neighbors. Kernel smoothing, local polynomials: Tsybakov (2009) 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: Gyorfi, Kohler, Krzyzak & Walk (2002), Wasserman (2006) General references, more methodological: Hastie & Tibshirani (1990), Hastie, Tibshirani & Friedman (2009), Simonoff (1996)

Throughout, our discussion will bounce back and forth between the multivariate case ($d > 1$) and univariate case ($d = 1$). Some methods have obvious (natural) multivariate extensions; some don't. In any case, we can always use low-dimensional (even just univariate) nonparametric regression methods as building blocks for a high-dimensional nonparametric method. We'll study this near the end, when we talk about additive models.

1.7 Holder Spaces and Sobolev Spaces

The class of Lipschitz functions $H(1, L)$ on $T \subset \mathbb{R}$ is the set of functions g such that

$$|g(y) - g(x)| \leq L|x - y| \quad \text{for all } x, y \in T.$$

A differentiable function is Lipschitz if and only if it has bounded derivative. Conversely a Lipschitz function is differentiable almost everywhere.

Let $T \subset \mathbb{R}$ and let β be an integer. The Holder space $H(\beta, L)$ is the set of functions g mapping T to $\mathbb{R}$ such that g is $\ell = \beta - 1$ times differentiable and satisfies

$$|g^{(\ell)}(y) - g^{(\ell)}(x)| \leq L|x - y| \quad \text{for all } x, y \in T.$$

(There is an extension to real valued β but we will not need that.) If $g \in H(\beta, L)$ and $\ell = \beta - 1$, then we can define the Taylor approximation of g at x by

$$\widetilde{g}(y) = g(y) + (y - x)g'(x) + \cdots + \frac{(y - x)^\ell}{\ell!} g^{(\ell)}(x)$$

and then $|g(y) - \widetilde{g}(y)| \leq |y - x|^\beta$.

The definition for higher dimensions is similar. Let $\mathcal{X}$ be a compact subset of $\mathbb{R}^d$. Let β and L be positive numbers. Given a vector $s = (s_1, \ldots, s_d)$, define $|s| = s_1 + \cdots + s_d$, $s! = s_1! \cdots s_d!$, $x^s = x_1^{s_1} \cdots x_d^{s_d}$ and

$$D^s = \frac{\partial^{s_1+\cdots+s_d}}{\partial x_1^{s_1} \cdots \partial x_d^{s_d}}.$$

Let β be a positive integer. Define the *Hölder class*

$$H_d(\beta, L) = \left\{ g : \ |D^s g(x) - D^s g(y)| \leq L\|x-y\|, \quad \text{for all } s \text{ such that } |s| = \beta - 1, \text{ and all } x, y \right\}. \tag{1}$$

For example, if $d = 1$ and $\beta = 2$ this means that

$$|g'(x) - g'(y)| \leq L\,|x - y|, \quad \text{for all } x, y.$$

The most common case is $\beta = 2$; roughly speaking, this means that the functions have bounded second derivatives.

Again, if $g \in H_d(\beta, L)$ then $g(x)$ is close to its Taylor series approximation:

$$|g(u) - g_{x,\beta}(u)| \leq L\|u - x\|^\beta \tag{2}$$

where

$$g_{x,\beta}(u) = \sum_{|s| \leq \beta} \frac{(u-x)^s}{s!} D^s g(x). \tag{3}$$

In the common case of $\beta = 2$, this means that

$$\left| p(u) - [p(x) + (x-u)^T \nabla p(x)] \right| \leq L||x-u||^2.$$

The Sobolev class $S_1(\beta, L)$ is the set of β times differentiable functions (technically, it only requires weak derivatives) $g : \mathbb{R} \to \mathbb{R}$ such that

$$\int (g^{(\beta)}(x))^2 dx \leq L^2.$$

Again this extends naturally to $\mathbb{R}^d$. Also, there is an extension to non-integer β. It can be shown that $H_d(\beta, L) \subset S_d(\beta, L)$.

2 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

$$\widehat{m}(x) = \frac{1}{k} \sum_{i \in \mathcal{N}_k(x)} Y_i, \tag{4}$$

where $\mathcal{N}_k(x)$ contains the indices of the k closest points of $X_1, \ldots, X_n$ 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 $\widehat{m}$, 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 $\widehat{m}$ essentially always looks jagged, especially for small or moderate k. Why is this? It helps to write

$$\widehat{m}(x) = \sum_{i=1}^n w_i(x) Y_i, \tag{5}$$

where the weights $w_i(x)$, $i = 1, \ldots, n$ 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 is $\widehat{m}(x)$.

The representation (5) also reveals that the k-nearest-neighbors estimate is in a class of estimates we call *linear smoothers*, i.e., writing $Y = (Y_1, \ldots, Y_n) \in \mathbb{R}^n$, the vector of fitted values

$$\widehat{\mu} = (\widehat{m}(X_1), \ldots, \widehat{m}(X_n)) \in \mathbb{R}^n$$

can simply be expressed as $\widehat{\mu} = SY$. (To be clear, this means that for fixed inputs $X_1, \ldots, X_n$, the vector of fitted values $\widehat{\mu}$ is a linear function of Y; it does not mean that $\widehat{m}(x)$ 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 *universally consistent*, which means $\mathbb{E}\|\widehat{m} - m_0\|_2^2 \to 0$ as $n \to \infty$, with no assumptions other than $\mathbb{E}(Y^2) \leq \infty$, provided that we take $k = k_n$ such that $k_n \to \infty$ and $k_n/n \to 0$; e.g., $k = \sqrt{n}$ will do. See Chapter 6.2 of Gyorfi et al. (2002).

Furthermore, assuming the underlying regression function m_0 is Lipschitz continuous, the k-nearest-neighbors estimate with $k \asymp n^{2/(2+d)}$ satisfies

$$\mathbb{E}\|\widehat{m} - m_0\|_2^2 \lesssim n^{-2/(2+d)}. \tag{6}$$

See Chapter 6.3 of Gyorfi et al. (2002). Later, we will see that this is optimal.

Proof sketch: assume that $\mathrm{Var}(Y|X = x) = \sigma^2$, a constant, for simplicity, and fix (condition on) the training points. Using the bias-variance tradeoff,

$$\begin{aligned} \mathbb{E}\big[\big(\widehat{m}(x) - m_0(x)\big)^2\big] &= \underbrace{\big(\mathbb{E}[\widehat{m}(x)] - m_0(x)\big)^2}_{\mathrm{Bias}^2(\widehat{m}(x))} + \underbrace{\mathbb{E}\big[\big(\widehat{m}(x) - \mathbb{E}[\widehat{m}(x)]\big)^2\big]}_{\mathrm{Var}(\widehat{m}(x))} \\ &= \bigg(\frac{1}{k} \sum_{i \in \mathcal{N}_k(x)} \big(m_0(X_i) - m_0(x)\big)\bigg)^2 + \frac{\sigma^2}{k} \\ &\leq \bigg(\frac{L}{k} \sum_{i \in \mathcal{N}_k(x)} \|X_i - x\|_2\bigg)^2 + \frac{\sigma^2}{k}. \end{aligned}$$

In the last line we used the Lipschitz property $|m_0(x) - m_0(z)| \leq L\|x - z\|_2$, for some constant $L > 0$. Now for "most" of the points we'll have $\|X_i - x\|_2 \leq C(k/n)^{1/d}$, for a

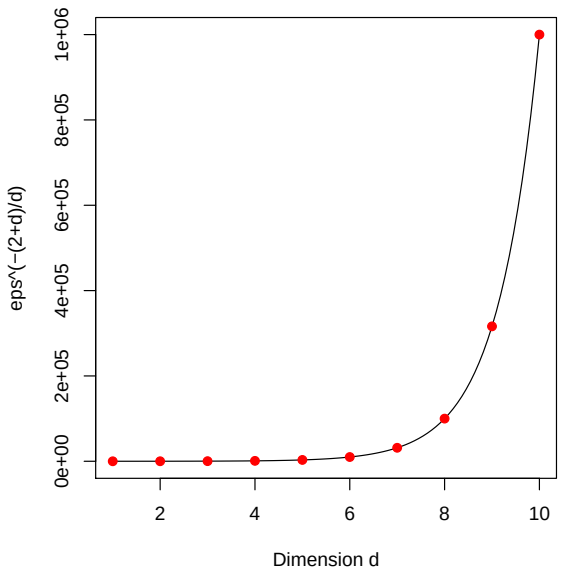

Figure 1: *The curse of dimensionality, with* $\epsilon = 0.1$

constant $C > 0$. (Think of a having input points X_i, $i = 1, \ldots, n$ spaced equally over (say) $[0, 1]^d$.) Then our bias-variance upper bound becomes

$$(CL)^2 \left(\frac{k}{n}\right)^{2/d} + \frac{\sigma^2}{k},$$

We can minimize this by balancing the two terms so that they are equal, giving $k^{1+2/d} \asymp n^{2/d}$, i.e., $k \asymp n^{2/(2+d)}$ as claimed. Plugging this in gives the error bound of $n^{-2/(2+d)}$, as claimed.

2.1 Curse of dimensionality

Note that the above error rate $n^{-2/(2+d)}$ exhibits a very poor dependence on the dimension d. To see it differently: given a small $\epsilon > 0$, think about how large we need to make n to ensure that $n^{-2/(2+d)} \leq \epsilon$. Rearranged, this says $n \geq \epsilon^{-(2+d)/2}$. That is, as we increase d, we require *exponentially more samples* n to achieve an error bound of ϵ. See Figure 1 for an illustration with $\epsilon = 0.1$

In fact, this phenomenon is not specific to k-nearest-neighbors, but a reflection of the *curse of dimensionality*, the principle that estimation becomes exponentially harder as the number of dimensions increases. This is made precise by minimax theory: we cannot hope to do better than the rate in(6) over $H_d(1, L)$, which we write for the space of L-Lipschitz functions in d dimensions, for a constant $L > 0$. It can be shown that

$$\inf_{\widehat{m}} \sup_{m_0 \in H_d(1,L)} \mathbb{E}\|\widehat{m} - m_0\|_2^2 \gtrsim n^{-2/(2+d)}, \tag{7}$$

where the infimum above is over all estimators $\widehat{m}$. See Chapter 3.2 of Gyorfi et al. (2002).

So why can we sometimes predict well in high dimensional problems? Presumably, it is because m_0 often (approximately) satisfies stronger assumptions. This suggests we should

look at classes of functions with more structure. One such example is the additive model, covered later in the notes.

3 Kernel Smoothing and Local Polynomials

3.1 Kernel smoothing

Kernel regression or *kernel smoothing* begins with a kernel function $K : \mathbb{R} \to \mathbb{R}$, satisfying

$$\int K(t)\,dt = 1, \quad \int tK(t)\,dt = 0, \quad 0 < \int t^2 K(t)\,dt < \infty.$$

Three common examples are the box-car kernel:

$$K(t) = \begin{cases} 1 & |x| \leq 1/2 \\ 0 & \text{otherwise} \end{cases},$$

the Gaussian kernel:

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

and the Epanechnikov kernel:

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

Warning! Don't confuse this with the notion of kernels in RKHS methods which we cover later.

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

$$\widehat{m}(x) = \frac{\sum_{i=1}^n K\left(\frac{\|x - X_i\|_2}{h}\right) Y_i}{\sum_{i=1}^n K\left(\frac{\|x - X_i\|_2}{h}\right)} = \sum_i w_i(x) Y_i \tag{8}$$

where $w_i(x) = K(\|x - X_i\|_2/h) / \sum_{j=1}^n K(\|x - x_j\|_2/h)$. Hence kernel smoothing is also a linear smoother.

In comparison to the k-nearest-neighbors estimator in (4), which can be thought of as a raw (discontinuous) moving average of nearby responses, the kernel estimator in (8) is a smooth moving average of responses. See Figure 2 for an example with $d = 1$.

3.2 Error Analysis

The kernel smoothing estimator is universally consistent ($\mathbb{E}\|\widehat{m} - m_0\|_2^2 \to 0$ as $n \to \infty$, with no assumptions other than $\mathbb{E}(Y^2) \leq \infty$), provided we take a compactly supported kernel K, and bandwidth $h = h_n$ satisfying $h_n \to 0$ and $nh_n^d \to \infty$ as $n \to \infty$. See Chapter 5.2 of Gyorfi et al. (2002). We can say more.

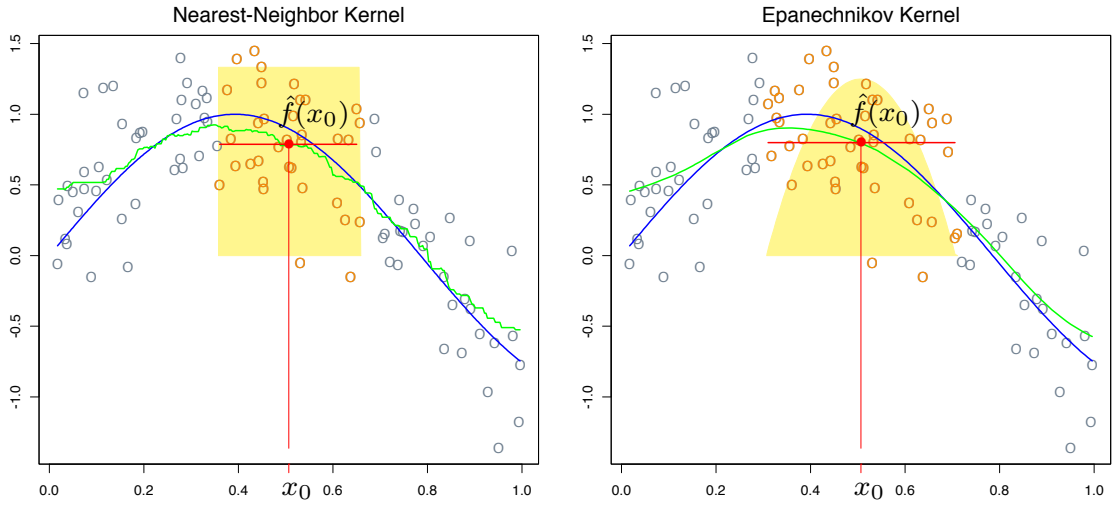

Figure 2: *Comparing k-nearest-neighbor and Epanechnikov kernels, when* $d = 1$. *From Chapter 6 of Hastie et al. (2009)*

Theorem. Suppose that $d = 1$ and that m'' is bounded. Also suppose that X has a non-zero, differentiable density p and that the support is unbounded. Then, the risk is

$$R_n = \frac{h_n^4}{4}\left(\int x^2 K(x)dx\right)^2 \int \left(m''(x) + 2m'(x)\frac{p'(x)}{p(x)}\right)^2 dx + \frac{\sigma^2 \int K^2(x)dx}{nh_n}\int \frac{dx}{p(x)} + o\left(\frac{1}{nh_n}\right) + o(h_n^4)$$

where p is the density of P_X.

