Nonparametric Regression
10716: Advanced Machine Learning
Pradeep Ravikumar (amending notes from Larry Wasserman, Ryan Tibshirani)

1 Introduction

Given a sample (X_1, Y_1), ..., (X_n, Y_n), where $X_i \in \mathbb{R}^d$ and $Y_i \in \mathbb{R}$, we consider the task of estimating the regression function

$$m(x) = \mathbb{E}(Y|X = x) \tag{1}$$

without making parametric assumptions (such as linearity) about the regression function $m(x)$. Estimating m is called *nonparametric regression* or *smoothing* (Härdle et al. 2012, Wasserman 2006). We can equivalently write

$$Y = m(X) + \epsilon$$

where $\mathbb{E}(\epsilon|X) = 0$. This follows since for $\epsilon = Y - m(X)$ and $\mathbb{E}(\epsilon|X)) = 0$ iff $m(X) = \mathbb{E}[Y|X]$

A related problem is *nonparametric prediction*. Given a pair (X, Y), we want to predict Y from X. The optimal predictor (under squared error loss) is the regression function $m(X)$. Hence, estimating m is of interest for its own sake and for the purposes of prediction.

Example 1 *Figure 1 shows data on bone mineral density. The plots show the relative change in bone density over two consecutive visits, for men and women. The smooth estimates of the regression functions suggest that a growth spurt occurs two years earlier for females. In this example, Y is change in bone mineral density and X is age.*

Example 2 *Figure 2 shows an analysis of some diabetes data from Efron, Hastie, Johnstone and Tibshirani (2004). The outcome Y is a measure of disease progression after one year. We consider four covariates (ignoring for now, six other variables): age, bmi (body mass index), and two variables representing blood serum measurements. A nonparametric regression model in this case takes the form*

$$Y = m(x_1, x_2, x_3, x_4) + \epsilon. \tag{2}$$

A simpler, but less general model, is the additive model

$$Y = m_1(x_1) + m_2(x_2) + m_3(x_3) + m_4(x_4) + \epsilon. \tag{3}$$

Figure 2 shows the four estimated functions $\widehat{m}_1, \widehat{m}_2, \widehat{m}_3$ and $\widehat{m}_4$.

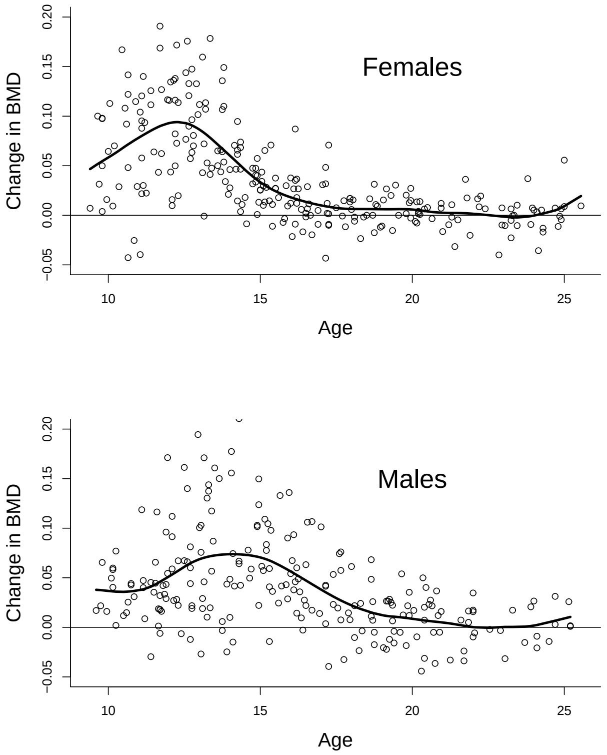

Figure 1: Bone Mineral Density Data

Notation. We use $m(x)$ to denote the regression function. Often we assume that X_i has a density denoted by $p(x)$. The support of the distribution of X_i is denoted by $\mathcal{X}$. We assume that $\mathcal{X}$ is a compact subset of $\mathbb{R}^d$. Recall that the trace of a square matrix A is denoted by $\mathrm{tr}(A)$ and is defined to be the sum of the diagonal elements of A.

2 Loss Functions, The Bias–Variance Tradeoff

Let $\widehat{m}(x)$ be an estimate of $m(x)$. Here, a common loss is the integrated squared loss:

$$L(\widehat{m}, m) = \int \Big((\widehat{m}(x) - m(x))^2\Big) dP(x),$$

where we use a weighted integral wrt the data distribution P. This could also be viewed as the expectation:

$$L(\widehat{m}, m) = \mathbb{E}_{X \sim P}[(\widehat{m}(X) - m(X))^2].$$

The corresponding risk is also known as the integrated mean squared error given by:

$$R(\widehat{m}, m) = \mathbb{E} \int \Big((\widehat{m}(x) - m(x))^2\Big) dP(x) = \mathbb{E}[(\widehat{m}(X) - m(X))^2], \tag{4}$$

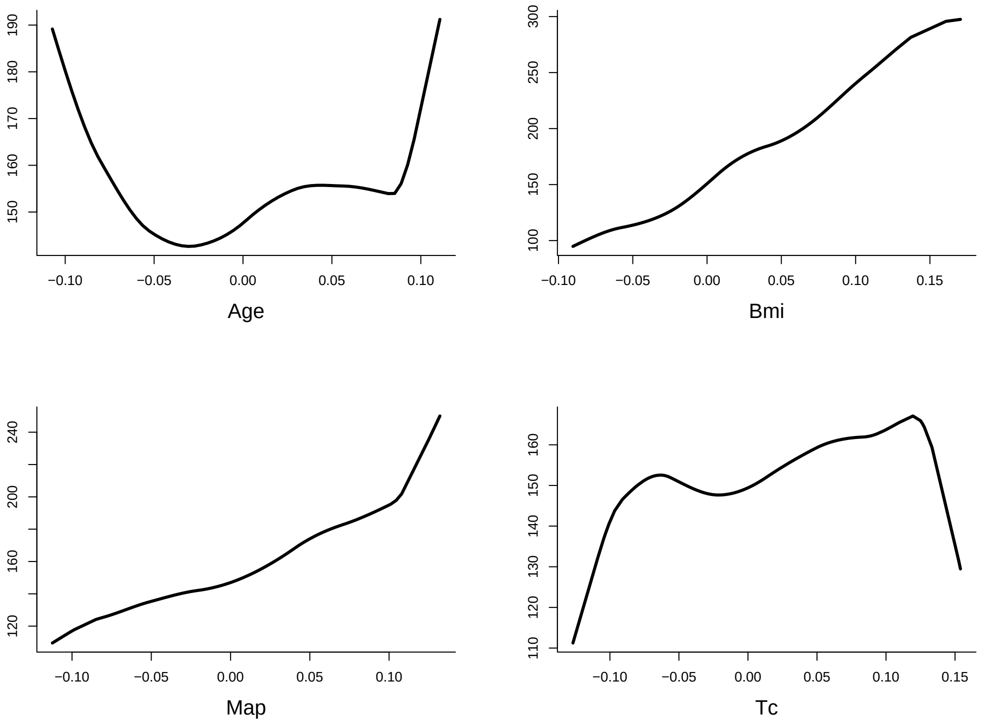

Figure 2: Diabetes Data

where the expectation is wrt the random input X as well as the data underlying $\widehat{m}$. Sometimes, we might also be interested in the *predictive risk*

$$R_p(m, \widehat{m}) = \mathbb{E}((Y - \widehat{m}(X))^2) \tag{5}$$

where (X, Y) denotes a new observation. It follows that

$$\begin{aligned} R_p(m, \widehat{m}) &= \sigma^2 + \mathbb{E}[(\widehat{m}(X) - m(X))^2], \qquad (6) \\ &= \sigma^2 + \int b_n^2(x) dP(x) + \int v_n(x) dP(x) \qquad (7) \end{aligned}$$

where $b_n(x) = \mathbb{E}(\widehat{m}(x)) - m(x)$ is the bias and $v(x) = \mathrm{Var}(\widehat{m}(x))$ is the variance.

