Nonparametric Regression: Nearest Neighbors and Kernels

Advanced Topics in Statistical Learning, Spring 2024

Ryan Tibshirani

1 Introduction

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

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

is called the regression function (of Y on X). The basic goal in nonparametric regression is to construct an estimator $\hat{f}$ of f_0 without assuming a specific parametric form for f_0, and instead only assuming that f_0 is smooth in some way.

We typically estimate $\hat{f}$ from i.i.d. samples $(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$, $i = 1, \ldots, n$ that have the same joint distribution as (X, Y). We often call X the *input*, *predictor*, *feature*, etc., and Y the *output*, *outcome*, *response*, etc. Remember that we called this the XY-Pairs model, which is equivalent to the Random-X signal plus noise model:

$$\begin{gathered}(x_i, y_i),\ i = 1, \ldots, n \text{ are i.i.d.,} \\ \text{where each } y_i = f_0(x_i) + \epsilon_i, \text{ and } \mathbb{E}[\epsilon_i] = 0.\end{gathered}$$

If we *additionally assume that each* $x_i \perp\!\!\!\perp \epsilon_i$, then recall we can condition on x_i, $i = 1, \ldots, n$, and we get the Fixed-X signal plus noise model:

$$\begin{gathered}x_i,\ i = 1, \ldots, n \text{ are fixed,} \\ \delta_i,\ i = 1, \ldots, n \text{ are i.i.d.,} \\ \text{where each } y_i = f_0(x_i) + \delta_i, \text{ and } \mathbb{E}[\delta_i] = 0.\end{gathered}$$

For the theory we will present in what follows, we will assume that each $x_i \perp\!\!\!\perp \epsilon_i$ (recall from the review lecture that this is not a completely innocuous assumption). Hence we will take the liberty of viewing x_i, $i = 1, \ldots, n$ as random or fixed—simply adopting whichever perspective is convenient at any given point. For example, we may treat them as fixed in some key parts of an analysis, and then integrate over them at the end.

This is done for simplicity and we should note that most of the classic theory in nonparametric regression can be done without the assumption that $x_i \perp\!\!\!\perp \epsilon_i$; e.g., see Gyorfi et al. (2002) (however, in exchange for removing this assumption, they typically make an assumption about boundedness of the features).

1.1 Notation

For nonrandom sequences a_n, b_n, we will write $a_n \lesssim b_n$ to mean $a_n = O(b_n)$, and for a random sequence A_n, we say "$A_n \lesssim b_n$ in probability" to mean $A_n = O_p(b_n)$. This just simplifies our notation a bit when b_n is a power of n with a fractional exponent. We also use $a_n \asymp b_n$ to mean $a_n = O(b_n)$ and $b = O(a_n)$.

Given $x_1 \ldots, x_n$, recall that the $L^2(P_n)$ norm, where P_n is the empirical distribution of $x_1 \ldots, x_n$, is defined by

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

You'll often see this written as $\|\cdot\|_{L^2(P_n)}$, but in this document we'll abbreviate this by $\|\cdot\|_n$, and we'll call this the *empirical* L^2 *norm* (and often drop "L^2" when it is clear from the context).

For $x_0 \sim P$, recall that the $L^2(P)$ norm is defined by

$$\|f\|_2^2 = \mathbb{E}[f^2(x_0)] = \int f^2(x)\, dP(x).$$

Again, you'll often see this written as $\|\cdot\|_{L^2(P)}$, but in we'll abbreviate this by $\|\cdot\|_2$, and we'll call this the *population* L^2 *norm* (often drop "L^2" when it is clear from the context).