The first term is the squared bias. The dependence on p and $p'$is the design bias and is undesirable. We'll fix this problem later using local linear smoothing. It follows that the optimal bandwidth is $h_n \approx n^{-1/5}$ yielding a risk of $n^{-4/5}$. In d dimensions, the term nh_n becomes nh_n^d. In that case It follows that the optimal bandwidth is $h_n \approx n^{-1/(4+d)}$ yielding a risk of $n^{-4/(4+d)}$.

If the support has boundaries then there is bias of order $O(h)$ near the boundary. This happens because of the asymmetry of the kernel weights in such regions. See Figure 3. Specifically, the bias is of order $O(h^2)$ in the interior but is of order $O(h)$ near the boundaries. The risk then becomes $O(h^3)$ instead of $O(h^4)$. We'll fix this problems using local linear smoothing. Also, the result above depends on assuming that P_X has a density. We can drop that assumption (and allow for boundaries) and get a slightly weaker result due to Gyorfi, Kohler, Krzyzak and Walk (2002).

For simplicity, we will use the spherical kernel $K(\|x\|) = I(\|x\| \leq 1)$; the results can be extended to other kernels. Hence,

$$\widehat{m}(x) = \frac{\sum_{i=1}^n Y_i\, I(\|X_i - x\| \leq h)}{\sum_{i=1}^n I(\|X_i - x\| \leq h)} = \frac{\sum_{i=1}^n Y_i\, I(\|X_i - x\| \leq h)}{n\, P_n(B(x,h))}$$

where P_n is the empirical measure and $B(x,h) = \{u : \|x-u\| \leq h\}$. If the denominator is 0 we define $\widehat{m}(x) = 0$. The proof of the following theorem is from Chapter 5 of Györfi, Kohler, Krzyżak and Walk (2002).

Theorem: Risk bound without density. Suppose that the distribution of X has compact support and that $\mathrm{Var}(Y|X=x) \leq \sigma^2 < \infty$ for all x. Then

$$\sup_{P \in H_d(1,L)} \mathbb{E}\|\widehat{m} - m\|_P^2 \leq c_1 h^2 + \frac{c_2}{nh^d}. \tag{9}$$

Hence, if $h \asymp n^{-1/(d+2)}$ then

$$\sup_{P \in H_d(1,L)} \mathbb{E}\|\widehat{m} - m\|_P^2 \leq \frac{c}{n^{2/(d+2)}}. \tag{10}$$

The proof is in the appendix. Note that the rate $n^{-2/(d+2)}$ is slower than the pointwise rate $n^{-4/(d+2)}$ because we have made weaker assumptions.

Recall from (7) we saw that this was the minimax optimal rate over $H_d(1,L)$. More generally, the minimax rate over $H_d(\alpha, L)$, for a constant $L > 0$, is

$$\inf_{\widehat{m}} \sup_{m_0 \in H_d(\alpha,L)} \mathbb{E}\|\widehat{m} - m_0\|_2^2 \gtrsim n^{-2\alpha/(2\alpha+d)}, \tag{11}$$

see again Chapter 3.2 of Gyorfi et al. (2002). However, as we saw above, with extra conditions, we got the rate $n^{-4/(4+d)}$ which is minimax for $H_d(2,L)$. We'll get that rate under weaker conditions with local linear regression.

If the support of the distribution of X lives on a smooth manifold of dimension $r < d$ then the term

$$\int \frac{dP(x)}{nP(B(x,h))}$$

is of order $1/(nh^r)$ instead of $1/(nh^d)$. In that case, we get the improved rate $n^{-2/(r+2)}$.

3.3 Local Linear Regression

We can alleviate this boundary bias issue by moving from a local constant fit to a local linear fit, or a local polynomial fit.

To build intuition, another way to view the kernel estimator in (8) is the following: at each input x, define the estimate $\widehat{m}(x) = \widehat{\theta}_x$, where $\widehat{\theta}_x$ is the minimizer of

$$\sum_{i=1}^n K\left(\frac{\|x - X_i\|}{h}\right)(Y_i - \theta)^2,$$

over all $\theta \in \mathbb{R}$. In other words, Instead we could consider forming the local estimate $\widehat{m}(x) = \widehat{\alpha}_x + \widehat{\beta}_x^T x$, where $\widehat{\alpha}_x, \widehat{\beta}_x$ minimize

$$\sum_{i=1}^n K\left(\frac{\|x - X_i\|}{h}\right)(Y_i - \alpha - \beta^T X_i)^2.$$

over all $\alpha \in \mathbb{R}$, $\beta \in \mathbb{R}^d$. This is called *local linear regression.*

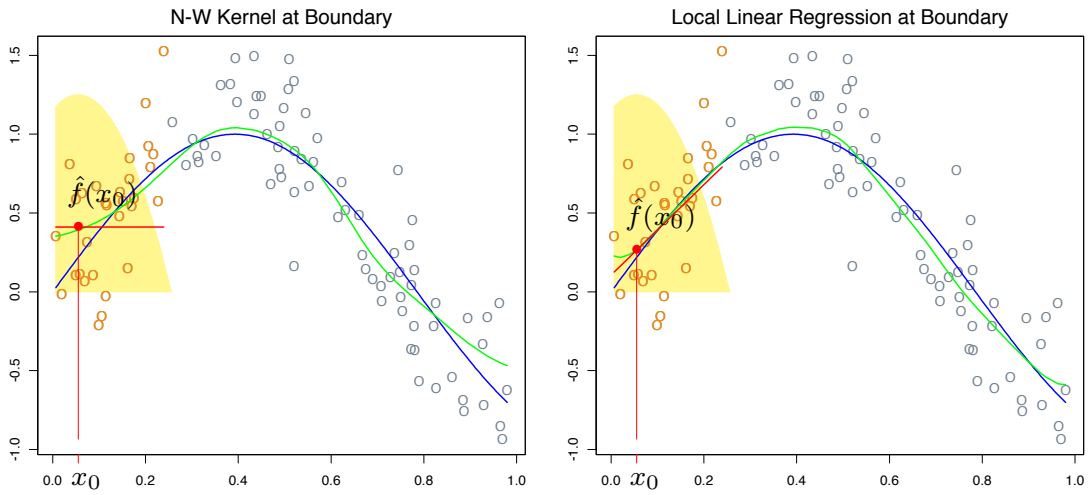

Figure 3: *Comparing (Nadaraya-Watson) kernel smoothing to local linear regression; the former is biased at the boundary, the latter is unbiased (to first-order). From Chapter 6 of Hastie et al. (2009)*

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

$$\widehat{m}(x) = b(x)^T (B^T \Omega B)^{-1} B^T \Omega Y,$$

where $b(x) = (1, x) \in \mathbb{R}^{d+1}$, $B \in \mathbb{R}^{n \times (d+1)}$ with ith row $b(X_i)$, and $\Omega \in \mathbb{R}^{n \times n}$ is diagonal with ith diagonal element $K(\|x - X_i\|_2 / h)$. We can write more concisely as $\widehat{m}(x) = w(x)^T Y$, where $w(x) = \Omega B (B^T \Omega B)^{-1} b(x)$, which shows local linear regression is a linear smoother too.

The vector of fitted values $\widehat{\mu} = (\widehat{m}(x_1), \ldots, \widehat{m}(x_n))$ can be expressed as

$$\widehat{\mu} = \begin{pmatrix} w_1(x)^T Y \\ \vdots \\ w_n(x)^T Y \end{pmatrix} = B (B^T \Omega B)^{-1} B^T \Omega Y = SY$$

which should look familiar to you from weighted least squares.

Now we'll sketch how the local linear fit reduces the bias, fixing (conditioning on) the training points. Compute at a fixed point x,

$$\mathbb{E}[\widehat{m}(x)] = \sum_{i=1}^n w_i(x) m_0(X_i).$$

Using a Taylor expansion of m_0 about x,

$$\mathbb{E}[\widehat{m}(x)] = m_0(x) \sum_{i=1}^n w_i(x) + \nabla m_0(x)^T \sum_{i=1}^n (X_i - x) w_i(x) + R,$$

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 $\widehat{m}$,

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

and so $\mathbb{E}[\widehat{m}(x)] = m_0(x) + R$, which means that $\widehat{m}$ is unbiased to first-order.

It can be shown that local linear regression removes boundary bias and design bias.

Theorem. Under some regularity conditions, the risk of $\widehat{m}$ is

$$\frac{h_n^4}{4} \int \left(\mathrm{tr}(m''(x) \int K(u) u u^T du) \right)^2 dP(x) + \frac{1}{n h_n^d} \int K^2(u) du \int \sigma^2(x) dP(x) + o(h_n^4 + (n h_n^d)^{-1}).$$

For a proof, see Fan & Gijbels (1996). For points near the boundary, the bias is $Ch^2 m''(x) + o(h^2)$ whereas, the bias is $Chm'(x) + o(h)$ for kernel estimators.

In fact, Fan (1993) shows a rather remarkable result. Let R_n be the minimax risk for estimating $m(x_0)$ over the class of functions with bounded second derivatives in a neighborhood of x_0. Let the maximum risk r_n of the local linear estimator with optimal bandwidth satisfies

$$1 + o(1) \geq \frac{R_n}{r_n} \geq (0.896)^2 + o(1).$$

Moreover, if we compute the minimax risk over all linear estimators we get $\frac{R_n}{r_n} \to 1$.

3.4 Higher-order smoothness

How can we hope to get optimal error rates over $H_d(\alpha, d)$, when $\alpha \geq 2$? With kernels there are basically two options: use local polynomials, or use higher-order kernels

Local polynomials build on our previous idea of local linear regression (itself an extension of kernel smoothing.) Consider $d = 1$, for concreteness. Define $\widehat{m}(x) = \widehat{\beta}_{x,0} + \sum_{j=1}^k \widehat{\beta}_{x,j} x^j$, where $\widehat{\beta}_{x,0}, \ldots, \widehat{\beta}_{x,k}$ minimize

$$\sum_{i=1}^n K\left(\frac{|x - X_i|}{h} \right) \left(Y_i - \beta_0 - \sum_{j=1}^k \beta_j X_i^j \right)^2.$$

over all $\beta_0, \beta_1, \ldots, \beta_k \in \mathbb{R}$. This is called ($k$th-order) *local polynomial regression*

Again we can express

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

where $b(x) = (1, x, \ldots, x^k)$, B is an $n \times (k+1)$ matrix with ith row $b(X_i) = (1, X_i, \ldots, X_i^k)$, and Ω is as before. Hence again, local polynomial regression is a linear smoother

Assuming that $m_0 \in H_1(\alpha, L)$ for a constant $L > 0$, a Taylor expansion shows that the local polynomial estimator $\widehat{m}$ of order k, where k is the largest integer strictly less than α and where the bandwidth scales as $h \asymp n^{-1/(2\alpha+1)}$, satisfies

$$\mathbb{E}\|\widehat{m} - m_0\|_2^2 \lesssim n^{-2\alpha/(2\alpha+1)}.$$

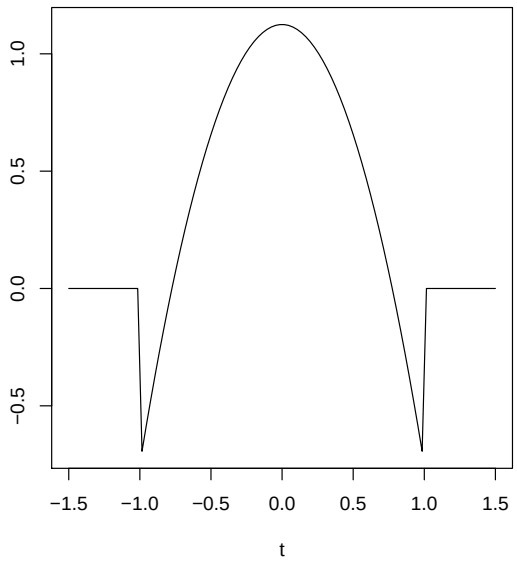

Figure 4: *A higher-order kernel function: specifically, a kernel of order 4*

See Chapter 1.6.1 of Tsybakov (2009). This matches the lower bound in (11) (when $d = 1$)

In multiple dimensions, $d > 1$, local polynomials become kind of tricky to fit, because of the explosion in terms of the number of parameters we need to represent a kth order polynomial in d variables. Hence, an interesting alternative is to return back kernel smoothing but use a *higher-order kernel*. A kernel function K is said to be of order k provided that

$$\int K(t)\,dt = 1, \quad \int t^j K(t)\,dt = 0, \quad j = 1, \ldots, k-1, \quad \text{and} \quad 0 < \int t^k K(t)\,dt < \infty.$$

This means that the kernels we were looking at so far were of order 2

An example of a 4th-order kernel is $K(t) = \frac{3}{8}(3 - 5t^2)1\{|t| \leq 1\}$, plotted in Figure 4. Notice that it takes negative values.

Lastly, while local polynomial regression and higher-order kernel smoothing can help "track" the derivatives of smooth functions $m_0 \in H_d(\alpha, L)$, $\alpha \geq 2$, it should be noted that they don't share the same universal consistency property of kernel smoothing (or k-nearest-neighbors). See Chapters 5.3 and 5.4 of Gyorfi et al. (2002)

4 Splines

Suppose that $d = 1$. Define an estimator by

$$\widehat{m} = \underset{f}{\operatorname{argmin}} \sum_{i=1}^{n} \big(Y_i - m(X_i)\big)^2 + \lambda \int_0^1 m''(x)^2\,dx. \tag{12}$$

Spline Lemma. The minimizer of (25) is a cubic spline with knots at the data points. (Proof in the Appendix.)

The key result presented above tells us that we can choose a basis $\eta_1, \dots, \eta_n$ for the set of kth-order natural splines with knots over $x_1, \dots, x_n$, and reparametrize the problem as

$$\widehat{\beta} = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \sum_{i=1}^n \Big(Y_i - \sum_{j=1}^n \beta_j \eta_j(X_i) \Big)^2 + \lambda \int_0^1 \Big(\sum_{j=1}^n \beta_j \eta_j''(x) \Big)^2 dx. \tag{13}$$

This is a finite-dimensional problem, and after we compute the coefficients $\widehat{\beta} \in \mathbb{R}^n$, we know that the smoothing spline estimate is simply $\widehat{m}(x) = \sum_{j=1}^n \widehat{\beta}_j \eta_j(x)$

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

$$N_{ij} = \eta_j(X_i) \quad \text{and} \quad \Omega_{ij} = \int_0^1 \eta_i''(x) \eta_j''(x)\, dx \quad \text{for } i, j = 1, \dots, n, \tag{14}$$

the problem in (27) can be written more succinctly as

$$\widehat{\beta} = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \ \|Y - N\beta\|_2^2 + \lambda \beta \Omega \beta, \tag{15}$$

showing the smoothing spline problem to be a type of generalized ridge regression problem. In fact, the solution in (29) has the explicit form

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

and therefore the fitted values $\widehat{\mu} = (\widehat{m}(x_1), \dots, \widehat{m}(x_n))$ are

$$\widehat{\mu} = N(N^T N + \lambda \Omega)^{-1} N^T Y \equiv SY. \tag{16}$$

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

A special property of smoothing splines: the fitted values in (30) 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

4.1 Error rates

Recall the *Sobolev class* of functions $S_1(m, C)$: for an integer $m \geq 0$ and $C > 0$, to contain all m times differentiable functions $f : \mathbb{R} \to \mathbb{R}$ such that

$$\int \big(f^{(m)}(x) \big)^2 dx \leq C^2.$$

(The Sobolev class $S_d(m, C)$ in d dimensions can be defined similarly, where we sum over all partial derivatives of order m.)