The estimator $\widehat{m}$ typically involves smoothing the data in some way. The main challenge is to determine how much smoothing to do. When the data are oversmoothed, the bias term is large and the variance is small. When the data are undersmoothed the opposite is true. This is called the *bias–variance tradeoff*. Minimizing risk corresponds to balancing bias and variance.

An estimator $\widehat{m}$ is *consistent* if

$$\|\widehat{m} - m\| \overset{P}{\to} 0. \tag{8}$$

When unspecified, the function norm $\|\cdot\|$ will typically mean the L_2 norm $\|\cdot\|_2$ in terms of P_X, acting on functions $m : \mathbb{R}^d \to \mathbb{R}$, by

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

The *minimax risk* over a set of functions $\mathcal{M}$ is

$$R_n(\mathcal{M}) = \inf_{\widehat{m}} \sup_{m\in\mathcal{M}} R(m, \widehat{m}) \tag{9}$$

and an estimator is *minimax* if its risk is equal to the minimax risk. We say that $\widehat{m}$ is *rate optimal* if

$$R(m, \widehat{m}) \asymp R_n(\mathcal{M}). \tag{10}$$

Typically the minimax rate is of the form $n^{-C/(C+d)}$ for some $C > 0$.

2.1 Outline

We are again going to be considering two broad categories of non-parametric regression estimators:

- Partition based estimators:
  - Among hard-partition based estimators, we will discuss piece-wise constant partitition estimators, and splines (or piece-wise polynomial estimators)
  - Among soft-partition based estimators, we will discuss K-Nearest Neighbor Regression, and Smoothing Kernel Regression Estimators
- Function Space based estimators:
  - Basis/Dictionary Series Estimators
  - Mercer Kernel (or Reproducing Kernel Hilbert Space (RKHS)) Regression Estimators
  - Wavelets

3 Partitions and Trees

Simple and interpretable estimators can be derived by partitioning the range of X. Let $\Pi_n = \{A_1, \ldots, A_N\}$ be a partition of $\mathcal{X}$ and define

$$\widehat{m}(x) = \sum_{j=1}^{N} \overline{Y}_j\, I(x \in A_j)$$

where $\overline{Y}_j = n_j^{-1} \sum_{i=1}^n Y_i I(X_i \in A_j)$ is the average of the Y_i's in A_j and $n_j = \#\{X_i \in A_j\}$. (We define $\overline{Y}_j$ to be 0 if $n_j = 0$.)

The simplest partition is based on cubes. Suppose that $\mathcal{X} = [0,1]^d$. Then we can partition $\mathcal{X}$ into $N = k^d$ cubes with lengths of size $h = 1/k$. Thus, $N = (1/h)^d$. The smoothing parameter is h.

Theorem 3 *Let $\widehat{m}(x)$ be the partition estimator. Suppose that*

$$m \in \mathcal{M} = \left\{ m : \ |m(x) - m(z)| \leq L\|x - z\|, \quad x, z, \in \mathbb{R}^d \right\} \tag{11}$$

and that $\mathrm{Var}(Y|X = x) \leq \sigma^2 < \infty$ *for all x. Then*

$$\mathbb{E}\|\widehat{m} - m\|_P^2 \leq c_1 h^2 + \frac{c_2}{nh^d}. \tag{12}$$

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

$$\mathbb{E}\|\widehat{m} - m\|_P^2 \leq \frac{c}{n^{2/(d+2)}}. \tag{13}$$

A *regression tree* is a partition estimator of the form

$$m(x) = \sum_{m=1}^M c_m I(x \in R_m) \tag{14}$$

where $c_1, \ldots, c_M$ are constants and $R_1, \ldots, R_M$ are disjoint rectangles that partition the space of covariates and whose sides are parallel to the coordinate axes. The model is fit in a greedy, recursive manner that can be represented as a tree; hence the name.

Denote a generic covariate value by $x = (x_1, \ldots, x_j, \ldots, x_d)$. The covariate for the i^{th} observation is $X_i = (X_{i1}, \ldots, X_{ij}, \ldots, X_{id})$. Given a covariate j and a split point s we define the rectangles $R_1 = R_1(j,s) = \{x : \ x_j \leq s\}$ and $R_2 = R_2(j,s) = \{x : \ x_j > s\}$ where, in this expression, x_j refers the the j^{th} covariate not the j^{th} observation. Then we take c_1 to be the average of all the Y_i's such that $X_i \in R_1$ and c_2 to be the average of all the Y_i's such that $X_i \in R_2$. Notice that c_1 and c_2 minimize the sums of squares $\sum_{X_i \in R_1}(Y_i - c_1)^2$ and $\sum_{X_i \in R_2}(Y_i - c_2)^2$. The choice of which covariate x_j to split on and which split point s to use is based on minimizing the residual sums if squares. The splitting process is on repeated on each rectangle R_1 and R_2.

Figure 3 shows a simple example of a regression tree; also shown are the corresponding rectangles. The function estimate $\widehat{m}$ is constant over the rectangles.

Generally one first grows a very large tree, then the tree is pruned to form a subtree by collapsing regions together. The size of the tree is a tuning parameter and is usually chosen by cross-validation.

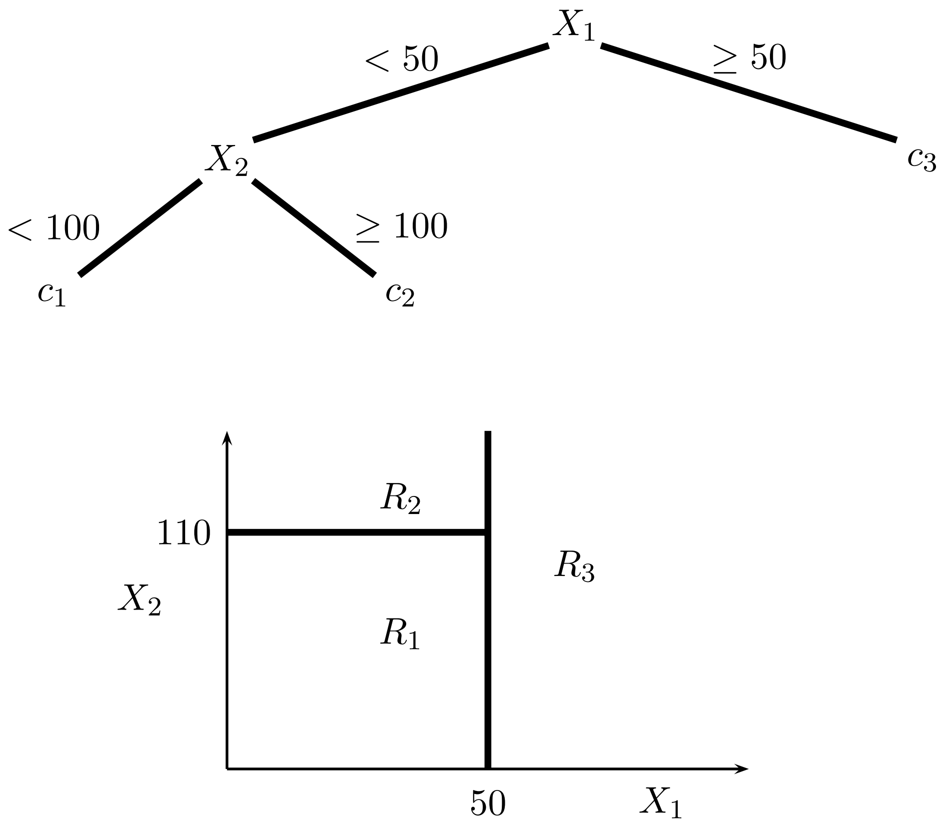

Figure 3: A regression tree for two covariates X_1 and X_2. The function estimate is $\widehat{m}(x) = c_1 I(x \in R_1) + c_2 I(x \in R_2) + c_3 I(x \in R_3)$ where R_1, R_2 and R_3 are the rectangles shown in the lower plot.