Key quantities of interests will be the in-sample and out-of-sample error incurred by an estimator $\hat{f}$ of f_0, which are, respectively,

$$\|\hat{f} - f_0\|_n^2 \quad \text{and} \quad \|\hat{f} - f_0\|_2^2.$$

In either case, this is a random quantity (since $\hat{f}$ is itself random). In general, in nonparametric regression, we study error[1] bounds in probability or in expectation, depending on what is more convenient in the given analysis. Upper bounds on the risk (as we will see in this lecture and the next) are usually easier to derive in probability, and lower bounds (as we will see in the minimax theory lecture) are typically easier to derive in expectation. We tend not to fuss about this discrepancy, particularly because upper bounds in probability can be translated into upper bounds in expectation whenever the probability control is sharp enough (as you saw on the homework).

1.2 What does "nonparametric" mean?

Importantly, in nonparametric regression we don't assume a particular parametric model for f_0. Still, in many approaches, we estimate f_0 using a linear combination of nonlinear basis functions, written as

$$\hat{f}(x) = \sum_{j=1}^{m} \hat{\beta}_j g_j(x).$$

A common question that comes to mind when learning this material: aren't the coefficients on the basis functions parameters? And so ... how is this nonparametric?

To be clear, the point is that *we don't assume a parametric form for* f_0, i.e., we don't assume f_0 is itself a linear combination of $g_1, \ldots, g_m$. In this sense, the estimated coefficients $\hat{\beta}_1, \ldots, \hat{\beta}_m$ are not really viewed as parameter estimates; and we are not concerned with how close they are to some "true" parameters. We only care about how close $\hat{f}$ is to f_0.

Of course, in order for this to "work", we need our representation to be sufficiently flexible—otherwise we can't guarantee $\hat{f}$ will be close f_0. The gap between our representation and f_0 is known as the *approximation error*. For traditional nonparametric theory (e.g., theory for k-nearest neighbors and kernel regression), this won't be an explicit part of the analysis; but for some advanced results, it will (we'll touch on this in a later lecture). Here is a concrete example of an approximation guarantee: for any $f_0 : [0,1] \to \mathbb{R}$ whose second derivative is integrable, and evenly-spaced points $t_1, \ldots, t_N \in [0,1]$, there is a cubic spline $\bar{f}$ with knots at $t_1, \ldots, t_N$ (to be defined precisely in the splines lecture) such that

$$\|\bar{f} - f_0\|_\infty \leq \frac{c}{N}\bigg[\int_0^1 [f_0''(x)]^2\, dx\bigg]^{1/2},$$

where $c > 0$ is a constant and $\|f\|_\infty = \sup_{x\in[0,1]} |f(x)|$ is the L^∞ norm on $[0,1]$. Note that $\int_0^1 [f_0''(x)]^2\, dx$ is a measure of the smoothness of f_0. If this remains constant, and we choose $N = \sqrt{n}$, then the squared L^2 approximation error—either in empirical or population norm ($\|\bar{f} - f_0\|_n^2$ or $\|\bar{f} - f_0\|_2^2$)—will be on the order of $1/n$, which will be negligible relative to the overall estimation error that we will encounter.

[1] In the review lecture, we were more precise about using "error" and "risk" to mean separate things; and defined the risk via the expectations $\mathbb{E}\|\hat{f} - f_0\|_n^2$ and $\mathbb{E}\|\hat{f} - f_0\|_2^2$, with respect to appropriate randomness in the training set that is used to define $\hat{f}$. In this and future lectures, we'll often take the liberty of being more flexible with our nomenclature.

1.3 What we cover here

The goal is to expose you to a variety of methods over this and our next lecture on nonparametric regression, and give you a flavor of some interesting results, under different assumptions. A few topics we will cover into more depth than others. Of the many texts you can consult for more details, proofs, etc., we highlight Gyorfi et al. (2002); Wasserman (2006); Tsybakov (2009) as general references on theory; Hastie et al. (2009) as a general reference on methods.

With nearest neighbor and kernel methods, we will be able to starting working directly in the multivariate case ($d > 1$) without much setup or forewarning; but with splines, we will need to spend a good deal of time in the univariate case ($d = 1$) first, and even then moving to the multivariate setting will be far from trivial. In general, some methods in nonparametric regression have obvious (natural) multivariate extensions, and others don't. Nonetheless, we can always use low-dimensional (even just univariate) nonparametric regression methods as building blocks for high-dimensional nonparametric regression—we may study this later in the course, if and when we end up talking about additive models.