Assuming $m_0 \in S_1(m, C)$ for the underlying regression function, where $C > 0$ is a constant, the smoothing spline estimator $\widehat{m}$ of polynomial order $k = 2m - 1$ with tuning parameter $\lambda \asymp n^{1/(2m+1)} \asymp n^{1/(k+2)}$ satisfies

$$\|\widehat{m} - m_0\|_n^2 \lesssim n^{-2m/(2m+1)} \quad \text{in probability.}$$

The proof of this result uses much more fancy techniques from empirical process theory (entropy numbers) than the proofs for kernel smoothing. See Chapter 10.1 of van de Geer (2000) This rate is seen to be minimax optimal over $S_1(m, C)$ (e.g., Nussbaum (1985)).

5 Mercer kernels, RKHS

5.1 Hilbert Spaces

A Hilbert space is a complete inner product space. We will see that a reproducing kernel Hilbert space (RKHS) is a Hilbert space with extra structure that makes it very useful for statistics and machine learning.

An example of a Hilbert space is

$$L_2[0,1] = \left\{ f : [0,1] \to \mathbb{R} : \ \int f^2 < \infty \right\}$$

endowed with the inner product

$$\langle f, g \rangle = \int f(x) g(x) dx.$$

The corresponding norm is

$$||f|| = \sqrt{\langle f, f \rangle} = \sqrt{\int f^2(x) dx}.$$

We write $f_n \to f$ to mean that $||f_n - f|| \to 0$ as $n \to \infty$.

5.2 Evaluation Functional

The evaluation functional δ_x assigns a real number to each function. It is defined by $\delta_x f = f(x)$. In general, the evaluation functional is not continuous. This means we can have $f_n \to f$ but $\delta_x f_n$ does not converge to $\delta_x f$. For example, let $f(x) = 0$ and $f_n(x) = \sqrt{n} I(x < 1/n^2)$. Then $||f_n - f|| = 1/\sqrt{n} \to 0$. But $\delta_0 f_n = \sqrt{n}$ which does not converge to $\delta_0 f = 0$. Intuitively, this is because Hilbert spaces can contain very unsmooth functions. We shall see that RKHS are Hilbert spaces where the evaluation functional is continuous. Intuitively, this means that the functions in the space are well-behaved.

5.3 Nonparametric Regression

We observe $(X_1, Y_1), \ldots, (X_n, Y_n)$ and we want to estimate $m(x) = \mathbb{E}(Y|X = x)$. The approach we used earlier was based on **smoothing kernels**:

$$\widehat{m}(x) = \frac{\sum_i Y_i \ K\left(\frac{||x - X_i||}{h}\right)}{\sum_i K\left(\frac{||x - X_i||}{h}\right)}.$$

Another approach is regularization: choose m to minimize

$$\sum_i (Y_i - m(X_i))^2 + \lambda J(m)$$

for some penalty J. This is equivalent to: choose $m \in \mathcal{M}$ to minimize $\sum_i (Y_i - m(X_i))^2$ where $\mathcal{M} = \{m : \ J(m) \leq L\}$ for some $L > 0$.

We would like to construct $\mathcal{M}$ so that it contains smooth functions. We shall see that a good choice is to use a RKHS.

5.4 Mercer Kernels

A RKHS is defined by a **Mercer kernel**. A Mercer kernel $K(x, y)$ is a function of two variables that is symmetric and positive definite. This means that, for any function f,

$$\int \int K(x, y)f(x)f(y)dx\, dy \geq 0.$$

(This is like the definition of a positive definite matrix: $x^T Ax \geq 0$ for each x.)

Our main example is the Gaussian kernel

$$K(x, y) = e^{-\frac{||x-y||^2}{\sigma^2}}.$$

Given a kernel K, let $K_x(\cdot)$ be the function obtained by fixing the first coordinate. That is, $K_x(y) = K(x, y)$. For the Gaussian kernel, K_x is a Normal, centered at x. We can create functions by taking linear combinations of the kernel:

$$f(x) = \sum_{j=1}^{k} \alpha_j K_{x_j}(x).$$

Let $\mathcal{H}_0$ denote all such functions:

$$\mathcal{H}_0 = \left\{ f : \sum_{j=1}^{k} \alpha_j K_{x_j}(x) \right\}.$$

Given two such functions $f(x) = \sum_{j=1}^{k} \alpha_j K_{x_j}(x)$ and $g(x) = \sum_{j=1}^{m} \beta_j K_{y_j}(x)$ we define an inner product

$$\langle f, g\rangle = \langle f, g\rangle_K = \sum_i \sum_j \alpha_i \beta_j K(x_i, y_j).$$

In general, f (and g) might be representable in more than one way. You can check that $\langle f, g\rangle_K$ is independent of how f (or g) is represented. The inner product defines a norm:

$$||f||_K = \sqrt{\langle f, f, \rangle} = \sqrt{\sum_j \sum_k \alpha_j \alpha_k K(x_j, x_k)} = \sqrt{\alpha^T \mathbb{K} \alpha}$$

where $\alpha = (\alpha_1, \ldots, \alpha_k)^T$ and $\mathbb{K}$ is the $k \times k$ matrix with $\mathbb{K}_{jk} = K(x_j, x_k)$.

5.5 The Reproducing Property

Let $f(x) = \sum_i \alpha_i K_{x_i}(x)$. Note the following crucial property:

$$\langle f, K_x\rangle = \sum_i \alpha_i K(x_i, x) = f(x).$$

This follows from the definition of $\langle f, g\rangle$ where we take $g = K_x$. This implies that

$$\langle K_x, K_y\rangle = K(x, y).$$

This is called the reproducing property. It also implies that K_x is the **representer** of the evaluation functional.

The completion of $\mathcal{H}_0$ with respect to $||\cdot||_K$ is denoted by $\mathcal{H}_K$ and is called the RKHS generated by K.

To verify that this is a well-defined Hilbert space, you should check that the following properties hold:

$$\begin{aligned} \langle f, g\rangle &= \langle g, f\rangle \\ \langle cf + dg, h\rangle &= c\langle f, h\rangle + c\langle g, h\rangle \\ \langle f, f\rangle = 0 \quad &\text{iff} \quad f = 0. \end{aligned}$$

The last one is not obvious so let us verify it here. It is easy to see that $f = 0$ implies that $\langle f, f\rangle = 0$. Now we must show that $\langle f, f\rangle = 0$ implies that $f(x) = 0$. So suppose that $\langle f, f\rangle = 0$. Pick any x. Then

$$\begin{aligned} 0 &\leq f^2(x) = \langle f, K_x\rangle^2 = \langle f, K_x\rangle \ \langle f, K_x\rangle \\ &\leq ||f||^2 \ ||K_x||^2 = \langle f, f\rangle^2 \ ||K_x||^2 = 0 \end{aligned}$$

where we used Cauchy-Schwartz. So $0 \leq f^2(x) \leq 0$ which means that $f(x) = 0$.

Returning to the evaluation functional, suppose that $f_n \to f$. Then

$$\delta_x f_n = \langle f_n, K_x\rangle \to \langle f, K_x\rangle = f(x) = \delta_x f$$

so the evaluation functional is continuous. **In fact, a Hilbert space is a RKHS if and only if the evaluation functionals are continuous.**

5.6 Examples

Example 1. Let $\mathcal{H}$ be all functions f on $\mathbb{R}$ such that the support of the Fourier transform of f is contained in $[-a, a]$. Then

$$K(x, y) = \frac{\sin(a(y - x))}{a(y - x)}$$

and

$$\langle f, g\rangle = \int fg.$$

Example 2. Let $\mathcal{H}$ be all functions f on $(0, 1)$ such that

$$\int_0^1 (f^2(x) + (f'(x))^2)x^2 dx < \infty.$$

Then

$$K(x, y) = (xy)^{-1}\left(e^{-x}\sinh(y)I(0 < x \leq y) + e^{-y}\sinh(x)I(0 < y \leq x)\right)$$

and

$$||f||^2 = \int_0^1 (f^2(x) + (f'(x))^2)x^2 dx.$$

Example 3. The Sobolev space of order m is (roughly speaking) the set of functions f such that $\int (f^{(m)})^2 < \infty$. For $m = 1$ and $\mathcal{X} = [0, 1]$ the kernel is

$$K(x, y) = \begin{cases} 1 + xy + \frac{xy^2}{2} - \frac{y^3}{6} & 0 \leq y \leq x \leq 1 \\ 1 + xy + \frac{yx^2}{2} - \frac{x^3}{6} & 0 \leq x \leq y \leq 1 \end{cases}$$

and

$$||f||_K^2 = f^2(0) + f'(0)^2 + \int_0^1 (f''(x))^2 dx.$$

5.7 Spectral Representation

Suppose that $\sup_{x,y} K(x, y) < \infty$. Define eigenvalues λ_j and orthonormal eigenfunctions ψ_j by

$$\int K(x, y)\psi_j(y)dy = \lambda_j \psi_j(x).$$

Then $\sum_j \lambda_j < \infty$ and $\sup_x |\psi_j(x)| < \infty$. Also,

$$K(x, y) = \sum_{j=1}^{\infty} \lambda_j \psi_j(x)\psi_j(y).$$

Define the **feature map** Φ by

$$\Phi(x) = (\sqrt{\lambda_1}\psi_1(x), \sqrt{\lambda_2}\psi_2(x), \ldots).$$

We can expand f either in terms of K or in terms of the basis $\psi_1, \psi_2, \ldots$:

$$f(x) = \sum_i \alpha_i K(x_i, x) = \sum_{j=1}^{\infty} \beta_j \psi_j(x).$$

Furthermore, if $f(x) = \sum_j a_j \psi_j(x)$ and $g(x) = \sum_j b_j \psi_j(x)$, then

$$\langle f, g \rangle = \sum_{j=1}^{\infty} \frac{a_j b_j}{\lambda_j}.$$

Roughly speaking, when $||f||_K$ is small, then f is smooth.

5.8 Representer Theorem

Let ℓ be a loss function depending on $(X_1, Y_1), \ldots, (X_n, Y_n)$ and on $f(X_1), \ldots, f(X_n)$. Let $\widehat{f}$ minimize

$$\ell + g(||f||_K^2)$$

where g is any monotone increasing function. Then $\widehat{f}$ has the form

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

for some $\alpha_1, \ldots, \alpha_n$.

5.9 RKHS Regression

Define $\widehat{m}$ to minimize

$$R = \sum_i (Y_i - m(X_i))^2 + \lambda ||m||_K^2.$$

By the representer theorem, $\widehat{m}(x) = \sum_{i=1}^n \alpha_i K(x_i, x)$. Plug this into R and we get

$$R = ||Y - \mathbb{K}\alpha||^2 + \lambda \alpha^T \mathbb{K} \alpha$$

where $\mathbb{K}_{jk} = K(X_j, X_k)$ is the Gram matrix. The minimizer over α is

$$\widehat{\alpha} = (\mathbb{K} + \lambda I)^{-1} Y$$

and $\widehat{m}(x) = \sum_j \widehat{\alpha}_j K(X_i, x)$. The fitted values are

$$\widehat{Y} = \mathbb{K}\widehat{\alpha} = \mathbb{K}(\mathbb{K} + \lambda I)^{-1} Y = LY.$$

So this is a linear smoother.

We can use cross-validation to choose λ. **Compare this with smoothing kernel regression.**

5.10 Logistic Regression

Let

$$m(x) = \mathbb{P}(Y = 1 | X = x) = \frac{e^{f(x)}}{1 + e^{f(x)}}.$$

We can estimate m by minimizing

$$-\text{loglikelihood} + \lambda ||f||_K^2.$$

Then $\widehat{f} = \sum_j K(x_j, x)$ and α may be found by numerical optimization. In this case, smoothing kernels are much easier.

5.11 Support Vector Machines

Suppose $Y_i \in \{-1, +1\}$. Recall the the linear SVM minimizes the penalized hinge loss:

$$J = \sum_i [1 - Y_i(\beta_0 + \beta^T X_i)]_+ + \frac{\lambda}{2} ||\beta||_2^2.$$

The dual is to maximize

$$\sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j Y_i Y_j \langle X_i, X_j \rangle$$

subject to $0 \leq \alpha_i \leq C$.

The RKHS version is to minimize

$$J = \sum_i [1 - Y_i f(X_i)]_+ + \frac{\lambda}{2} ||f||_K^2.$$

The dual is the same except that $\langle X_i, X_j \rangle$ is replaced with $K(X_i, X_j)$. This is called the kernel trick.

5.12 The Kernel Trick

This is a fairly general trick. In many algorithms you can replace $\langle x_i, x_j \rangle$ with $K(x_i, x_j)$ and get a nonlinear version of the algorithm. This is equivalent to replacing x with $\Phi(x)$ and replacing $\langle x_i, x_j \rangle$ with $\langle \Phi(x_i), \Phi(x_j) \rangle$. However, $K(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$ and $K(x_i, x_j)$ is much easier to compute.

In summary, by replacing $\langle x_i, x_j \rangle$ with $K(x_i, x_j)$ we turn a linear procedure into a nonlinear procedure without adding much computation.

5.13 Hidden Tuning Parameters

There are hidden tuning parameters in the RKHS. Consider the Gaussian kernel

$$K(x, y) = e^{-\frac{||x-y||^2}{\sigma^2}}.$$

For nonparametric regression we minimize $\sum_i (Y_i - m(X_i))^2$ subject to $||m||_K \leq L$. We control the bias variance tradeoff by doing cross-validation over L. But what about σ?

This parameter seems to get mostly ignored. Suppose we have a uniform distribution on a circle. The eigenfunctions of $K(x, y)$ are the sines and cosines. The eigenvalues λ_k die off like $(1/\sigma)^{2k}$. So σ affects the bias-variance tradeoff since it weights things towards lower order Fourier functions. In principle we can compensate for this by varying L. But clearly there is some interaction between L and σ. The practical effect is not well understood. We'll see this again when we discuss interpolation.