Example 4 *Figure 4 shows a tree for the rock data. Notice that the variable shape does not appear in the tree. This means that the shape variable was never the optimal covariate to split on in the algorithm. The result is that tree only depends on area and peri. This illustrates an important feature of tree regression: it automatically performs variable selection in the sense that a covariate x_j will not appear in the tree if the algorithm finds that the variable is not important.*

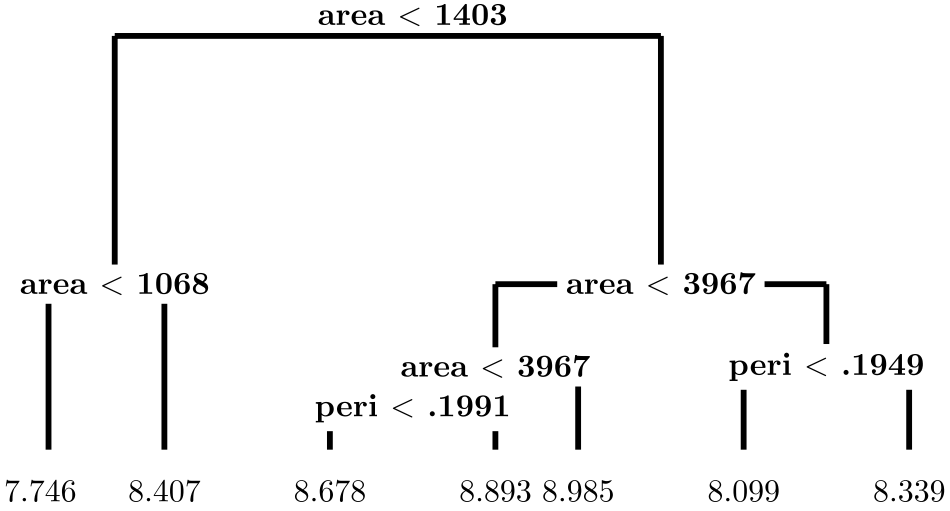

Figure 4: Regression tree for the rock data.

4 Piece-wise polynomials: Splines

A classical parametric regression model is simply polynomial regression, where as features we use monomials of degree upto k for some $k \in \mathbb{N}$, and then perform linear regression given these monomial features.

A non-parametric extension of this is to fit *piece-wise* polynomials, that is, we split the input domain into multiple regions, and fit polynomials in each. These are also known as *splines*.

Let's assume that $d = 1$ for simplicity. 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 $j = 0, 1, \ldots, k-1$

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 5

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* $B_1, \ldots, B_{p+k+1}$, defined as

$$\begin{gathered} B_1(x) = 1,\ B_2(x) = x,\ \ldots\ B_{k+1}(x) = x^k, \\ B_{k+1+j}(x) = (x - t_j)_+^k, \quad j = 1, \ldots, p. \end{gathered} \tag{15}$$

(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 $p + k + 1$.

Proof. (thanks to Vishwajeet Agarwal for the short proof) Any polynomial $p(x)$ of degree k that, at some knot t, has the same value and $k-1$ derivatives of another k degree polynomial $p_0(x)$ can be written as

$$p(x) = p_0(x) + c_k(x - t)^k.$$

To see this, consider the k-degree polynomial $g(x) = p(x+t) - p_0(x+t) = \sum_{j=0}^k c_j x^k$. Note that: $g^{(j)}(0) = c_j$. But under the matching constraint, $g^{(j)}(0) = 0$, for $j = 0, \ldots, k-1$. We thus have that $g(x) = c_k x^k$, so that by a reparameterization, $p(x) = p_0(x) + c_k(x - t)^k$. Consider for now a single knot t_0. Then, with the k-th degree polynomial $f_0(x) = \sum_{j=0}^k a_j x^k$ in the partition $(-\infty, t_0]$, the polynomial $f_1(x)$ in the next partition has to satisfy the matching constraints at the knot t_1, and hence can be written as $f_1(x) = f_0(x) + (x - t_1)^k$. Since we want this piece to be active only in $[t_0, \infty)$, we can compactly specify the piecewise polynomial over both the bins as $f_0(x) + (x - t_1)_+^k$. The argument naturally extends to more knots. □

While these basis functions are natural, a much better computational choice, both for speed and numerical accuracy, is the *B-spline* basis. This was a major development in spline theory and is now pretty much the standard in software. See de Boor (1978) or the Appendix of Chapter 5 in Hastie et al. (2009) for details.

We can then perform regression given these basis functions, the resulting approach is also called **regression splines**. This would work well provided we choose good knots $t_1, \ldots, t_p$; but in general choosing knots is a tricky business. Another problem with regression splines is that the estimates tend to display erratic behavior, i.e., they have high variance at the boundaries of the input domain. (This is the opposite problem to that 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

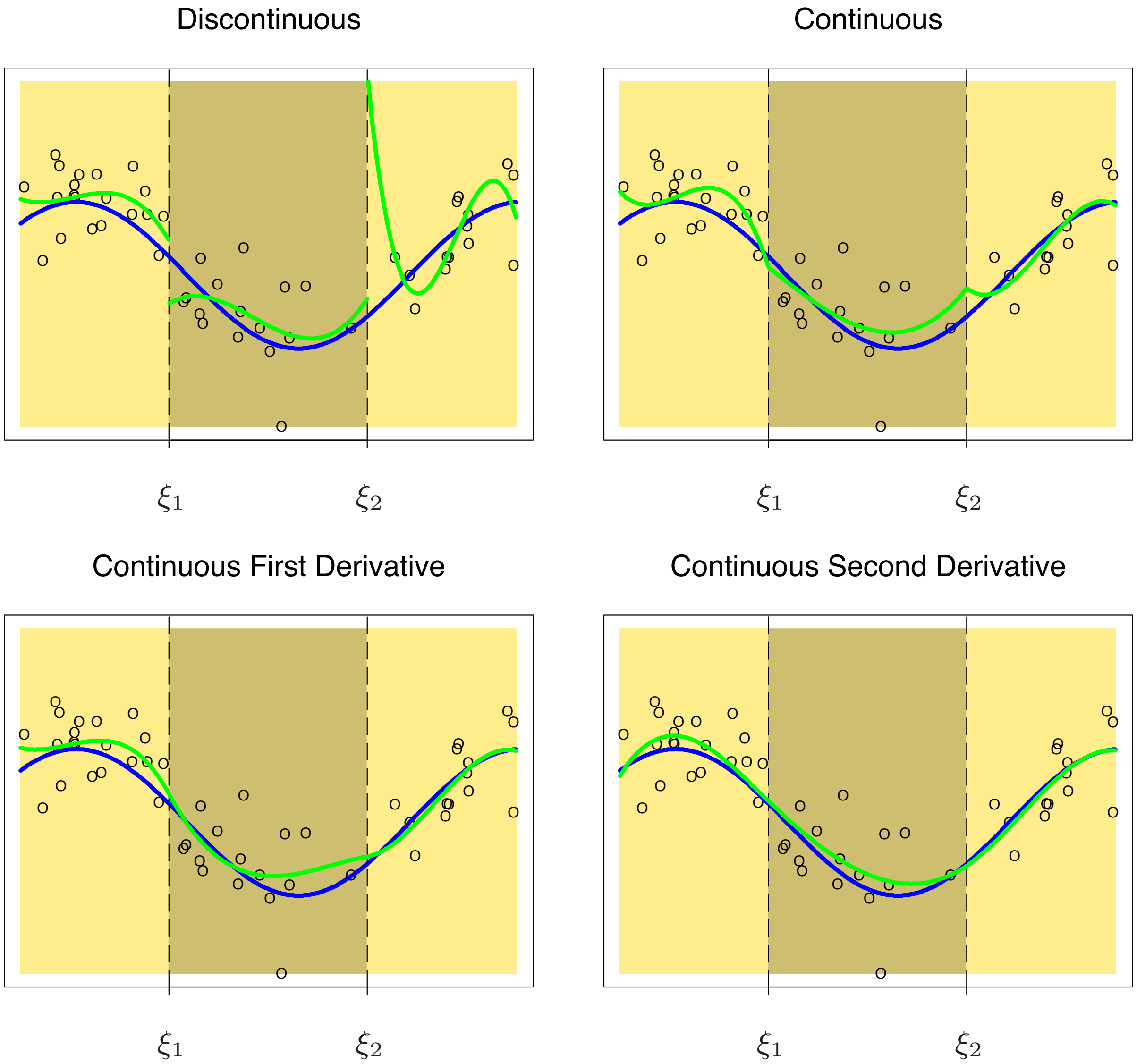

Figure 5: *Illustration of the effects of enforcing continuity at the knots, across various orders of the derivative, for a cubic piecewise polynomial. From Chapter 5 of Hastie et al. (2009)*

- 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. 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

Smoothing splines are simply regularized regression splines, 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)

4.1 Splines as Penalized Regression

Splines can also be motivated via the general approach of *penalized regression* (or *regularized regression*) where $\widehat{m}$ is defined to be the minimizer of

$$\sum_{i=1}^{n}(Y_i - \widehat{m}(X_i))^2 + \lambda J(\widehat{m}) \tag{16}$$

where $\lambda \geq 0$ and $J(\widehat{m})$ is a penalty (or regularization) term.