Lastly, a lot of what we cover for nonparametric regression also carries over to nonparametric classification, which we may also cover later in the course.

2 Nearest neighbors methods

Here's a basic nonparametric method to start us off, arguably the most basic of them all: *k-nearest neighbors* (kNN) regression. We fix an integer $k \geq 1$ and define

$$\hat{f}(x) = \frac{1}{k} \sum_{i \in \mathcal{N}_k(x)} y_i, \tag{1}$$

where $\mathcal{N}_k(x)$ contains the indices of the k closest (in ℓ_2 distance) 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 $\hat{f}$, with small k corresponding to a more flexible fit, and large k less flexible. For $k = n$, the estimator $\hat{f}$ is a constant function, simply predicting the grand mean $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$ at all x.

But it does have its limitations, one apparent one being that the fitted function $\hat{f}$ essentially always looks jagged, especially for small or moderate k. Why is this? It helps to write

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

where the weights functions are defined as

$$w_i(x) = \begin{cases} 1/k & \text{if } i \in \mathcal{N}_k(x) \\ 0 & \text{otherwise} \end{cases}, \quad i = 1, \ldots, n.$$

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

2.1 Linear smoothers

The representation (2) also reveals that the k-nearest neighbors estimator is in a class of estimators we call *linear smoothers*: these are methods for which (2) holds for some weight functions $w_1, \ldots, w_n$. To be clear, this means that for fixed x and inputs $x_1, \ldots, x_n$, the prediction $\hat{f}(x)$ is a linear function of $y_1, \ldots, y_n$; and it does not mean $\hat{f}$ need behave linearly as a function of x!

We note that in a linear smoother, each weight $w_i(x)$ can actually also depend on the inputs $x_1, \ldots, x_n$ in addition to x (as it does in kNN regression), but critically, it cannot depend on $y_1, \ldots, y_n$. Moreover,

writing $Y = (y_1, \ldots, y_n) \in \mathbb{R}^n$ for the response vector, and $\hat{Y} = (\hat{f}(x_1), \ldots, \hat{f}(x_n)) \in \mathbb{R}^n$ for the vector of fitted values, for a linear smoother we have

$$\hat{Y} = SY,$$

for a matrix $S \in \mathbb{R}^{n \times n}$. Again, the matrix S can depend on the inputs $x_1, \ldots, x_n$, but not on $y_1, \ldots, w_n$. The class of linear smoothers is quite large, in the sense that it contains many popular estimators, as we'll see in the coming sections.

2.2 Universal consistency

The k-nearest neighbors estimator is *universally consistent*, which means that $\mathbb{E}\|\hat{f} - f_0\|_2^2 \to 0$ as $n \to \infty$, 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). What makes this "universal" is that it places essentially no assumptions on the problem (in particular no assumptions on f_0). See Chapter 6.2 of Gyorfi et al. (2002).

2.3 Rate of convergence

Furthermore, assuming the underlying regression function f_0 is Lipschitz continuous, which means that for a constant $L > 0$,

$$|f_0(x) - f_0(z)| \leq L\|x - z\|_2, \quad \text{for all } x, z,$$

and the input point distribution is supported on $[0,1]^d$ and meets mild conditions, the k-nearest neighbors estimator with $k \asymp n^{2/(2+d)}$ satisfies

$$\mathbb{E}\big[\|\hat{f} - f_0\|_2^2 \,\big|\, x_{1:n}\big] \lesssim n^{-2/(2+d)} \quad \text{in probability,} \tag{3}$$

with respect to the randomness over draws of the input points $x_{1:n} = \{x_1, \ldots, x_n\}$.