Now consider the polynomial kernel $K(x, y) = (1 + \langle x, y \rangle)^d$. This kernel has the same eigenfunctions but the eigenvalues decay at a polynomial rate depending on d. So there is an interaction between L, d and, the choice of kernel itself.

6 Linear smoothers

6.1 Definition

Every estimator we have discussed so far is a linear smoother meaning that $\widehat{m}(x) = \sum_i w_i(x) Y_i$ for some weights $w_i(x)$ that do not depend on the $Y_i's$. Hence, the fitted values $\widehat{\mu} = (\widehat{m}(X_1), \ldots, \widehat{m}(X_n))$ are of the form $\widehat{\mu} = SY$ for some matrix $S \in \mathbb{R}^{n \times n}$ depending on the inputs $X_1, \ldots, X_n$—and also possibly on a tuning parameter such as h in kernel smoothing, or λ in smoothing splines—but not on the Y_i's. We call S, the smoothing matrix. For comparison, recall that in linear regression, $\widehat{\mu} = HY$ for some projection matrix H.

For linear smoothers $\widehat{\mu} = SY$, the effective degrees of freedom is defined to be

$$\nu \equiv \mathrm{df}(\widehat{\mu}) \equiv \sum_{i=1}^{n} S_{ii} = \mathrm{tr}(S),$$

the trace of the smooth matrix S

6.2 Cross-validation

K-fold cross-validation can be used to estimate the prediction error and choose tuning parameters.

For linear smoothers $\widehat{\mu} = (\widehat{m}(x_1), \ldots \widehat{m}(x_n)) = SY$, leave-one-out cross-validation can be particularly appealing because in many cases we have the seemingly magical reduction

$$\mathrm{CV}(\widehat{m}) = \frac{1}{n} \sum_{i=1}^{n} \big(Y_i - \widehat{m}^{-i}(X_i)\big)^2 = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{Y_i - \widehat{m}(X_i)}{1 - S_{ii}} \right)^2, \tag{17}$$

where $\widehat{m}^{-i}$ denotes the estimated regression function that was trained on all but the ith pair (X_i, Y_i). This leads to a big computational savings since it shows us that, to compute leave-one-out cross-validation error, we don't have to actually ever compute $\widehat{m}^{-i}$, $i = 1, \ldots, n$.

Why does (17) hold, and for which linear smoothers $\widehat{\mu} = Sy$? Just rearranging (17) perhaps demystifies this seemingly magical relationship and helps to answer these questions. Suppose we knew that $\widehat{m}$ had the property

$$\widehat{m}^{-i}(X_i) = \frac{1}{1 - S_{ii}} \big(\widehat{m}(X_i) - S_{ii} Y_i\big). \tag{18}$$

That is, to obtain the estimate at X_i under the function $\widehat{m}^{-i}$ fit on all but (X_i, Y_i), we take the sum of the linear weights (from our original fitted function $\widehat{m}$) across all but the ith point, $\widehat{m}(X_i) - S_{ii} Y_i = \sum_{i \neq j} S_{ij} Y_j$, 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).

From the special property (18), it is easy to show the leave-one-out formula (17). We have

$$Y_i - \widehat{m}^{-i}(X_i) = Y_i - \frac{1}{1 - S_{ii}} \big(\widehat{m}(X_i) - S_{ii} Y_i\big) = \frac{Y_i - \widehat{m}(X_i)}{1 - S_{ii}},$$

and then squaring both sides and summing over n gives (17).

Finally, *generalized cross-validation* is a small twist on the right-hand side in (17) 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 $\mathrm{tr}(S)/n$,

$$\mathrm{GCV}(\widehat{m}) = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{Y_i - \widehat{m}(X_i)}{1 - \mathrm{tr}(S)/n} \right)^2 = (1 - \nu/n)^{-2} \widehat{R}$$

where ν is the effective degrees of freedom and $\widehat{R}$ is the training error. This can be of computational advantage in some cases where $\mathrm{tr}(S)$ is easier to compute that individual elements S_{ii}.

7 Additive models

7.1 Motivation and definition

Computational efficiency and statistical efficiency are both very real concerns as the dimension d grows large, in nonparametric regression. If you're trying to fit a kernel, thin-plate

spline, or RKHS estimate in > 20 dimensions, without any other kind of structural constraints, then you'll probably be in trouble (unless you have a very fast computer and tons of data).

Recall from (11) that the minimax rate over the Holder class $H_d(\alpha, L)$ is $n^{-2\alpha/(2\alpha+d)}$, which has an exponentially bad dependence on the dimension d. This is usually called the curse of dimensionality (though the term apparently originated with Bellman (1962), who encountered an analogous issue but in a separate context—dynamic programming).

What can we do? One answer is to change what we're looking for, and fit estimates with less flexibility in high dimensions. Think of a linear model in d variables: there is a big difference between this and a fully nonparametric model in d variables. Is there some middle man that we can consider, that would make sense?

Additive models play the role of this middle man. Instead of considering a full d-dimensional function of the form

$$m(x) = m(x(1), \ldots, x(d)) \tag{19}$$

we restrict our attention to functions of the form

$$m(x) = m_1(x(1)) + \cdots + m_d(x(d)). \tag{20}$$

As each function m_j, $j = 1, \ldots, d$ is univariate, fitting an estimate of the form (20) is certainly less ambitious than fitting one of the form (19). On the other hand, the scope of (20) is still big enough that we can capture interesting (marginal) behavior in high dimensions.

There is a link to naive-Bayes classification that we will discuss later.

The choice of estimator of the form (20) need not be regarded as an assumption we make about the true function m_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.

A classic result by Stone (1985) encapsulates this idea precisely. He shows that, while it may be difficult to estimate an arbitrary regression function m_0 in multiple dimensions, we can still estimate its *best additive approximation* $\overline{m}^{\text{add}}$ well. Assuming each component function $\overline{m}^{\text{add}}_{0,j}$, $j = 1, \ldots, d$ lies in the Holder class $H_1(\alpha, L)$, for constant $L > 0$, and we can use an additive model, with each component $\widehat{m}_j$, $j = 1, \ldots, d$ estimated using an appropriate kth degree spline, to give

$$\mathbb{E}\|\widehat{m}_j - \overline{m}^{\text{add}}_j\|_2^2 \lesssim n^{-2\alpha/(2\alpha+1)}, \quad j = 1, \ldots, d.$$

Hence each component of the best additive approximation $\overline{f}^{\text{add}}$ to m_0 can be estimated at the optimal univariate rate. Loosely speaking, though we cannot hope to recover m_0 arbitrarily, we can recover its major structure along the coordinate axes.

7.2 Backfitting

Estimation with additive models is actually very simple; we can just choose our favorite univariate smoother (i.e., nonparametric estimator), and cycle through estimating each

function m_j, $j = 1, \ldots, d$ individually (like a block coordinate descent algorithm). Denote the result of running our chosen univariate smoother to regress $Y = (Y_1, \ldots, Y_n) \in \mathbb{R}^n$ over the input points $Z = (Z_1, \ldots, Z_n) \in \mathbb{R}^n$ as

$$\widehat{m} = \text{Smooth}(Z, Y).$$

E.g., we might choose $\text{Smooth}(\cdot, \cdot)$ to be a cubic smoothing spline with some fixed value of the tuning parameter λ, or even with the tuning parameter selected by generalized cross-validation

Once our univariate smoother has been chosen, we initialize $\widehat{m}_1, \ldots, \widehat{m}_d$ (say, to all to zero) and cycle over the following steps for $j = 1, \ldots, d, 1, \ldots, d, \ldots$:

1. define $r_i = Y_i - \sum_{\ell \neq j} \widehat{m}_\ell(x_{i\ell})$, $i = 1, \ldots, n$;
2. smooth $\widehat{m}_j = \text{Smooth}(x(j), r)$;
3. center $\widehat{m}_j = \widehat{m}_j - \frac{1}{n} \sum_{i=1}^n \widehat{m}_j(X_i(j))$.

This algorithm is known as *backfitting*. In last step above, we are removing the mean from each fitted function $\widehat{m}_j$, $j = 1, \ldots, d$, otherwise the model would not be identifiable. Our final estimate therefore takes the form

$$\widehat{m}(x) = \overline{Y} + \widehat{m}_1(x(1)) + \cdots + \widehat{m}(x(d))$$

where $\overline{Y} = \frac{1}{n} \sum_{i=1}^n Y_i$. Hastie & Tibshirani (1990) provide a very nice exposition on the some of the more practical aspects of backfitting and additive models.

In many cases, backfitting is equivalent to blockwise coordinate descent performed on a joint optimization criterion that determines the total additive estimate. E.g., for the additive cubic smoothing spline optimization problem,

$$\widehat{m}_1, \ldots, \widehat{m}_d = \underset{m_1, \ldots, m_d}{\text{argmin}} \sum_{i=1}^n \left(Y_i - \sum_{j=1}^d m_j(x_{ij}) \right)^2 + \sum_{j=1}^d \lambda_j \int_0^1 m_j''(t)^2 \, dt,$$

backfitting is exactly blockwise coordinate descent (after we reparametrize the above to be in finite-dimensional form, using a natural cubic spline basis).

The beauty of backfitting is that it allows us to think *algorithmically*, and plug in whatever we want for the univariate smoothers. This allows for several extensions. One extension: we don't need to use the same univariate smoother for each dimension, rather, we could mix and match, choosing $\text{Smooth}_j(\cdot, \cdot)$, $j = 1, \ldots, d$ to come from entirely different methods or giving estimates with entirely different structures.

Another extension: to capture interactions, we can perform smoothing over (small) groups of variables instead of individual variables. For example we could fit a model of the form

$$m(x) = \sum_j m_j(x(j)) + \sum_{j<k} m_{jk}(x(j), x(k)).$$

7.3 Error rates

Error rates for additive models are both kind of what you'd expect and surprising. What you'd expect: if the underlying function m_0 is additive, and we place standard assumptions on its component functions, such as $f_{0,j} \in S_1(m, C)$, $j = 1, \ldots, d$, for a constant $C > 0$, a somewhat straightforward argument building on univariate minimax theory gives us the lower bound

$$\inf_{\widehat{m}} \sup_{m_0 \in \oplus_{j=1}^d S_1(m,C)} \mathbb{E}\|\widehat{m} - m_0\|_2^2 \gtrsim dn^{-2m/(2m+1)}.$$

This is simply d times the univariate minimax rate. (Note that we have been careful to track the role of d here, i.e., it is not being treated like a constant.) Also, standard methods like backfitting with univariate smoothing splines of polynomial order $k = 2m - 1$, will also match this upper bound in error rate (though the proof to get the sharp linear dependence on d is a bit trickier).

7.4 Sparse additive models

Recently, *sparse additive models* have received a good deal of attention. In truly high dimensions, we might believe that only a small subset of the variables play a useful role in modeling the regression function, so might posit a modification of (20) of the form

$$m(x) = \sum_{j \in S} m_j(x(j))$$

where $S \subseteq \{1, \ldots, d\}$ is an unknown subset of the full set of dimensions.

This is a natural idea, and to estimate a sparse additive model, we can use methods that are like nonparametric analogies of the lasso (more accurately, the group lasso). This is a research topic still very much in development; some recent works are Lin & Zhang (2006), Ravikumar et al. (2009), Raskutti et al. (2012). We'll cover this in more detail when we talk about the sparsity, the lasso, and high-dimensional estimation.

8 Variance Estimation and Confidence Bands

Let

$$\sigma^2(x) = \mathrm{Var}(Y|X = x).$$

We can estimate $\sigma^2(x)$ as follows. Let $\widehat{m}(x)$ be an estimate of the regression function. Let $e_i = Y_i - \widehat{m}(X_i)$. Now apply nonparametric regression again treating e_i^2 as the response. The resulting estimator $\widehat{\sigma}^2(x)$ can be shown to be consistent under some regularity conditions.

Ideally we would also like to find random functions ℓ_n and u_n such that

$$P(\ell_n(x) \leq m(x) \leq u_n(x) \text{ for all } x) \to 1 - \alpha.$$

For the reasons we discussed earlier with density functions, this is essentially an impossible problem.

We can, however, still get an informal (but useful) estimate the variability of $\widehat{m}(x)$. Suppose that $\widehat{m}(x) = \sum_i w_i(x) Y_i$. The conditional variance is $\sum_i w_i^2(x)\sigma^2(x)$ which can

be estimated by $\sum_i w_i^2(x)\widehat{\sigma}^2(x)$. An asymptotic, pointwise (biased) confidence band is $\widehat{m}(x) \pm z_{\alpha/2}\sqrt{\sum_i w_i^2(x)\widehat{\sigma}^2(x)}$.

A better idea is to bootstrap the quantity

$$\frac{\sqrt{n}\sup_x |\widehat{m}(x) - \mathbb{E}[\widehat{m}(x)]|}{\widehat{\sigma}(x)}$$

to get a bootstrap quantile t_n. Then

$$\left[\widehat{m}(x) - \frac{t_n\widehat{\sigma}(x)}{\sqrt{n}},\ \widehat{m}(x) + \frac{t_n\widehat{\sigma}(x)}{\sqrt{n}}\right]$$

is a bootstrap variability band.