Suppose we choose the penalty J as

$$J(g) = \int (g''(x))^2 dx.$$

The minimizer of (16) can then be shown to be cubic splines with knots at $\{x_1, \ldots, x_n\}$.

Theorem 5 *Let $\widehat{m}$ be the minimizer of (16) where $J(g) = \int (g''(x))^2 dx$. Then $\widehat{m}$ is a cubic spline with knots at the points $X_1, \ldots, X_n$.*

According to this result, the minimizer $\widehat{m}$ of (16) is contained in $\mathcal{M}_n$, the set of all cubic splines with knots at $\{X_1, \ldots, X_n\}$. However, we still have to find which function in $\mathcal{M}_n$ is the minimizer.

Let $B_1, \ldots, B_{n+4}$ be the truncated power basis for $\mathcal{M}_n$ as defined earlier, given knots at $\{X_1, \ldots, X_n\}$: $B_1(x) = 1$, $B_2(x) = x$, $B_3(x) = x^2$, $B_4(x) = x^3$ and

$$B_j(x) = (x - X_{j-4})^3_+ \quad j = 5, \ldots, n+4.$$

(As noted in earlier section, in practice, another basis for $\mathcal{M}$ called the B-spline basis is used since it has better numerical properties.) Thus, every $g \in \mathcal{M}_n$ can be written as $g(x) = \sum_{j=1}^N \beta_j B_j(x)$ for some coefficients $\beta_1, \ldots, \beta_N$. If we substitute $\widehat{m}(x) = \sum_{j=1}^N \beta_j B_j(x)$ into (16), the minimization problem becomes: find $\beta = (\beta_1, \ldots, \beta_N)$ to minimize

$$(Y - \mathbb{B}\beta)^T (Y - \mathbb{B}\beta) + \lambda \beta^T \Omega \beta \tag{17}$$

where $Y = (Y_1, \ldots, Y_n)$, $\mathbb{B}_{ij} = B_j(X_i)$ and $\Omega_{jk} = \int B_j''(x) B_k''(x) dx$. The solution is

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

and hence

$$\widehat{m}(x) = \sum_j \widehat{\beta}_j B_j(x) = \ell(x)^T Y$$

where $\ell(x) = b(x)(\mathbb{B}^T \mathbb{B} + \lambda \Omega)^{-1} \mathbb{B}^T$ and $b(x) = (B_1(x), \ldots, B_N(x))^T$. Hence, the spline smoother is another example of a linear smoother.

The parameter λ is a smoothing parameter. As $\lambda \to 0$, $\widehat{m}$ tends to the interpolating function $\widehat{m}(X_i) = Y_i$. As $\lambda \to \infty$, $\widehat{m}$ tends to the least squares linear fit.

4.2 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 (**??**) 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 minimax optimal over $W_1(m, C)$ (e.g., Nussbaum (1985)).

4.3 Multivariate splines

Splines can be extended to multiple dimensions, in two different ways: *thin-plate splines* and *tensor-product splines*. See Chapter 7 of Green & Silverman (1994), and Chapters 15 and 20.4 of Gyorfi et al. (2002)). These multivariate extensions however are highly nontrivial, 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 is easier to study RKHS based estimators, which in fact covers smoothing splines (and thin-plate splines) as a special case.

5 Basis Functions and Dictionaries

Suppose that

$$m \in L_2(a,b) = \left\{ g : [a,b] \to \mathbb{R} : \ \int_a^b g^2(x)dx < \infty \right\}.$$

Let $\phi_1, \phi_2, \ldots$ be an orthonormal basis for $L_2(a,b)$. This means that $\int \phi_j^2(x)dx = 1$, $\int \phi_j \phi_k(x)dx = 0$ for $j \neq k$ and the only function $b(x)$ such that $\int b(x)\phi_j(x)dx = 0$ for all j is $b(x) = 0$. It follows that any $m \in L_2(a,b)$ can be written as

$$m(x) = \sum_{j=1}^{\infty} \beta_j \phi_j(x)$$

where $\beta_j = \int m(x)\phi_j(x)dx$. For $[a,b] = [0,1]$, an example is the cosine basis

$$\phi_0(x) = 1, \quad \phi_j(x) = \sqrt{2}\cos(\pi j x),\ j = 1, 2, \ldots$$

To use a basis for nonparametric regression, we regress Y on the first J basis functions and we treat J as a smoothing parameter. In other words we take $\widehat{m}(x) = \sum_{j=1}^{J} \widehat{\beta}_j \phi_j(x)$ where $\widehat{\beta} = (B^T B)^{-1} B^T Y$ and $B_{ij} = \phi_j(X_i)$. It follows that $\widehat{m}(x)$ is a linear smoother. See Chapters 7 and 8 of Wasserman (2006) for theoretical properties of orthogonal function smoothers.

It is not necessary to use orthogonal functions for smoothing. Let $\mathcal{D} = \{\psi_1, \ldots, \psi_N\}$ be any collection of functions, called a *dictionary*. The collection $\mathcal{D}$ could be very large. For example, $\mathcal{D}$ might be the union of several different bases. The smoothing problem is to decide which functions in $\mathcal{D}$ to use for approximating m. One way to approach this problem is to use the Lasso: regress Y on $\mathcal{D}$ using an ℓ_1 penalty.

We might discuss more about such "linear additive" models in a subsequent lecture.

6 k-nearest-neighbors regression

Before we study smoothing kernel based approaches, it is instructive to study their basic precursor which is *k-nearest-neighbors* regression. We fix an integer $k \geq 1$ and define

$$\widehat{m}(x) = \frac{1}{k} \sum_{i \in \mathcal{N}_k(x)} Y_i, \tag{18}$$

where $\mathcal{N}_k(x)$ contains the indices of the k closest points of $X_1, \ldots, X_n$ to x.

A small k corresponding to a more flexible fit, and large k less flexible.

However the fitted function $\widehat{m}$ essentially always looks jagged, especially for small or moderate k. Why is this? It helps to write

$$\widehat{m}(x) = \sum_{i=1}^{n} w_i(x) Y_i, \tag{19}$$

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 $\widehat{m}(x)$.

6.1 Consistency

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

$$\mathbb{E}\|\widehat{m} - m_0\|_2^2 \lesssim n^{-2/(2+d)}. \tag{20}$$

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,

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

In the last line we used the Lipschitz property $|m_0(x)-m_0(z)| \le L\|x-z\|_2$, for some constant $L > 0$. Now for "most" of the points we'll have $\|X_i - x\|_2 \le C(k/n)^{1/d}$, for a constant $C > 0$. (Think of a having input points X_i, $i = 1, \dots, n$ spaced equally over (say) $[0, 1]^d$.) Then our bias-variance upper bound becomes

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

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.