Proof sketch: denote $\sigma^2 = \mathrm{Var}[\epsilon_0]$, and condition on the input points $x_{1:n}$, which for simplicity we omit notationally in the expectation statements that follow. Conditioning on x_0, and using the bias-variance decomposition,

$$\begin{aligned}
\mathbb{E}\big[(\hat{f}(x_0) - f_0(x_0))^2 \,\big|\, x_0\big] &= \underbrace{\big(\mathbb{E}[\hat{f}(x_0)|x_0] - f_0(x_0)\big)^2}_{\mathrm{Bias}^2(\hat{f}(x_0)|x_0)} + \underbrace{\mathbb{E}\big[\big(\hat{f}(x_0) - \mathbb{E}[\hat{f}(x_0)|x_0]\big)^2 \,\big|\, x_0\big]}_{\mathrm{Var}(\hat{f}(x_0)|x_0)} \\
&= \bigg[\frac{1}{k} \sum_{i \in \mathcal{N}_k(x_0)} \big(f_0(x_i) - f_0(x_0)\big)\bigg]^2 + \frac{\sigma^2}{k} \\
&\leq \bigg[\frac{L}{k} \sum_{i \in \mathcal{N}_k(x_0)} \|x_i - x_0\|_2\bigg]^2 + \frac{\sigma^2}{k}.
\end{aligned}$$

In the last line we used the Lipschitz property. Now it turns out that $\mathbb{E}[\max_{i \in \mathcal{N}_k(x_0)} \|x_i - x_0\|_2] \lesssim (k/n)^{1/d}$, in probability. (To get intuition for this, think of the case when the inputs $x_1, \ldots, x_n$ form a lattice on $[0,1]^d$.) You will prove this probability bound on the homework. Then taking an expectation over x_0, our bias-variance upper bound (in probability) becomes

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

Balancing the two terms so that they are equal gives $k^{1+2/d} \asymp n^{2/d}$, i.e., $k \asymp n^{2/(2+d)}$. And plugging this in gives the error rate of $n^{-2/(2+d)}$, as claimed.

2.4 Curse of dimensionality

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

$$n \geq \epsilon^{-(2+d)/2}.$$

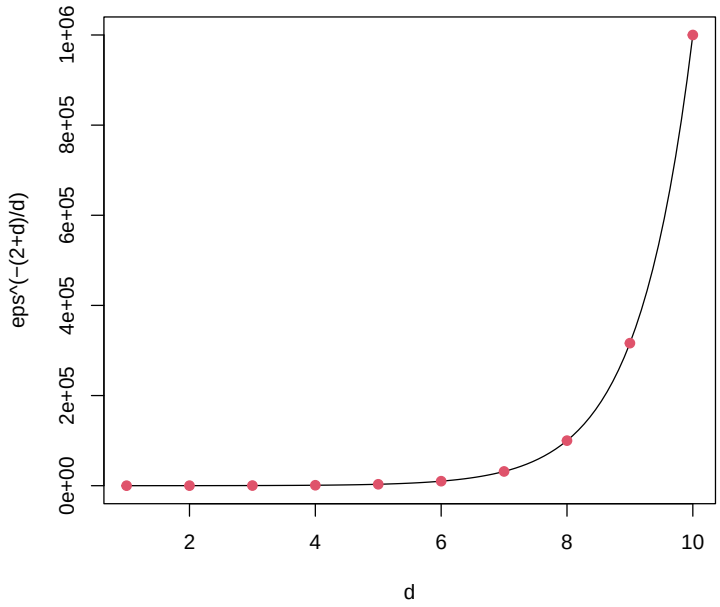

Figure 1: *The curse of dimensionality, with* $\epsilon = 0.1$.

That is, as we increase d, we require *exponentially more samples* n in order to achieve an error bound of ϵ. See Figure 1 for an illustration.

In fact, this phenomenon is not specific to k-nearest neighbors—it is 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 (3) over $C^1(L; [0,1]^d)$, which we write for the space of L-Lipschitz functions on $[0,1]^d$. It can be shown that

$$\inf_{\hat{f}} \sup_{f_0 \in C^1(L;[0,1]^d)} \mathbb{E}\|\hat{f} - f_0\|_2^2 \gtrsim n^{-2/(2+d)}, \tag{4}$$

where the infimum is over all estimators $\hat{f}$. This is true for a uniform input distribution, or for fixed inputs points on a lattice. We will revisit (and prove) this in the minimax theory lecture.