9 Wavelet smoothing

Not every nonparametric regression estimate needs to be a linear smoother (though this does seem to be very common), and *wavelet smoothing* is one of the leading nonlinear tools for nonparametric estimation. The theory of wavelets is elegant and we only give a brief 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

We assume $d = 1$. Local adaptivity in higher dimensions is not nearly as settled as it is with smoothing splines or (especially) kernels (multivariate extensions of wavelets are possible, i.e., *ridgelets* and *curvelets*, but are complex)

Consider basis functions, $\phi_1, \ldots, \phi_n$, evaluated over n equally spaced inputs over $[0, 1]$:

$$X_i = i/n, \quad i = 1, \ldots, n.$$

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

$$W_{ij} = \phi_j(X_i), \quad i, j = 1, \ldots, n$$

The goal, given outputs $y = (y_1, \ldots, y_n)$ 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):

$$\widetilde{\theta} = W^T y,$$

we threshold the coefficients θ (the threshold function T_λ to be defined shortly):

$$\widehat{\theta} = T_\lambda(\widetilde{\theta}),$$

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

$$\widehat{\mu} = W\widehat{\theta}$$

The wavelet and inverse wavelet transforms (multiplication by W^T and W) each require $O(n)$ operations, and are practically extremely fast due do clever pyramidal multiplication schemes that exploit the special structure of wavelets

The threshold function T_λ is usually taken to be hard-thresholding, i.e.,

$$[T_\lambda^{\text{hard}}(z)]_i = z_i \cdot 1\{|z_i| \geq \lambda\}, \quad i = 1, \ldots, n,$$

or soft-thresholding, i.e.,

$$[T_\lambda^{\text{soft}}(z)]_i = \big(z_i - \text{sign}(z_i)\lambda\big) \cdot 1\{|z_i| \geq \lambda\}, \quad i = 1, \ldots, n.$$

These thresholding functions are both also $O(n)$, and computationally trivial, making wavelet smoothing very fast overall

We should emphasize that wavelet smoothing is not a linear smoother, i.e., there is no single matrix S such that $\widehat{\mu} = Sy$ 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

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

and then the wavelet smoothing fitted values are $\widehat{\mu} = W\widehat{\theta}$. Here $\|\theta\|_0 = \sum_{i=1}^n 1\{\theta_i \neq 0\}$, the number of nonzero components of θ, called the "ℓ_0 norm". For soft-thresholding, we solve

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

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

9.1 The strengths of wavelets, the limitations of linear smoothers

Apart from its computational efficiency, 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

We can make this more precise by considering convergence rates over an appropriate function class. In particular, we define the *total variation class* $M(k, C)$, for an integer $k \geq 0$ and $C > 0$, to contain all k times (weakly) differentiable functions whose kth derivative satisfies

$$\text{TV}(f^{(k)}) = \sup_{0 = z_1 < z_2 < \ldots < z_N < z_{N+1} = 1} \sum_{j=1}^N |f^{(k)}(z_{i+1}) - f^{(k)}(z_i)| \ \leq \ C.$$

(Note that if f has $k+1$ continuous derivatives, then $\text{TV}(f^{(k)}) = \int_0^1 |f^{(k+1)}(x)|\, dx$.)

For the wavelet smoothing estimator, denoted by $\widehat{m}^{\text{wav}}$, Donoho & Johnstone (1998) provide a seminal analysis. Assuming that $m_0 \in M(k, C)$ for a constant $C > 0$ (and further conditions on the setup), they show that (for an appropriate scaling of the smoothing parameter λ),

$$\mathbb{E}\|\widehat{m}^{\text{wav}} - m_0\|_2^2 \lesssim n^{-(2k+2)/(2k+3)} \quad \text{and} \quad \inf_{\widehat{m}} \sup_{m_0 \in M(k,C)} \mathbb{E}\|\widehat{m} - m_0\|_2^2 \gtrsim n^{-(2k+2)/(2k+3)}. \tag{21}$$

Thus wavelet smoothing attains the minimax optimal rate over the function class $M(k, C)$. (For a translation of this result to the notation of the current setting, see Tibshirani (2014).)

Some important questions: (i) just how big is the function class $M(k, C)$? And (ii) can a linear smoother also be minimax optimal over $M(k, C)$?

It is not hard to check $M(k, C) \supseteq S_1(k + 1, C')$, the (univariate) Sobolev space of order $k + 1$, for some other constant $C' > 0$. We know from the previously mentioned theory on Sobolev spaces that the minimax rate over $S_1(k + 1, C')$ is again $n^{-(2k+2)/(2k+3)}$. This suggests that these two function spaces might actually be somewhat close in size

But in fact, the overall minimax rates here are sort of misleading, and we will see from the behavior of linear smoothers that the function classes are actually quite different. Donoho & Johnstone (1998) showed that the minimax error over $M(k, C)$, *restricted to linear smoothers*, satisfies

$$\inf_{\widehat{m} \text{ linear}} \sup_{m_0 \in M(k,C)} \mathbb{E}\|\widehat{m} - m_0\|_2^2 \gtrsim n^{-(2k+1)/(2k+2)}. \tag{22}$$

(See again Tibshirani (2014) for a translation to the notation of the current setting.) Hence the answers to our questions are: (ii) linear smoothers cannot cope with the heterogeneity of functions in $M(k, C)$, and are are bounded away from optimality, which means (i) we can interpret $M(k, C)$ as being much larger than $S_1(k + 1, C')$, because linear smoothers can be optimal over the latter class but not over the former. See Figure 5 for a diagram

Let's back up to emphasize just how remarkable the results (21), (22) really are. Though it may seem like a subtle difference in exponents, there is actually a significant difference in the minimax rate and minimax linear rate: e.g., when $k = 0$, this is a difference of $n^{-1/2}$ (optimal) and $n^{-1/2}$ (optimal among linear smoothers) for estimating a function of bounded variation. Recall also just how broad the linear smoother class is: kernel smoothing, regression splines, smoothing splines, RKHS estimators ... none of these methods can achieve a better rate than $n^{-1/2}$ over functions of bounded variation

Practically, the differences between 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 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, though you might say they marked the beginning of the story, wavelets are not the end of the story when it comes to local adaptivity. The natural thing to do, it might seem, is to make (say) kernel smoothing or smoothing splines more locally adaptive by allowing for a local bandwidth parameter or a local penalty parameter. People have tried this, but it

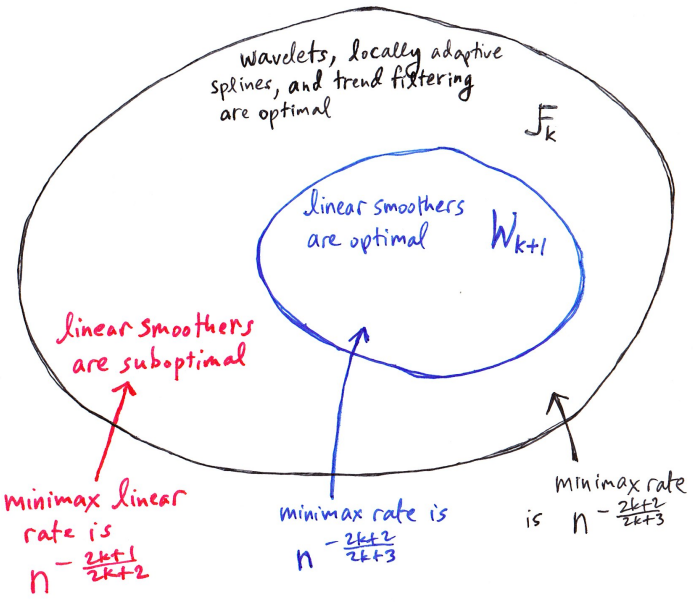

Figure 5: *A diagram of the minimax rates over $M(k, C)$ (denoted $\mathcal{F}_k$ in the picture) and $S_1(k+1, C)$ (denoted $\mathcal{W}_{k+1}$ in the picture)*

is both difficult theoretically and practically to get right. A cleaner approach is to redesign the kind of penalization used in constructing smoothing splines directly.

10 More on Splines: Regression and Smoothing Splines

10.1 Splines

- Regression splines and smoothing splines are motivated from a different perspective than kernels and local polynomials; in the latter case, 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*. Let's assume that $d = 1$ for simplicity. (We'll stay in the univariate case, for the most part, in this section.) A kth-order spline f is a piecewise polynomial function of degree k that is continuous and has continuous derivatives of orders $1, \ldots, k-1$, at its knot points. Specifically, there are $t_1 < \ldots < t_p$ such that f is a polynomial of degree k on each of the intervals

$$(-\infty, t_1], [t_1, t_2], \ldots, [t_p, \infty)$$

  and $f^{(j)}$ is continuous at $t_1, \ldots, t_p$, for each $j = 0, 1, \ldots, k-1$

- Splines have some special (some might say: amazing!) properties, and they have been a topic of interest among statisticians and mathematicians for a very long time. See

de Boor (1978) for an in-depth coverage. Informally, a spline is a lot smoother than a piecewise polynomial, and so modeling with splines can serve as a way of reducing the variance of fitted estimators. See Figure 6

- 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 $t_1, \ldots, t_p$? The most natural way is to use the *truncated power basis*, $g_1, \ldots, g_{p+k+1}$, defined as

$$\begin{aligned} g_1(x) = 1, \; g_2(x) = x, \; \ldots \; g_{k+1}(x) = x^k, \\ g_{k+1+j}(x) = (x - t_j)_+^k, \quad j = 1, \ldots, p. \end{aligned} \tag{23}$$

  (Here x_+ denotes the positive part of x, i.e., $x_+ = \max\{x, 0\}$.) From this we can see that the space of kth-order splines with knots at $t_1, \ldots, t_p$ has dimension $p + k + 1$

- 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. The key idea: B-splines have local support, so a basis matrix that we form with them (to be defined below) is banded. See de Boor (1978) or the Appendix of Chapter 5 in Hastie et al. (2009) for details

10.2 Regression splines

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

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

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

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

$$G_{ij} = g_j(x_i), \quad i = 1, \ldots, n, \; j = 1, \ldots, p + k + 1,$$

  we can just use least squares to determine the optimal coefficients $\widehat{\beta} = (\widehat{\beta}_1, \ldots, \widehat{\beta}_{p+k+1})$,

$$\widehat{\beta} = \underset{\beta \in \mathbb{R}^{p+k+1}}{\operatorname{argmin}} \; \|y - G\beta\|_2^2,$$

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

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

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

  and regression splines are linear smoothers

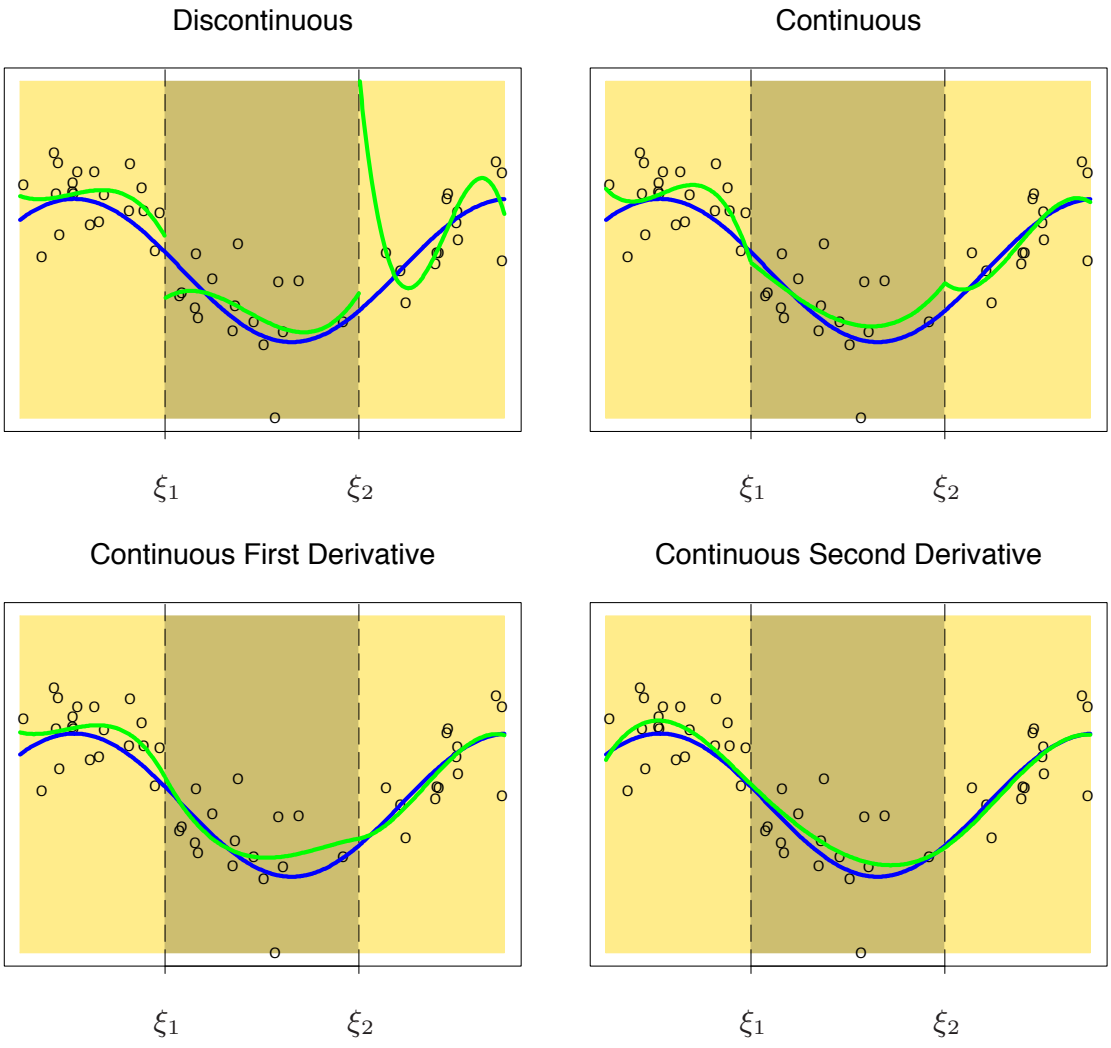

Figure 6: *Illustration of the effects of enforcing continuity at the knots, across various orders of the derivative, for a cubic piecewise polynomial. From Chapter 5 of Hastie et al. (2009)*

- This is a classic method, and can work well provided we choose good knots $t_1, \ldots, t_p$; 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

10.3 Natural splines

- A problem with regression splines is that the estimates tend to display erractic behavior, i.e., they have high variance, at the boundaries of the input domain. (This is the opposite problem to that with kernel smoothing, which had poor bias at the boundaries.) This only gets worse as the polynomial order k gets larger

- A way to remedy this problem is to force the piecewise polynomial function to have a 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 $t_1 < \ldots < t_p$, is a piecewise polynomial function f such that

  - f is a polynomial of degree k on each of $[t_1, t_2], \ldots, [t_{p-1}, t_p]$,
  - f is a polynomial of degree $(k-1)/2$ on $(-\infty, t_1]$ and $[t_p, \infty)$,
  - f is continuous and has continuous derivatives of orders $1, \ldots, k-1$ at $t_1, \ldots, t_p$.