6.2 Curse of dimensionality

As discussed in the non-parametric density estimation lecture, the above error rate $n^{-2/(2+d)}$ exhibits a very poor dependence on the dimension d, requiring number of samples n scaling exponentially in the dimension d to achieve error ϵ: $n \ge \epsilon^{-(2+d)/2}$. See Figure 6 for an illustration with $\epsilon = 0.1$

This *curse of dimensionality* is unfortunately necessary: we cannot hope to do better than the rate in(20) over the space of L-Lipschitz functions in d dimensions, which we denote $H_d(1, L)$, for a constant $L > 0$. It can be shown that

$$\inf_{\widehat{m}} \sup_{m_0 \in H_d(1,L)} \mathbb{E}\|\widehat{m} - m_0\|_2^2 \gtrsim n^{-2/(2+d)}, \tag{21}$$

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.

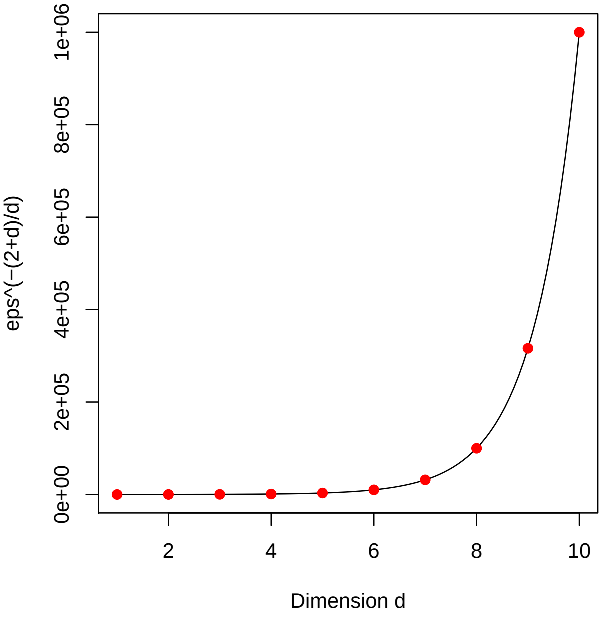

Figure 6: *The curse of dimensionality, with* $\epsilon = 0.1$

7 The Kernel Estimator

Another simple nonparametric estimator is the kernel estimator. The word “kernel” is often used in two different ways. Here are we referring to smoothing kernels. Later we will discuss *Mercer kernels* which are a distinct (but related) concept.

A one-dimensional *smoothing kernel* is any smooth function K such that

$$\int K(x)\,dx = 1, \quad \int xK(x)dx = 0 \quad \text{and} \quad \sigma_K^2 \equiv \int x^2 K(x)dx > 0. \tag{22}$$

Let $h > 0$ be a positive number, called the *bandwidth*. The *Nadaraya–Watson kernel estimator* is defined by

$$\widehat{m}(x) \equiv \widehat{m}_h(x) = \frac{\sum_{i=1}^n Y_i\ K\left(\frac{\|x-X_i\|}{h}\right)}{\sum_{i=1}^n K\left(\frac{\|x-X_i\|}{h}\right)} = \sum_{i=1}^n Y_i \ell_i(x) \tag{23}$$

where $\ell_i(x) = K(\|x - X_i\|/h)/\sum_j K(\|x - X_j\|/h)$.

Thus $\widehat{m}(x)$ is a local average of the Y_i's. It can be shown that the optimal kernel is the Epanechnikov kernel. But, as with density estimation, the choice of kernel K is not too important. Estimates obtained by using different kernels are usually numerically very similar. This observation is confirmed by theoretical calculations which show that the risk is very

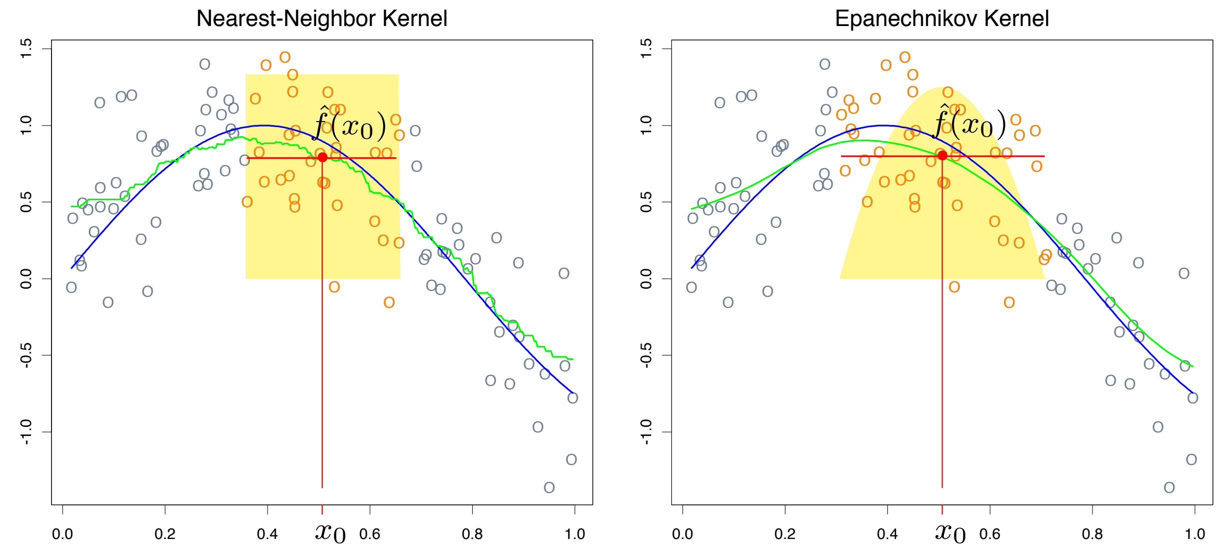

Figure 7: *Comparing k-nearest-neighbor and Epanechnikov kernels, when $d = 1$. From Chapter 6 of Hastie et al. (2009)*

insensitive to the choice of kernel. What does matter much more is the choice of bandwidth h which controls the amount of smoothing. Small bandwidths give very rough estimates while larger bandwidths give smoother estimates.

The kernel estimator can be derived by minimizing the localized squared error

$$\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) \left(c - Y_i\right)^2. \tag{24}$$

A simple calculation shows that this is minimized by the kernel estimator $c = \widehat{m}(x)$ as given in equation (23).

Kernel regression and kernel density estimation are related. Let $\widehat{p}(x, y)$ be the kernel density estimator and define

$$\widehat{m}(x) = \widehat{E}(Y|X = x) = \int y\widehat{p}(y|x)dy = \frac{\int y\widehat{p}(x, y)dy}{\widehat{p}(x)} \tag{25}$$

where $\widehat{p}(x) = \int \widehat{p}(x, y)dy$. Then $\widehat{m}(x)$ is the Nadaraya-Watson kernel regression estimator.

In comparison to the k-nearest-neighbors estimator in (18), which can be thought of as a raw (discontinuous) moving average of nearby responses, the kernel estimator in (23) is a smooth moving average of responses. See Figure 7 for an example with $d = 1$.

7.1 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.

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

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

where p is the density of P_X.

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 $n^{-4/(4+d)}$.

Biases of the bias. The first term in the risk bound from the theorem is the squared bias, and it has two disturbing properties. The first is that it has a dependence on p and p', which is also called **design bias**. We'll fix this problem later using local linear smoothing.

If the support has boundaries then there is bias of order $O(h)$ near the boundary, in contrast to $O(h^2)$ in the interior. This is also called **boundary bias**. The risk then becomes $O(h^3)$ instead of $O(h^4)$. This happens because of the asymmetry of the kernel weights in such regions. See Figure 8. We'll also 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 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

$$\sup_{P \in H_d(1,L)} \mathbb{E}\|\widehat{m} - m\|_P^2 \leq c_1 h^2 + \frac{c_2}{nh^d}. \tag{26}$$

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

$$\sup_{P \in H_d(1,L)} \mathbb{E}\|\widehat{m} - m\|_P^2 \leq \frac{c}{n^{2/(d+2)}}. \tag{27}$$

Recall from (21) 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

$$\inf_{\widehat{m}} \sup_{m_0 \in H_d(\alpha,L)} \mathbb{E}\|\widehat{m} - m_0\|_2^2 \gtrsim n^{-2\alpha/(2\alpha+d)}, \tag{28}$$

see again Chapter 3.2 of Gyorfi et al. (2002). But on the other hand this rate $n^{-2/(d+2)}$ is slower than the pointwise rate $n^{-4/(d+4)}$ earlier (which is minimax for $H_d(2,L)$) because we have made weaker assumptions.