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

- What is the dimension of the span of kth order natural splines with knots at $t_1, \ldots, t_p$? Recall for splines, this was $p+k+1$ (the number of truncated power basis functions). For natural splines, we can compute this dimension by counting:

$$\underbrace{(k+1)\cdot(p-1)}_{a} + \underbrace{\Big(\frac{(k-1)}{2}+1\Big)\cdot 2}_{b} - \underbrace{k\cdot p}_{c} = p.$$

  Above, a is the number of free parameters in the interior intervals $[t_1, t_2], \ldots, [t_{p-1}, t_p]$, b is the number of free parameters in the exterior intervals $(-\infty, t_1], [t_p, \infty)$, and c is the number of constraints at the knots $t_1, \ldots, t_p$. The fact that the total dimension is p 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-spline 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

10.4 Smoothing splines

- Smoothing splines, at the end of the day, are given by a regularized regression over the natural spline basis, placing knots at all inputs $x_1, \ldots, x_n$. They circumvent the problem of knot selection (as they just use the inputs as knots), and they control

for overfitting by shrinking the coefficients of the estimated function (in its basis expansion)

- Interestingly, we can motivate and define a smoothing spline directly from a functional minimization perspective. With inputs $x_1, \ldots, x_n$ lying in an interval $[0, 1]$, the *smoothing spline* estimate $\widehat{f}$, of a given odd integer order $k \geq 0$, is defined as

$$\widehat{f} = \underset{f}{\operatorname{argmin}} \sum_{i=1}^{n} \big(y_i - f(x_i)\big)^2 + \lambda \int_0^1 \big(f^{(m)}(x)\big)^2 \, dx, \quad \text{where } m = (k+1)/2. \tag{24}$$

  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), $i = 1, \ldots, n$, with a penalty term that is large when the mth 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

$$\widehat{f} = \underset{f}{\operatorname{argmin}} \sum_{i=1}^{n} \big(y_i - f(x_i)\big)^2 + \lambda \int_0^1 f''(x)^2 \, dx \tag{25}$$

- Remarkably, it so happens that the minimizer in the general smoothing spline problem (38) is unique, and is a natural kth-order spline with knots at the input points $x_1, \ldots, x_n$! Here we give a proof for the cubic case, $k = 3$, from Green & Silverman (1994) (see also Exercise 5.7 in Hastie et al. (2009))

  The key result can be stated as follows: if $\widetilde{f}$ is any twice differentiable function on $[0, 1]$, and $x_1, \ldots, x_n \in [0, 1]$, then there exists a natural cubic spline f with knots at $x_1, \ldots, x_n$ such that $f(x_i) = \widetilde{f}(x_i)$, $i = 1, \ldots, n$ and

$$\int_0^1 f''(x)^2 \, dx \leq \int_0^1 \widetilde{f}''(x)^2 \, dx.$$

  Note that this would in fact prove that we can restrict our attention in (25) to natural splines with knots at $x_1, \ldots, x_n$

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

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

Consider

$$\begin{aligned}\int_0^1 f''(x)h''(x)\,dx &= f''(x)h'(x)\Big|_0^1 - \int_0^1 f'''(x)h'(x)\;dx \\ &= -\int_{x_1}^{x_n} f'''(x)h'(x)\;dx \\ &= -\sum_{j=1}^{n-1} f'''(x)h(x)\Big|_{x_j}^{x_{j+1}} + \int_{x_1}^{x_n} f^{(4)}(x)h'(x)\;dx \\ &= -\sum_{j=1}^{n-1} f'''(x_j^+)\big(h(x_{j+1}) - h(x_j)\big),\end{aligned}$$

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 $x \le x_1$ and $x \ge x_n$, 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

$$\int_0^1 f''(x)h''(x)\,dx = 0.$$

From this, it follows that

$$\begin{aligned}\int_0^1 \widetilde{f}''(x)^2\,dx &= \int_0^1 \big(f''(x) + h''(x)\big)^2\,dx \\ &= \int_0^1 f''(x)^2\,dx + \int_0^1 h''(x)^2\,dx + 2\int_0^1 f''(x)h''(x)\,dx \\ &= \int_0^1 f''(x)^2\,dx + \int_0^1 h''(x)^2\,dx,\end{aligned}$$

and therefore

$$\int_0^1 f''(x)^2\,dx \le \int_0^1 \widetilde{f}''(x)^2\,dx, \tag{26}$$

with equality if and only if $h''(x) = 0$ for all $x \in [0,1]$. 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 (45) holds strictly except when $\widetilde{f} = f$, so the solution in (25) is uniquely a natural spline with knots at the inputs

10.5 Finite-dimensional form

- The key result presented above tells us that we can choose a basis $\eta_1, \dots, \eta_n$ for the set of kth-order natural splines with knots over $x_1, \dots, x_n$, and reparametrize the problem (38) as

$$\widehat{\beta} = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \sum_{i=1}^n \Big(y_i - \sum_{j=1}^n \beta_j \eta_j(x_i)\Big)^2 + \lambda \int_0^1 \Big(\sum_{j=1}^n \beta_j \eta_j^{(m)}(x)\Big)^2\,dx. \tag{27}$$

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

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

$$N_{ij} = \eta_j(x_i) \quad \text{and} \quad \Omega_{ij} = \int_0^1 \eta_i^{(m)}(x) \eta_j^{(m)}(x)\, dx \quad \text{for } i, j = 1, \ldots, n, \tag{28}$$

the problem in (27) can be written more succintly as

$$\widehat{\beta} = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \ \|y - N\beta\|_2^2 + \lambda \beta \Omega \beta, \tag{29}$$

showing the smoothing spline problem to be a type of generalized ridge regression problem. In fact, the solution in (29) has the explicit form

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

and therefore the fitted values $\widehat{\mu} = (\widehat{f}(x_1), \ldots, \widehat{f}(x_n))$ are

$$\widehat{\mu} = N(N^T N + \lambda \Omega)^{-1} N^T y. \tag{30}$$

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

- A special property of smoothing splines: the fitted values in (30) 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

10.6 Reinsch form

- It is informative to rewrite the fitted values in (30) is what is called Reinsch form,

$$\begin{aligned} \widehat{\mu} &= N(N^T N + \lambda \Omega)^{-1} N^T y \\ &= N\Big(N^T \big(I + \lambda (N^T)^{-1} \Omega N^{-1}\big) N\Big)^{-1} N^T y \\ &= (I + \lambda Q)^{-1} y, \end{aligned} \tag{31}$$

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

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

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

where $D = \operatorname{diag}(d_1, \ldots, d_n)$

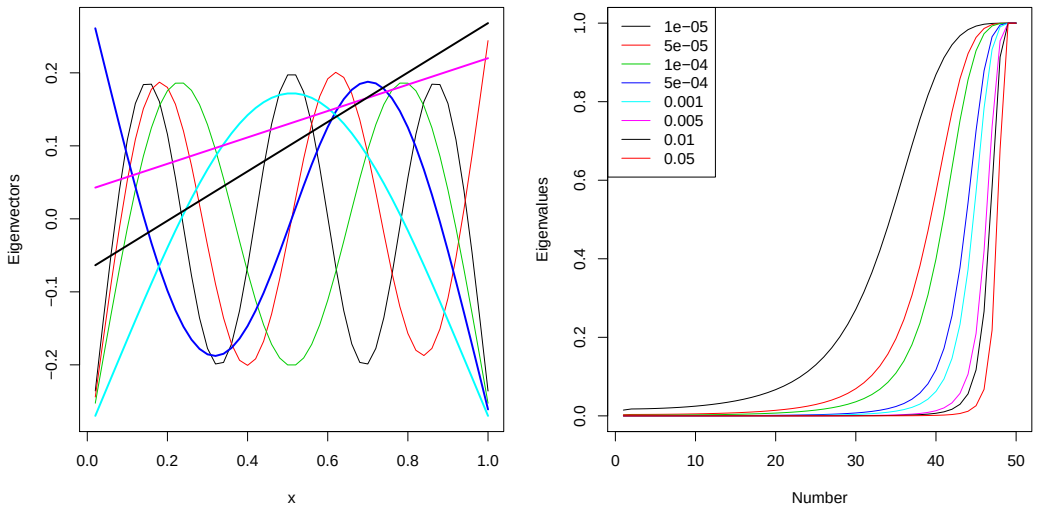

Figure 7: *Eigenvectors and eigenvalues for the Reinsch form of the cubic smoothing spline operator, defined over* $n = 50$ *evenly spaced inputs on* $[0, 1]$. *The left plot shows the bottom 7 eigenvectors of the Reinsch matrix* Q. *We can see that the smaller the eigenvalue, the "smoother" the eigenvector. The right plot shows the weights* $w_j = 1/(1+\lambda d_j)$, $j = 1, \ldots, n$ *implicitly used by the smoothing spline estimator* (32), *over 8 values of* λ. *We can see that when* λ *is larger, the weights decay faster, so the smoothing spline estimator places less weight on the "nonsmooth" eigenvectors*

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

$$\widehat{\mu} = \sum_{j=1}^{n} \frac{u_j^T y}{1 + \lambda d_j} u_j. \tag{32}$$

  Interpretation: smoothing splines perform a regression on the orthonormal basis $u_1, \ldots, u_n \in \mathbb{R}^n$, 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 $u_1, \ldots, u_n$? These are known as the *Demmler-Reinsch basis*, and a lot of their properties can be worked out analytically (**?**). 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. See Figure 7

10.7 Kernel smoothing equivalence

- Something interesting happens when we plot the rows of the smoothing spline matrix S. For evenly spaced inputs, they look like the translations of a kernel! See Figure 8, left plot. For unevenly spaced inputs, the rows still have a kernel shape; now, the bandwidth appears to adapt to the density of the input points: lower density, larger bandwidth. See Figure 8, right plot

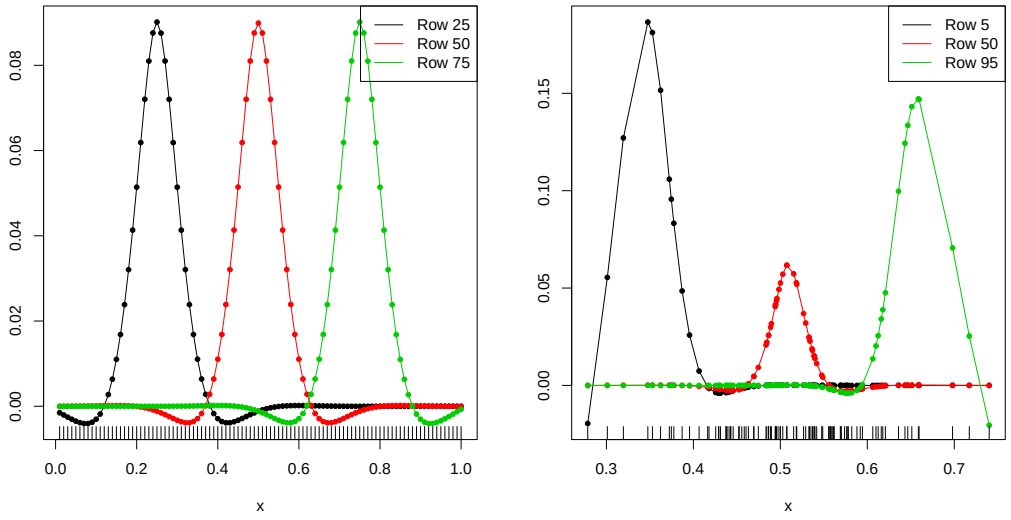

Figure 8: *Rows of the cubic smoothing spline operator S defined over $n = 100$ evenly spaced input points on $[0, 1]$. The left plot shows 3 rows of S (in particular, rows 25, 50, and 75) for $\lambda = 0.0002$. These look precisely like translations of a kernel. The right plot considers a setup where the input points are concentrated around 0.5, and shows 3 rows of S (rows 5, 50, and 95) for the same value of λ. These still look like kernels, but the bandwidth is larger in low-density regions of the inputs*

- What we are seeing is an empirical validation of a beautiful asymptotic result by **?**. It turns out that the cubic smoothing spline estimator is 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 "Silverman kernel":

$$K(x) = \frac{1}{2}\exp(-|x|/\sqrt{2})\sin(|x|/\sqrt{2} + \pi/4), \tag{33}$$

  and a local choice of bandwidth $h(x) = \lambda^{1/4} q(x)^{-1/4}$, where $q(x)$ is the density of the input points. That is, the bandwidth adapts to the local distribution of inputs. See Figure 9 for a plot of the Silverman kernel

- The Silverman kernel is "kind of" a higher-order kernel. It satisfies

$$\int K(x)\,dx = 1, \quad \int x^j K(x)\,dx = 0, \quad j = 1, \ldots, 3, \quad \text{but} \quad \int x^4 K(x)\,dx = -24.$$

  So it lies outside the scope of usual kernel analysis

- There is more recent work that connects smoothing splines of all orders to kernel smoothing. See, e.g., **??**.

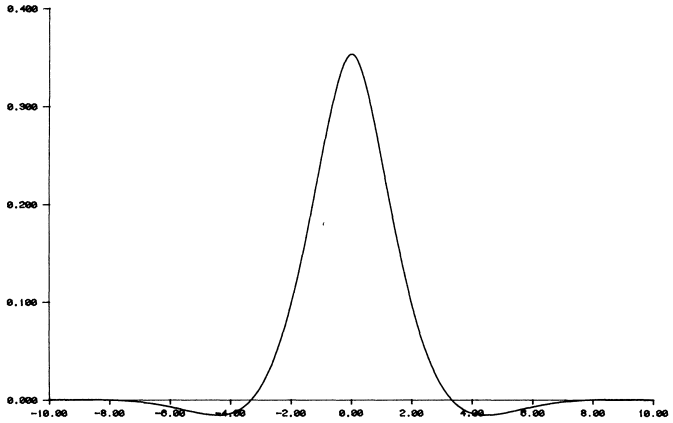

Figure 9: *The Silverman kernel in* (33)*, which is the (asymptotically) equivalent implicit kernel used by smoothing splines. Note that it can be negative. From* **?**

10.8 Error rates

- Define the *Sobolev class* of functions $W_1(m, C)$, for an integer $m \geq 0$ and $C > 0$, to contain all m times differentiable functions $f : \mathbb{R} \to \mathbb{R}$ such that

$$\int \left(f^{(m)}(x)\right)^2 dx \leq C^2.$$

  (The Sobolev class $W_d(m, C)$ in d dimensions can be defined similarly, where we sum over all partial derivatives of order m.)

- Assuming $f_0 \in W_1(m, C)$ for the underlying regression function, where $C > 0$ is a constant, the smoothing spline estimator $\widehat{f}$ in (38) of polynomial order $k = 2m - 1$ with tuning parameter $\lambda \asymp n^{1/(2m+1)} \asymp n^{1/(k+2)}$ satisfies

$$\|\widehat{f} - f_0\|_n^2 \lesssim n^{-2m/(2m+1)} \quad \text{in probability.}$$

  The proof of this result uses much more fancy techniques from empirical process theory (entropy numbers) than the proofs for kernel smoothing. See Chapter 10.1 of van de Geer (2000)

- This rate is seen to be minimax optimal over $W_1(m, C)$ (e.g., Nussbaum (1985)). Also, it is worth noting that the Sobolev $W_1(m, C)$ and Holder $H_1(m, L)$ classes are *equivalent* in the following sense: given $W_1(m, C)$ for a constant $C > 0$, there are $L_0, L_1 > 0$ such that

$$H_1(m, L_0) \subseteq W_1(m, C) \subseteq H_1(m, L_1).$$

  The first containment is easy to show; the second is far more subtle, and is a consequence of the Sobolev embedding theorem. (The same equivalences hold for the d-dimensional versions of the Sobolev and Holder spaces.)

10.9 Multivariate splines

- Splines can be extended to multiple dimensions, in two different ways: *thin-plate splines* and *tensor-product splines*. The former construction is more computationally efficient but more in some sense more limiting; the penalty for a thin-plate spline, of polynomial order $k = 2m - 1$, is

$$\sum_{\alpha_1+\ldots+\alpha_d=m} \int \left| \frac{\partial^m f(x)}{\partial x_1^{\alpha_1} x_2^{\alpha_2} \ldots \partial x_d^{\alpha_d}} \right|^2 dx,$$

  which is rotationally invariant. Both of these concepts are discussed in Chapter 7 of Green & Silverman (1994) (see also Chapters 15 and 20.4 of Gyorfi et al. (2002))

- The multivariate extensions (thin-plate and tensor-product) of splines are highly non-trivial, especially when we compare them to the (conceptually) simple extension of kernel smoothing to higher dimensions. In multiple dimensions, if one wants to study penalized nonparametric estimation, it's (argurably) easier to study reproducing kernel Hilbert space estimators. We'll see, in fact, that this covers smoothing splines (and thin-plate splines) as a special case

References

Bellman, R. (1962), *Adaptive Control Processes*, Princeton University Press.

de Boor, C. (1978), *A Practical Guide to Splines*, Springer.

Devroye, L., Gyorfi, L., & Lugosi, G. (1996), *A Probabilistic Theory of Pattern Recognition*, Springer.

Donoho, D. L. & Johnstone, I. (1998), 'Minimax estimation via wavelet shrinkage', *Annals of Statistics* **26**(8), 879–921.

Fan, J. (1993), 'Local linear regression smoothers and their minimax efficiencies', *The Annals of Statistics* pp. 196–216.

Fan, J. & Gijbels, I. (1996), *Local polynomial modelling and its applications: monographs on statistics and applied probability 66*, Vol. 66, CRC Press.

Green, P. & Silverman, B. (1994), *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach*, Chapman & Hall/CRC Press.

Gyorfi, L., Kohler, M., Krzyzak, A. & Walk, H. (2002), *A Distribution-Free Theory of Nonparametric Regression*, Springer.

Hastie, T. & Tibshirani, R. (1990), *Generalized Additive Models*, Chapman and Hall.

Hastie, T., Tibshirani, R. & Friedman, J. (2009), *The Elements of Statistical Learning; Data Mining, Inference and Prediction*, Springer. 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.

Kim, S.-J., Koh, K., Boyd, S. & Gorinevsky, D. (2009), 'ℓ_1 trend filtering', *SIAM Review* **51**(2), 339–360.

Lin, Y. & Zhang, H. H. (2006), 'Component selection and smoothing in multivariate nonparametric regression', *Annals of Statistics* **34**(5), 2272–2297.

Mallat, S. (2008), *A wavelet tour of signal processing*, Academic Press. Third edition.

Mammen, E. & van de Geer, S. (1997), 'Locally apadtive regression splines', *Annals of Statistics* **25**(1), 387–413.

Nussbaum, M. (1985), 'Spline smoothing in regression models and asymptotic efficiency in l_2', *Annals of Statistics* **13**(3), 984–997.

Raskutti, G., Wainwright, M. & Yu, B. (2012), 'Minimax-optimal rates for sparse additive models over kernel classes via convex programming', *Journal of Machine Learning Research* **13**, 389–427.

Ravikumar, P., Liu, H., Lafferty, J. & Wasserman, L. (2009), 'Sparse additive models', *Journal of the Royal Statistical Society: Series B* **75**(1), 1009–1030.

Scholkopf, B. & Smola, A. (2002), 'Learning with kernels'.

Simonoff, J. (1996), *Smoothing Methods in Statistics*, Springer.

Steidl, G., Didas, S. & Neumann, J. (2006), 'Splines in higher order TV regularization', *International Journal of Computer Vision* **70**(3), 214–255.

Stone, C. (1985), 'Additive regression models and other nonparametric models', *Annals of Statistics* **13**(2), 689–705.

Tibshirani, R. J. (2014), 'Adaptive piecewise polynomial estimation via trend filtering', *Annals of Statistics* **42**(1), 285–323.

Tsybakov, A. (2009), *Introduction to Nonparametric Estimation*, Springer.

van de Geer, S. (2000), *Empirical Processes in M-Estimation*, Cambdrige University Press.

Wahba, G. (1990), *Spline Models for Observational Data*, Society for Industrial and Applied Mathematics.

Wang, Y., Smola, A. & Tibshirani, R. J. (2014), 'The falling factorial basis and its statistical properties', *International Conference on Machine Learning* **31**.

Wasserman, L. (2006), *All of Nonparametric Statistics*, Springer.

Yang, Y. (1999), 'Nonparametric classification–Part I: Rates of convergence', *IEEE Transactions on Information Theory* **45**(7), 2271–2284.

Appendix: Locally adaptive estimators

10.10 Locally adaptive regression splines

Locally adaptive regression splines (Mammen & van de Geer 1997), as their name suggests, can be viewed as variant of smoothing splines that exhibit better local adaptivity. For a given integer order $k \geq 0$, the estimate is defined as

$$\widehat{m} = \underset{f}{\operatorname{argmin}} \sum_{i=1}^{n} \big(Y_i - m(X_i)\big)^2 + \lambda \operatorname{TV}(f^{(k)}). \tag{34}$$

The minimization domain is infinite-dimensional, the space of all functions for which the criterion is finite

Another remarkable variational result, similar to that for smoothing splines, shows that (34) has a kth order spline as a solution (Mammen & van de Geer 1997). This *almost* turns the minimization into a finite-dimensional one, but there is one catch: the knots of this kth-order spline are generally not known, i.e., they need not coincide with the inputs $x_1, \ldots, x_n$. (When $k = 0, 1$, they do, but in general, they do not)

To deal with this issue, we can redefine the locally adaptive regression spline estimator to be

$$\widehat{m} = \underset{f \in \mathcal{G}_k}{\operatorname{argmin}} \sum_{i=1}^{n} \big(Y_i - m(X_i)\big)^2 + \lambda \operatorname{TV}(f^{(k)}), \tag{35}$$

i.e., we restrict the domain of minimization to be $\mathcal{G}_k$, the space of kth-order spline functions with knots in T_k, where T_k is a subset of $\{x_1, \ldots, x_n\}$ of size $n-k-1$. The precise definition of T_k is not important; it is just given by trimming away $k+1$ boundary points from the inputs

As we already know, the space $\mathcal{G}_k$ of kth-order splines with knots in T_k has dimension $|T_k| + k + 1 = n$. Therefore we can choose a basis $g_1, \ldots, g_n$ for the functions in $\mathcal{G}_k$, and the problem in (35) becomes one of finding the coefficients in this basis expansion,

$$\widehat{\beta} = \underset{f \in \mathcal{G}_k}{\operatorname{argmin}} \sum_{i=1}^{n} \Big(Y_i - \sum_{j=1}^{n} \beta_j g_j(X_i)\Big)^2 + \lambda \operatorname{TV}\Big\{\Big(\sum_{j=1}^{n} \beta_j g_j(X_i)\Big)^{(k)}\Big\}, \tag{36}$$

and then we have $\widehat{m}(x) = \sum_{j=1}^{n} \widehat{\beta}_j g_j(x)$

Now define the basis matrix $G \in \mathbb{R}^{n \times n}$ by

$$G_{ij} = g_j(X_i), \quad i = 1, \ldots, n.$$

Suppose we choose $g_1, \ldots, g_n$ to be the truncated power basis. Denoting $T_k = \{t_1, \ldots, t_{n-k-1}\}$, we compute

$$\Big(\sum_{j=1}^{n} \beta_j g_j(X_i)\Big)^{(k)} = k. + k. \sum_{j=k+2}^{n} \beta_j 1\{x \geq t_{j-k-1}\},$$

and so

$$\operatorname{TV}\Big\{\Big(\sum_{j=1}^{n} \beta_j g_j(X_i)\Big)^{(k)}\Big\} = k. \sum_{j=k+2}^{n} |\beta_j|.$$

Hence the locally adaptive regression spline problem (36) can be expressed as

$$\widehat{\beta} = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \ \|y - G\beta\|_2^2 + \lambda k. \sum_{i=k+2}^{n} |\beta_i|. \tag{37}$$

This is a lasso regression problem on the truncated power basis matrix G, with the first $k+1$ coefficients (those corresponding to the pure polynomial functions, in the basis expansion) left unpenalized

This reveals a key difference between the locally adaptive regression splines (37) (originally, problem (35)) and the smoothing splines (29) (originally, problem

$$\widehat{m} = \underset{f}{\operatorname{argmin}} \sum_{i=1}^{n} \big(Y_i - m(X_i)\big)^2 + \lambda \int_0^1 \big(f^{(m)}(x)\big)^2 \, dx, \quad \text{where } m = (k+1)/2. \tag{38}$$

In the first problem, the total variation penalty is translated into an ℓ_1 penalty on the coefficients of the truncated power basis, and hence this acts a knot selector for the estimated function. That is, at the solution in (37), the estimated spline has knots at a subset of T_k (at a subset of the input points $x_1, \ldots, x_n$), with fewer knots when λ is larger. In contrast, recall, at the smoothing spline solution in (29), the estimated function has knots at each of the inputs $x_1, \ldots, x_n$. This is a major difference between the ℓ_1 and ℓ_2 penalties

From a computational perspective, the locally adaptive regression spline problem in (37) is actually a lot harder than the smoothing spline problem in (29). Recall that the latter reduces to solving a single banded linear system, which takes $O(n)$ operations. On the other hand, fitting locally adaptive regression splines in (37) requires solving a lasso problem with a dense $n \times n$ regression matrix G; this takes something like $O(n^3)$ operations. So when $n = 10,000$, there is a big difference between the two.

There is a tradeoff here, as with extra computation comes much improved local adaptivity of the fits. See Figure 10 for an example. Theoretically, when $m_0 \in M(k, C)$ for a constant $C > 0$, Mammen & van de Geer (1997) show the locally adaptive regression spline estimator, denoted $\widehat{m}^{\text{lrs}}$, with $\lambda \asymp n^{1/(2k+3)}$, satisfies

$$\|\widehat{m}^{\text{lrs}} - m_0\|_n^2 \lesssim n^{-(2k+2)/(2k+3)} \quad \text{in probability},$$

so (like wavelets) it achieves the minimax optimal rate over $n^{-(2k+2)/(2k+3)}$. In this regard, as we discussed previously, they actually have a big advantage over any linear smoother (not just smoothing splines)

10.11 Trend filtering

At a high level, you can think of trend filtering as computationally efficient version of locally adaptive regression splines, though their original construction (Steidl et al. 2006, Kim et al. 2009) comes from a fairly different perspective. We will begin by describing their connection to locally adaptive regression splines, following Tibshirani (2014)

Revisit the formulation of locally adaptive regression splines in (35), where the minimization domain is $\mathcal{G}_k = \text{span}\{g_1, \ldots, g_n\}$, and $g_1, \ldots, g_n$ are the kth-order truncated power basis

$$\begin{aligned} &g_1(x) = 1, \ g_2(x) = x, \ \ldots \ g_{k+1}(x) = x^k, \\ &g_{k+1+j}(x) = (x - t_j)_+^k, \quad j = 1, \ldots, p. \end{aligned} \tag{39}$$

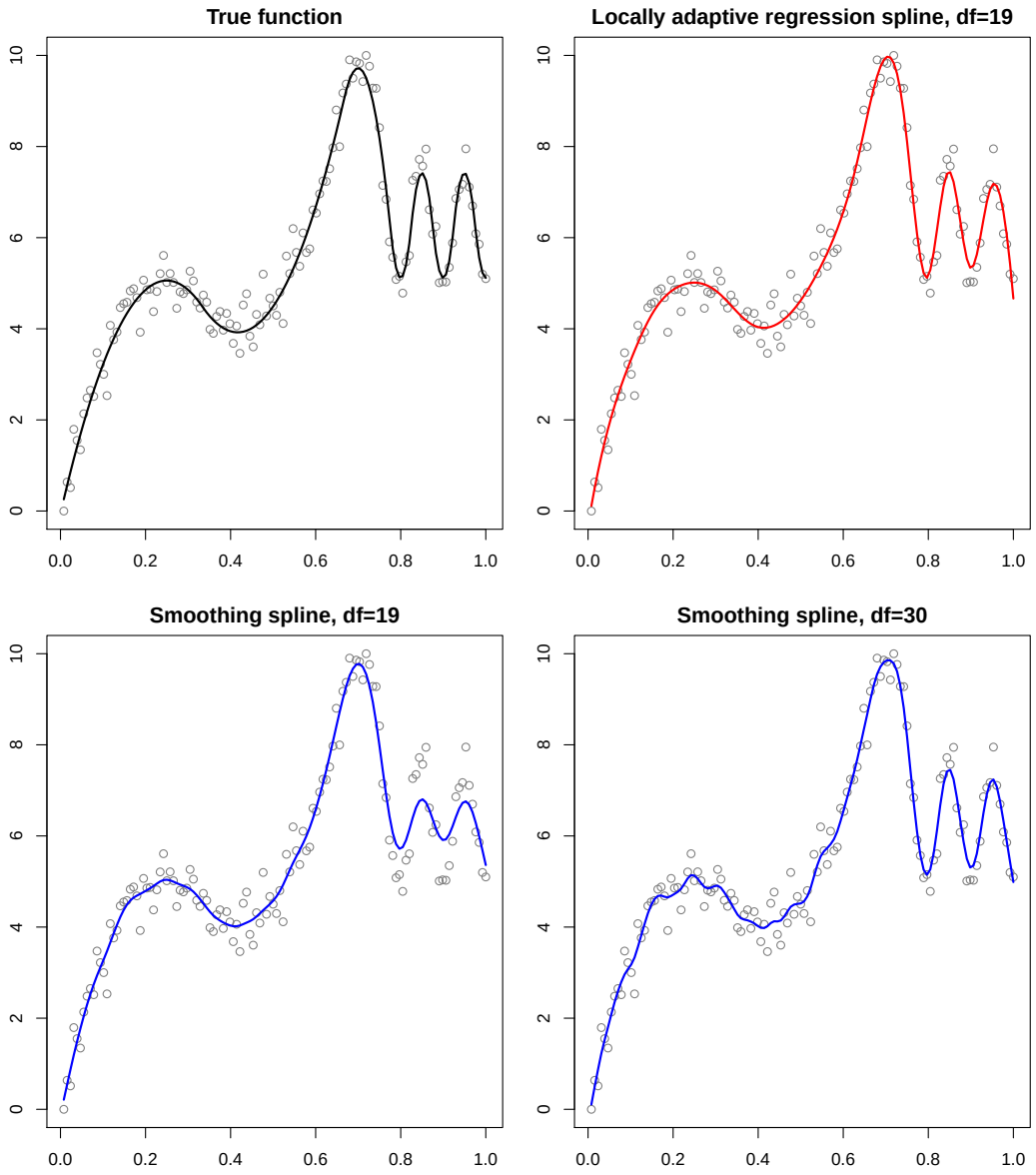

Figure 10: *The top left plot shows a simulated true regression function, which has inhomogeneous smoothness: smoother towards the left part of the domain, wigglier towards the right. The top right plot shows the locally adaptive regression spline estimate with 19 degrees of freedom; notice that it picks up the right level of smoothness throughout. The bottom left plot shows the smoothing spline estimate with the same degrees of freedom; it picks up the right level of smoothness on the left, but is undersmoothed on the right. The bottom right panel shows the smoothing spline estimate with 33 degrees of freedom; now it is appropriately wiggly on the right, but oversmoothed on the left. Smoothing splines cannot simultaneously represent different levels of smoothness at different regions in the domain; the same is true of any linear smoother*

having knots in a set $T_k \subseteq \{X_1, \ldots X_n\}$ with size $|T_k| = n - k - 1$. The *trend filtering* problem is given by replacing $\mathcal{G}_k$ with a different function space,

$$\widehat{m} = \underset{f \in \mathcal{H}_k}{\operatorname{argmin}} \sum_{i=1}^{n} \big(Y_i - m(X_i)\big)^2 + \lambda \operatorname{TV}(f^{(k)}), \tag{40}$$

where the new domain is $\mathcal{H}_k = \operatorname{span}\{h_1, \ldots, h_n\}$. Assuming that the input points are ordered, $x_1 < \ldots < x_n$, the functions $h_1, \ldots, h_n$ are defined by

$$\begin{aligned} h_j(x) &= \prod_{\ell=1}^{j-1} (x - x_\ell), \quad j = 1, \ldots, k+1, \\ h_{k+1+j}(x) &= \prod_{\ell=1}^{k} (x - x_{j+\ell}) \cdot 1\{x \geq x_{j+k}\}, \quad j = 1, \ldots, n-k-1. \end{aligned} \tag{41}$$