8 Local Polynomials Estimators

Recall that the kernel estimator can be derived by minimizing the localized squared error

$$\sum_{i=1}^n K\left(\frac{x - X_i}{h}\right) \left(c - Y_i\right)^2. \tag{29}$$

To reduce the design bias and the boundary bias we simply replace the constant c with a polynomial. In fact, it is enough to use a polynomial of order 1; in other words, we fit a local linear estimator instead of a local constant. The idea is that, for u near x, we can write, $m(u) \approx \beta_0(x) + \beta_1(x)(u - x)$. We define $\widehat{\beta}(x) = (\widehat{\beta}_0(x), \widehat{\beta}_1(x))$ to minimize

$$\sum_{i=1}^n K\left(\frac{x - X_i}{h}\right) \left(Y_i - \beta_0(x) - \beta_1(x)(X_i - x)\right)^2.$$

Then $\widehat{m}(u) \approx \widehat{\beta}_0(x) + \widehat{\beta}_1(x)(u - x)$. In particular, $\widehat{m}(x) = \widehat{\beta}_0(x)$. The minimizer is easily seen to be

$$\widehat{\beta}(x) = (\widehat{\beta}_0(x), \widehat{\beta}_1(x))^T = (\mathbb{B}^T W \mathbb{B})^{-1} \mathbb{B}^T W \mathbb{Y}$$

where $\mathbb{Y} = (Y_1, \ldots, Y_n)$,

$$\mathbb{B} = \begin{pmatrix} 1 & X_1 - x \\ 1 & X_2 - x \\ \vdots & \vdots \\ 1 & X_n - x \end{pmatrix}, \quad W = \begin{pmatrix} K_h(x - X_1) & 0 & \cdots & 0 \\ 0 & K_h(x - X_2) & \cdots & 0 \\ \vdots & \vdots & \cdots & \cdots \\ 0 & 0 & \cdots & K_h(x - X_n) \end{pmatrix}.$$

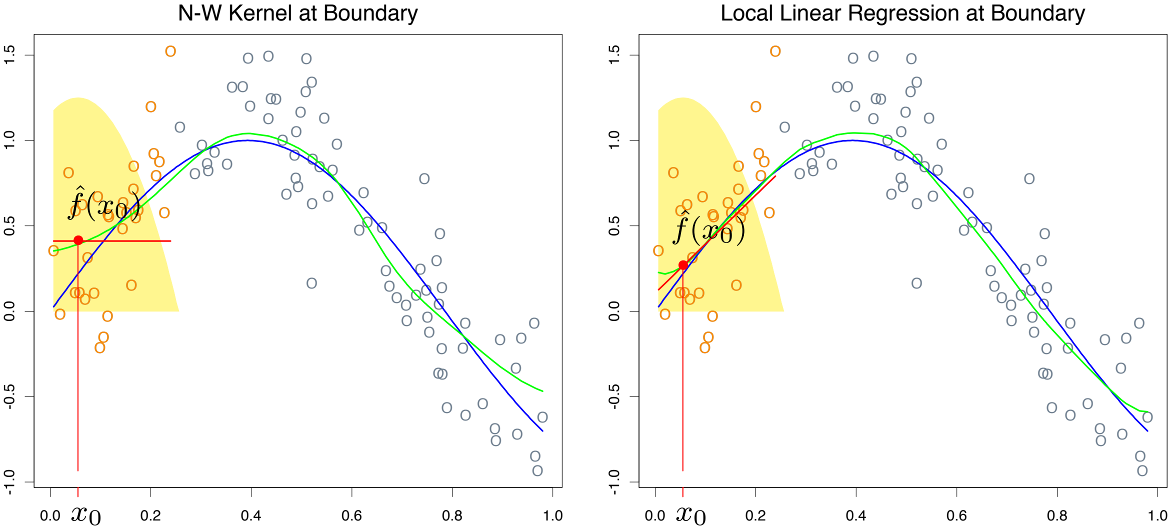

Figure 8: *Comparing (Nadaraya-Watson) kernel smoothing to local linear regression; the former is biased at the boundary, the latter is unbiased (to first-order). From Chapter 6 of Hastie et al. (2009)*

Then $\widehat{m}(x) = \widehat{\beta}_0(x)$.

It can be shown that local linear regression removes boundary bias and design bias. See Figure 8.

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

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

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.

8.1 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

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

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 W 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

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

See Chapter 1.6.1 of Tsybakov (2009). This matches the lower bound in (28) (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 to kernel smoothing but use a *higher-order kernel*. Recall that 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.

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)

9 Reproducing Kernel Hilbert Space (RKHS) Regression

A classical approach to nonparametric regression is to consider functions lying in a so-called Reproducing Kernel Hilbert Space (RKHS), and penalize the estimation loss by the RKHS norm of the function (or equivalently, constrain the RKHS norm by some constant). We next cover the necessary background.

9.1 Hilbert Spaces

A Hilbert space is a complete inner product space. A reproducing kernel Hilbert space (RKHS) is simply 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 $n \to \infty$. Whenever for any sequence $\{f_n\}$ s.t. $f_n \in \mathcal{H}$, $f_n \to f$, it also holds that $f \in \mathcal{H}$, we say that $\mathcal{H}$ is a complete space.

9.2 Evaluation Functional

The evaluation functional δ_x assigns a real number to each function. It is defined by $L_x f = f(x)$. This evaluation functional is clearly a linear functional, but in general,is not continuous. This means we can have $f_n \to f$ but $L_x f_n$ does not converge to $L_x f$. 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 $L_0 f_n = \sqrt{n}$ which does not converge to $L_0 f = 0$. Intuitively, this is because Hilbert spaces can contain very unsmooth functions. We shall see that RKHS are Hilbert spaces where the evaluation functional is continous. Intuitively, this means that the functions in the space are well-behaved.

What has this got to do with kernels? Hang on; we're getting there.

10 Mercer Kernels

Let us first define what a Mercer kernel is: it is a function $K(x, y)$ 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

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

Suppose the evaluation functional L_x of the Hilbert space is continuous. Then we can make use of an important theorem — the Reisz representation theorem — that says that any continuous linear functional L_x of a Hilbert space has a representer $K_x \in \mathcal{H}$ so that:

$$L_x(f) = \langle f, K_x \rangle_{\mathcal{H}}.$$

Define the bivariate function:

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

By the symmetry of the dot product, this is a symmetric function. It can also be shown to be positive semi-definite since:

$$\begin{aligned} \int\int a(x)K(x, y)a(y)dxdy &= \int\int a(x)\langle K_x, K_y \rangle_{\mathcal{H}} a(y)dxdy \\ &= \langle \int a(x)K_x dx, \int a(y)K_y dy \rangle_{\mathcal{H}} \\ &= \| \int a(x)K_x dx \|^2_{\mathcal{H}} \geq 0. \end{aligned}$$

Thus the function specified using the Reisz representers is a Mercer kernel. We can also associate $K(x, \cdot)$ with $K_x(\cdot)$ and as a function in $\mathcal{H}$ since:

$$K(x, y) = \langle K_x, K_y \rangle_{\mathcal{H}} = K_x(y),$$

due to the definition of the Reisz representers. We can thus show that $\{K(x, \dot{)}\}$ satisfy what is known as the reproducing kernel property:

$$K(x, y) = \langle K(x, \cdot), K(y, \cdot) \rangle_{\mathcal{H}},$$

and further that:

$$f(x) = \langle K(x, \cdot), f \rangle_{\mathcal{H}}.$$

So far, we have seen that any Hilbert space with a continuous linear functional is associated with a Mercer kernel that satisfies the reproducing kernel property.

We can also go in the other direction: any Mercer kernel is associated with a Hilbert space with a continuous linear functional. Suppose we are given a Mercer kernel K. Let $K_x(\cdot)$

be the function ontained 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 $\mathbb{K}_{jk} = K(x_j, x_k)$.

It can be seen that the reproducing kernel property introduced earlier holds.

Let $f(x) = \sum_i K_{x_i}(x)$. Note that we do have:

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

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

This also implies that

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

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 $f = 0$ impies 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

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

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

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

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

10.1 Examples

Example 6 *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 7 *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

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

and

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

Example 8 *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

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

10.2 Spectral Representation, RKHS as Orthogonal Series

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,

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

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.

10.3 Kernel Trick

Define the **feature map** Φ by

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

We can then see that $K(x,y)$ is the corresponding ℓ_2 inner product of the two ℓ_2 sequences $\Phi(x)$ and $\Phi(y)$. The key advantage of an RKHS is that this inner product is made computationally feasible by just evaluating the kernel $K(x,y)$.

Thus, in any algorithm that uses its features x only via inner products $\langle x_i, x_j \rangle$, we can then replace the features $\{x_i\}$ by their (infinite dimensional) feature maps $\{\Phi(x_i)\}$, and just substitute the linear feature inner products with the feature map inner products $K(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$ and get a nonlinear version of the algorithm. This is called the "kernel trick" since $K(x_i, x_j)$ is easy to compute, allowing us to turn a linear procedure into a nonlinear procedure without adding much computation.

10.4 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 $\alpha_1, \ldots, \alpha_n$.

10.5 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.**

One could also combine RKHS estimation with losses other than squared error, which we discuss further when we consider nonparametric classification.

10.6 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.

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

10.7 Two Sample Test

Gretton, Borgwardt, Rasch, Scholkopf and Smola (GBRSS 2008) show how to use kernels for two sample testing. Suppose that

$$X_1, \ldots, X_m \sim P \qquad Y_1, \ldots, Y_n \sim Q.$$

We want to test the null hypothesis $H_0 : P = Q$.

Let $\mathcal{F} = \{f : \; ||f||_K \leq 1\}$. Define

$$M = \sup_{f \in \mathcal{F}} \Bigg| \mathbb{E}_P[f(X)] - \mathbb{E}_Q[f(X)] \Bigg|.$$

Under weak regularity conditions on K, it can be shown that $M = 0$ if and only if $P = Q$. Thus we can test H_0 by estimating M.

Define

$$\widehat{M} = \sup_{f \in \mathcal{F}} \left| \frac{1}{m} \sum_{i=1}^m f(X_i) - \frac{1}{n} \sum_{i=1}^m f(Y_i) \right|.$$

Some calcculations show that

$$\widehat{M}^2 = \frac{1}{m^2} \sum_{j,k} K(X_j, X_k) - \frac{2}{mn} \sum_{j,k} K(X_j, Y_k) + \frac{1}{n^2} \sum_{j,k} K(Y_j, Y_k).$$

We reject H_0 if $\widehat{M} > t$.

To determine t, using McDiarmmid's inequality and a Rademacher bound, GBRSS shows that

$$\mathbb{P}\left(|\widehat{M} - M| > 2 \left(\sqrt{\frac{C}{m}} + \sqrt{\frac{C}{n}} \right) + \epsilon \right) \leq \exp\left(-\frac{\epsilon^2 mn}{C(m+n)} \right).$$

There is a connection with smoothing kernels. Let

$$\widehat{f}_X(u) = \frac{1}{m}\sum_{i=1}^n \kappa(X_i - u)$$

and similarly for $\widehat{f}_Y$. Then

$$\int |\widehat{f}_X(u) - \widehat{f}_Y(u)|^2 du = \widehat{M}^2$$

where $\widehat{M}$ is based on the kernel $K(x,y) = \int \kappa(x-z)\kappa(y-z)dz$. So they are really the same!

In practice, one would use the Gaussian kernel $K_\sigma(x,y) = e^{-\frac{||x-y||^2}{\sigma^2}}$. Call the resulting statistic $\widehat{M}_\sigma$. For hypothesis testing, there is no need to choose a bandwidth σ. Just define

$$\widehat{M} = \sup_\sigma \widehat{M}_\sigma.$$

Since the distribution of $\widehat{M}$ under H_0 is very complex and involved unknown quantities, the critical value can be obtained using permutation methods.

11 Choosing the Smoothing Parameter

The estimators depend on the bandwidth h. Let $R(h)$ denote the risk of $\widehat{m}_h$ when bandwidth h is used. We will estimate $R(h)$ and then choose h to minimize this estimate. As we know, the *training error*

$$\widetilde{R}(h) = \frac{1}{n}\sum_{i=1}^n (Y_i - \widehat{m}_h(X_i))^2 \tag{30}$$

is biased downwards. We will estimate the risk using the cross-validation.

11.1 Leave-One-Out Cross-Validation

The *leave-one-out cross-validation score* is defined by

$$\widehat{R}(h) = \frac{1}{n}\sum_{i=1}^n (Y_i - \widehat{m}_{(-i)}(X_i))^2 \tag{31}$$

where $\widehat{m}_{(-i)}$ is the estimator obtained by omitting the $i^{\rm th}$ pair (X_i, Y_i), that is, $\widehat{m}_{(-i)}(x) = \sum_{j=1}^n Y_j \ell_{j,(-i)}(x)$ and

$$\ell_{j,(-i)}(x) = \begin{cases} 0 & \text{if } j = i \\ \frac{\ell_j(x)}{\sum_{k\neq i}\ell_k(x)} & \text{if } j \neq i. \end{cases} \tag{32}$$

Theorem 9 *Let $\widehat{m}$ be a linear smoother. Then the leave-one-out cross-validation score $\widehat{R}(h)$ can be written as*

$$\widehat{R}(h) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Y_i - \widehat{m}_h(X_i)}{1 - L_{ii}}\right)^2 \tag{33}$$

where $L_{ii} = \ell_i(X_i)$ is the i^{th} diagonal element of the smoothing matrix L.

The smoothing parameter h can then be chosen by minimizing $\widehat{R}(h)$. An alternative is *generalized cross-validation* in which each L_{ii} in equation (33) is replaced with its average $n^{-1}\sum_{i=1}^{n} L_{ii} = \nu/n$ where $\nu = \text{tr}(L)$ is the effective degrees of freedom. (Note that ν depends on h.) Thus, we minimize

$$\text{GCV}(h) = \frac{\widetilde{R}}{(1-\nu/n)^2}. \tag{34}$$

Usually, GCV and cross-validation are very similar.

Using the approximation $(1-x)^{-2} \approx 1 + 2x$ we see that

$$\text{GCV}(h) \approx \frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{m}(X_i))^2 + \frac{2\nu\widehat{\sigma}^2}{n} \equiv C_p \tag{35}$$

where $\widehat{\sigma}^2 = n^{-1}\sum_{i=1}^{n}(Y_i - \widehat{m}(X_i))^2$. Equation (35) is the nonparametric version of the C_p statistic.