(Our convention is to take the empty product to be 1, so that $h_1(x) = 1$.) These are dubbed the *falling factorial basis*, and are piecewise polynomial functions, taking an analogous form to the truncated power basis functions in (10.11). Loosely speaking, they are given by replacing an rth-order power function in the truncated power basis with an appropriate r-term product, e.g., replacing x^2 with $(x - x_2)(x - x_1)$, and $(x - t_j)^k$ with $(x - x_{j+k})(x - x_{j+k-1}) \cdot \ldots, (x - x_{j+1})$

Defining the falling factorial basis matrix

$$H_{ij} = h_j(X_i), \quad i, j = 1, \ldots, n,$$

it is now straightforward to check that the proposed problem of study, trend filtering in (40), is equivalent to

$$\widehat{\beta} = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \ \|y - H\beta\|_2^2 + \lambda k. \sum_{i=k+2}^{n} |\beta_i|. \tag{42}$$

This is still a lasso problem, but now in the falling factorial basis matrix H. Compared to the locally adaptive regression spline problem (37), there may not seem to be much of a difference here—like G, the matrix H is dense, and solving (42) would be slow. So why did we go to all the trouble of defining trend filtering, i.e., introducing the somewhat odd basis $h_1, \ldots, h_n$ in (41)?

The usefulness of trend filtering (42) is seen after reparametrizing the problem, by inverting H. Let $\theta = H\beta$, and rewrite the trend filtering problem as

$$\widehat{\theta} = \underset{\theta \in \mathbb{R}^n}{\operatorname{argmin}} \ \|y - \theta\|_2^2 + \lambda \|D\theta\|_1, \tag{43}$$

where $D \in \mathbb{R}^{(n-k-1) \times n}$ denotes the last $n - k - 1$ rows of $k. \cdot H^{-1}$. Explicit calculation shows that D is a banded matrix (Tibshirani 2014, Wang et al. 2014). For simplicity of exposition, consider the case when $X_i = i$, $i = 1, \ldots, n$. Then, e.g., the first 3 orders of difference operators are:

$$D = \begin{bmatrix} -1 & 1 & 0 & \ldots \\ 0 & -1 & 1 & \ldots \\ \vdots & & & \end{bmatrix}, \quad \text{when } k = 0$$

$$D = \begin{bmatrix} 1 & -2 & 1 & 0 & \ldots \\ 0 & 1 & -2 & 1 & \ldots \\ 0 & 0 & 1 & -2 & \ldots \\ \vdots & & & & \end{bmatrix}, \quad \text{when } k = 1$$

$$D = \begin{bmatrix} -1 & 3 & -3 & 1 & \ldots \\ 0 & -1 & 3 & -3 & \ldots \\ 0 & 0 & -1 & 3 & \ldots \\ \vdots & & & & \end{bmatrix}, \quad \text{when } k = 2.$$

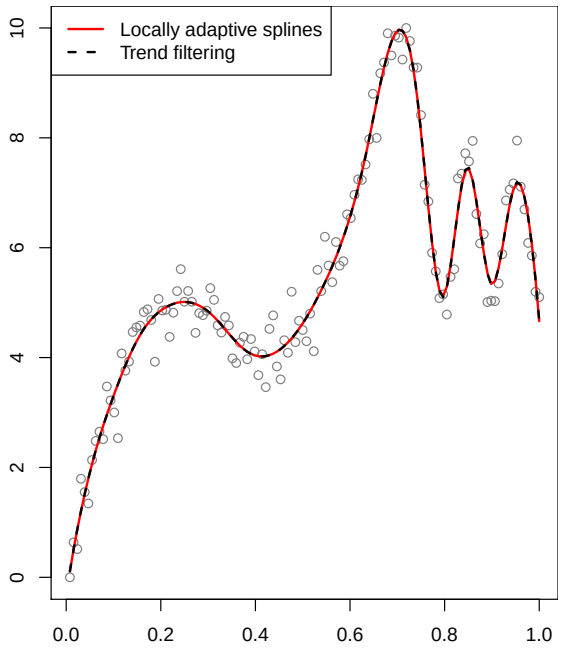

Figure 11: *Trend filtering and locally adaptive regression spline estimates, fit on the same data set as in Figure 10. The two are tuned at the same level, and the estimates are visually indistinguishable*

One can hence interpret D as a type of discrete derivative operator, of order $k+1$. This also suggests an intuitive interpretation of trend filtering (43) as a discrete approximation to the original locally adaptive regression spline problem in (34)

The bandedness of D means that the trend filtering problem (43) can be solved efficiently, in close to linear time (complexity $O(n^{1.5})$ in the worst case). Thus trend filtering estimates are much easier to fit than locally adaptive regression splines

But what of their statistical relevancy? Did switching over to the falling factorial basis (41) wreck the local adaptivity properties that we cared about in the first place? Fortunately, the answer is no, and in fact, trend filtering and locally adaptive regression spline estimates are extremely hard to distinguish in practice. See Figure 11

Moreover, Tibshirani (2014), Wang et al. (2014) prove that the estimates from trend filtering and locally adaptive regression spline estimates, denoted $\widehat{m}^{\mathrm{tf}}$ and $\widehat{m}^{\mathrm{lrs}}$, respectively, when the tuning parameter λ for each scales as $n^{1/(2k+3)}$, satisfy

$$\|\widehat{m}^{\mathrm{tv}} - \widehat{m}^{\mathrm{lrs}}\|_n^2 \lesssim n^{-(2k+2)/(2k+3)} \quad \text{in probability.}$$

This coupling shows that trend filtering converges to the underlying function m_0 at the rate $n^{-(2k+2)/(2k+3)}$ whenever locally adaptive regression splines do, making them also minimax optimal over $M(k, C)$. In short, trend filtering offers provably significant improvements over linear smoothers, with a computational cost that is not too much steeper than a single banded linear system solve

10.12 Proof of (9)

Let

$$m_h(x) = \frac{\sum_{i=1}^n m(X_i) I(\|X_i - x\| \leq h)}{nP_n(B(x,h))}.$$

Let $A_n = \{P_n(B(x,h)) > 0\}$. When A_n is true,

$$\mathbb{E}\left((\widehat{m}_h(x) - m_h(x))^2 \;\middle|\; X_1, \ldots, X_n \right) = \frac{\sum_{i=1}^n \mathrm{Var}(Y_i|X_i) I(\|X_i - x\| \leq h)}{n^2 P_n^2(B(x,h))} \leq \frac{\sigma^2}{nP_n(B(x,h))}.$$

Since $m \in \mathcal{M}$, we have that $|m(X_i) - m(x)| \leq L\|X_i - x\| \leq Lh$ for $X_i \in B(x,h)$ and hence

$$|m_h(x) - m(x)|^2 \leq L^2 h^2 + m^2(x) I_{A_n(x)^c}.$$

Therefore,

$$\begin{aligned}
\mathbb{E}\int (\widehat{m}_h(x) - m(x))^2 dP(x) &= \mathbb{E}\int (\widehat{m}_h(x) - m_h(x))^2 dP(x) + \mathbb{E}\int (m_h(x) - m(x))^2 dP(x) \\
&\leq \mathbb{E}\int \frac{\sigma^2}{nP_n(B(x,h))} I_{A_n(x)} dP(x) + L^2 h^2 + \int m^2(x) \mathbb{E}(I_{A_n(x)^c}) dP(x). \qquad (44)
\end{aligned}$$

To bound the first term, let $Y = nP_n(B(x,h))$. Note that $Y \sim \text{Binomial}(n,q)$ where $q = \mathbb{P}(X \in B(x,h))$. Now,

$$\begin{aligned}
\mathbb{E}\left(\frac{I(Y>0)}{Y} \right) &\leq \mathbb{E}\left(\frac{2}{1+Y} \right) = \sum_{k=0}^n \frac{2}{k+1} \binom{n}{k} q^k (1-q)^{n-k} \\
&= \frac{2}{(n+1)q} \sum_{k=0}^n \binom{n+1}{k+1} q^{k+1} (1-q)^{n-k} \\
&\leq \frac{2}{(n+1)q} \sum_{k=0}^{n+1} \binom{n+1}{k} q^k (1-q)^{n-k+1} \\
&= \frac{2}{(n+1)q} (q + (1-q))^{n+1} = \frac{2}{(n+1)q} \leq \frac{2}{nq}.
\end{aligned}$$

Therefore,

$$\mathbb{E}\int \frac{\sigma^2 I_{A_n(x)}}{nP_n(B(x,h))} dP(x) \leq 2\sigma^2 \int \frac{dP(x)}{nP(B(x,h))}.$$

We may choose points $z_1, \ldots, z_M$ such that the support of P is covered by $\bigcup_{j=1}^M B(z_j, h/2)$ where $M \leq c_2/(nh^d)$. Thus,

$$\begin{aligned}
\int \frac{dP(x)}{nP(B(x,h))} &\leq \sum_{j=1}^M \int \frac{I(z \in B(z_j, h/2))}{nP(B(x,h))} dP(x) \leq \sum_{j=1}^M \int \frac{I(z \in B(z_j, h/2))}{nP(B(z_j, h/2))} dP(x) \\
&\leq \frac{M}{n} \leq \frac{c_1}{nh^d}.
\end{aligned}$$

The third term in (44) is bounded by

$$\begin{aligned}
\int m^2(x)\mathbb{E}(I_{A_n(x)^c})dP(x) &\leq \sup_x m^2(x)\int (1-P(B(x,h)))^n dP(x) \\
&\leq \sup_x m^2(x)\int e^{-nP(B(x,h))}dP(x) \\
&= \sup_x m^2(x)\int e^{-nP(B(x,h))}\frac{nP(B(x,h))}{nP(B(x,h))}dP(x) \\
&\leq \sup_x m^2(x)\sup_u(ue^{-u})\int \frac{1}{nP(B(x,h))}dP(x) \\
&\leq \sup_x m^2(x)\sup_u(ue^{-u})\frac{c_1}{nh^d} = \frac{c_2}{nh^d}.
\end{aligned}$$

10.13 Proof of the Spline Lemma

The key result can be stated as follows: if $\widetilde{f}$ is any twice differentiable function on $[0,1]$, and $x_1,\ldots,x_n \in [0,1]$, then there exists a natural cubic spline f with knots at $x_1,\ldots,x_n$ such that $m(X_i) = \widetilde{f}(X_i)$, $i = 1,\ldots,n$ and

$$\int_0^1 f''(x)^2\,dx \leq \int_0^1 \widetilde{f}''(x)^2\,dx.$$

Note that this would in fact prove that we can restrict our attention in (25) to natural splines with knots at $x_1,\ldots,x_n$.

The natural spline basis with knots at $x_1,\ldots,x_n$ is n-dimensional, so given any n points $z_i = \widetilde{f}(X_i)$, $i = 1,\ldots,n$, we can always find a natural spline f with knots at $x_1,\ldots,x_n$ that satisfies $m(X_i) = z_i$, $i = 1,\ldots,n$. Now define

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

Consider

$$\begin{aligned}
\int_0^1 f''(x)h''(x)\,dx &= f''(x)h'(x)\Big|_0^1 - \int_0^1 f'''(x)h'(x)\;dx \\
&= -\int_{x_1}^{x_n} f'''(x)h'(x)\;dx \\
&= -\sum_{j=1}^{n-1} f'''(x)h(x)\Big|_{x_j}^{x_{j+1}} + \int_{x_1}^{x_n} f^{(4)}(x)h'(x)\;dx \\
&= -\sum_{j=1}^{n-1} f'''(x_j^+)\big(h(x_{j+1}) - h(x_j)\big),
\end{aligned}$$

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}), $j = 1,\ldots,n-1$, 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 $j = 1, \ldots, n$, we have

$$\int_0^1 f''(x) h''(x)\, dx = 0.$$

From this, it follows that

$$\begin{aligned}
\int_0^1 \widetilde{f}''(x)^2\, dx &= \int_0^1 \big(f''(x) + h''(x)\big)^2\, dx \\
&= \int_0^1 f''(x)^2\, dx + \int_0^1 h''(x)^2\, dx + 2\int_0^1 f''(x) h''(x)\, dx \\
&= \int_0^1 f''(x)^2\, dx + \int_0^1 h''(x)^2\, dx,
\end{aligned}$$

and therefore

$$\int_0^1 f''(x)^2\, dx \leq \int_0^1 \widetilde{f}''(x)^2\, dx, \tag{45}$$

with equality if and only if $h''(x) = 0$ for all $x \in [0, 1]$. Note that $h'' = 0$ implies that h must be linear, and since we already know that $h(x_j) = 0$ for all $j = 1, \ldots, n$, this is equivalent to $h = 0$. In other words, the inequality (45) holds strictly except when $\widetilde{f} = f$, so the solution in (25) is uniquely a natural spline with knots at the inputs.