Example 10 (Doppler function) *Let*

$$m(x) = \sqrt{x(1-x)}\sin\left(\frac{2.1\pi}{x + .05}\right), \quad 0 \le x \le 1 \tag{36}$$

which is called the Doppler function. *This function is difficult to estimate and provides a good test case for nonparametric regression methods. The function is spatially inhomogeneous which means that its smoothness (second derivative) varies over x. The function is plotted in the top left plot of Figure 9. The top right plot shows 1000 data points simulated from $Y_i = m(i/n) + \sigma\epsilon_i$ with $\sigma = .1$ and $\epsilon_i \sim N(0,1)$. The bottom left plot shows the cross-validation score versus the effective degrees of freedom using local linear regression. The minimum occurred at 166 degrees of freedom corresponding to a bandwidth of .005. The fitted function is shown in the bottom right plot. The fit has high effective degrees of freedom and hence the fitted function is very wiggly. This is because the estimate is trying to fit the rapid fluctuations of the function near $x = 0$. If we used more smoothing, the right-hand side of the fit would look better at the cost of missing the structure near $x = 0$. This is always a problem when estimating spatially inhomogeneous functions.*

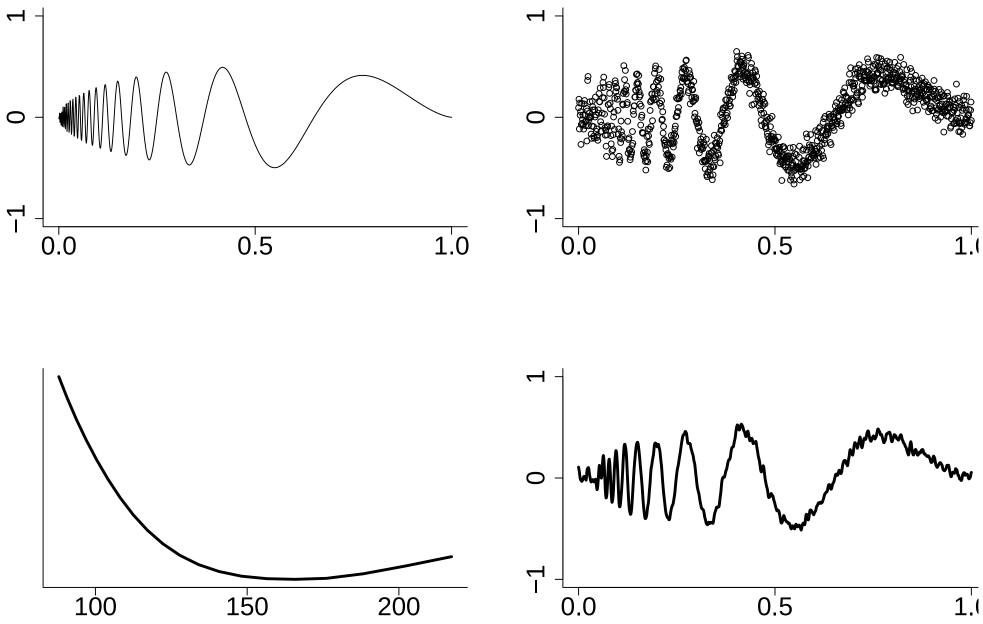

Figure 9: The Doppler function estimated by local linear regression. The function (top left), the data (top right), the cross-validation score versus effective degrees of freedom (bottom left), and the fitted function (bottom right).

11.2 Data Splitting

As with density estimation, stronger guarantees can be made using a *data splitting* version of cross-validation. Suppose the data are $(X_1, Y_1), \ldots, (X_{2n}, Y_{2n})$. Now randomly split the data into two halves that we denote by

$$\mathcal{D} = \Big\{ (\widetilde{X}_1, \widetilde{Y}_1), \ldots, (\widetilde{X}_n, \widetilde{Y}_n) \Big\}$$

and

$$\mathcal{E} = \Big\{ (X_1^*, Y_1^*), \ldots, (X_n^*, Y_n^*) \Big\}.$$

Construct regression estimators $\mathcal{M} = \{m_1, \ldots, m_N\}$ from $\mathcal{D}$. Define the risk estimator

$$\widehat{R}(m_j) = \frac{1}{n} \sum_{i=1}^{n} |Y_i^* - m_j(X_i^*)|^2.$$

Finally, let

$$\widehat{m} = \text{argmin}_{m \in \mathcal{M}} \widehat{R}(m).$$

Theorem 11 *There exists $C > 0$ such that*

$$\mathbb{E}(\|\widehat{m} - m\|_P^2) \leq 2 \inf_{m_* \in \mathcal{M}} \mathbb{E}\|m_* - m\|_P^2 + \frac{C \log N}{n}.$$

12 Linear Smoothers

Kernel estimators and local polynomial estimator are examples of *linear smoothers*.

Definition: An estimator $\widehat{m}$ of m is a *linear smoother* if, for each x, there is a vector $\ell(x) = (\ell_1(x), \ldots, \ell_n(x))^T$ such that

$$\widehat{m}(x) = \sum_{i=1}^{n} \ell_i(x) Y_i = \ell(x)^T Y \tag{37}$$

where $Y = (Y_1, \ldots, Y_n)^T$.

For kernel estimators, $\ell_i(x) = \frac{K(\|x-X_i\|/h)}{\sum_{j=1}^n K(\|x-X_j\|/h)}$. For local linear estimators, we can deduce the weights from the expression for $\widehat{\beta}(x)$. Here is an interesting fact: the following estimators are linear smoothers: Gaussian process regression, splines, RKHS estimators.

Example 12 *You should note confuse linear smoothers with linear regression. In linear regression we assume that $m(x) = x^T\beta$. In fact, least squares linear regression is a special case of linear smoothing. If $\widehat{\beta}$ denotes the least squares estimator then $\widehat{m}(x) = x^T\widehat{\beta} = x^T(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^TY = \ell(x)^TY$ where $\ell(x) = x^T(\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T$.*

Define the vector of *fitted values* $\widehat{Y} = (\widehat{m}(X_1), \ldots, \widehat{m}(X_n))^T$. It follows that $\widehat{Y} = LY$ where

$$L = \begin{pmatrix} \ell(X_1)^T \\ \ell(X_2)^T \\ \vdots \\ \ell(X_n)^T \end{pmatrix} = \begin{pmatrix} \ell_1(X_1) & \ell_2(X_1) & \cdots & \ell_n(X_1) \\ \ell_1(X_2) & \ell_2(X_2) & \cdots & \ell_n(X_2) \\ \vdots & \vdots & \vdots & \vdots \\ \ell_1(X_n) & \ell_2(X_n) & \cdots & \ell_n(X_n) \end{pmatrix}. \tag{38}$$

The matrix L defined in (38) is called the *smoothing matrix*. The i^{th} row of L is called the *effective kernel* for estimating $m(X_i)$. We define the *effective degrees of freedom* by

$$\nu = \text{tr}(L). \tag{39}$$

The effective degrees of freedom behave very much like the number of parameters in a linear regression model.

Remark. The weights in all the smoothers we will use have the property that, for all x, $\sum_{i=1}^n \ell_i(x) = 1$. This implies that the smoother preserves constants.

13 Wavelets

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 $[0, 1]$:

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

Thus the inputs here are fixed and not random, such a setting is called the fixed design regression setting. The assumption of evenly spaced inputs is crucial for fast computations; we also typically assume with wavelets that n is a power of 2. 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 can then write the wavelet smoothing estimate in a familiar form, following our previous discussions on basis functions and regularization.

We now form a wavelet basis matrix $W \in \mathbb{R}^{n \times n}$, defined by

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

There are two popular wavelet estimates. The first, hard-thresholding wavelet estimates, solve

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

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".

The second, soft-thresholding wavelet estimates, solve

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

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

For both of these, 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):

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

to get our wavelet parameter estimates. To get the prediction estimate, we 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.

13.1 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 $\mathrm{TV}(f^{(k)}) = \int_0^1 |f^{(k+1)}(x)|\,dx$.)

For the wavelet smoothing estimator, denoted by $\widehat{m}^{\mathrm{wav}}$, 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),

$$\mathbb{E}\|\widehat{m}^{\mathrm{wav}} - m_0\|_2^2 \lesssim n^{-(2k+2)/(2k+3)} \quad \text{and} \quad \inf_{\widehat{m}} \sup_{m_0 \in M(k,C)} \mathbb{E}\|\widehat{m} - m_0\|_2^2 \gtrsim n^{-(2k+2)/(2k+3)}. \tag{40}$$

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).)

Donoho & Johnstone (1998) showed that the minimax error over $M(k,C)$, *restricted to linear smoothers*, is much larger:

$$\inf_{\widehat{m} \text{ linear}} \sup_{m_0 \in M(k,C)} \mathbb{E}\|\widehat{m} - m_0\|_2^2 \gtrsim n^{-(2k+1)/(2k+2)}. \tag{41}$$

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 is both difficult theoretically and practically to get right.

References

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

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

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.

Härdle, W. K., Müller, M., Sperlich, S. & Werwatz, A. (2012), *Nonparametric and semiparametric models*, Springer Science & Business Media.

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

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

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

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.

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