So to circumvent this curse, we'll need to make more assumptions about what it is that we're looking for in high dimensions. One such example is the additive model, covered near the end.

3 Kernel smoothing

One level up in sophistication is *kernel smoothing* or *kernel regression*. We begin with a kernel function $K : \mathbb{R} \to \mathbb{R}_+$, satisfying

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

Note carefully that—for now—we are assuming the kernel can only take nonnegative values, $K \geq 0$. Three common examples are the spherical (also called rectangular or boxcar) kernel:

$$K(t) = 1\{|t| \leq 1\},$$

the Gaussian kernel:

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

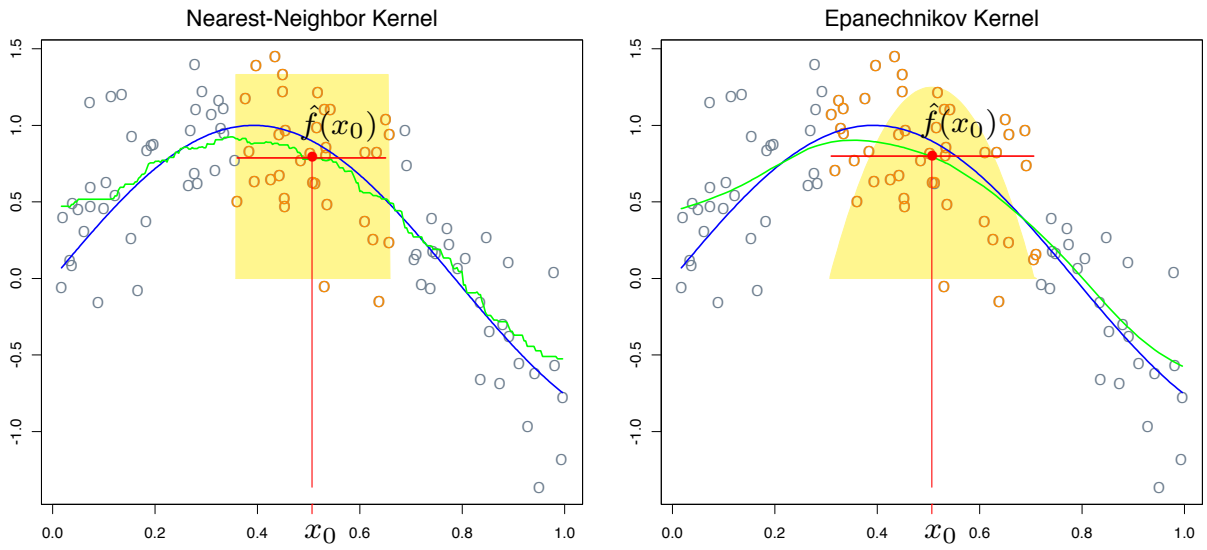

Figure 2: *Comparing kNN and Epanechnikov kernels. Credit: Chapter 6.1 of Hastie et al. (2009).*

and the Epanechnikov kernel:

$$K(t) = 3/4(1-t^2)1\{|x| \leq 1\}.$$

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

$$\hat{f}(x) = \frac{\sum_{i=1}^n K\left(\frac{\|x - x_i\|_2}{h}\right) y_i}{\sum_{i=1}^n K\left(\frac{\|x - x_i\|_2}{h}\right)}, \tag{5}$$

Note that kernel smoothing is also a linear smoother (2), with choice of weights

$$w_i(x) = \frac{K\left(\frac{\|x - x_i\|_2}{h}\right)}{\sum_{j=1}^n K\left(\frac{\|x - x_j\|_2}{h}\right)}, \quad i = 1, \ldots, n.$$

When K is continuous (Gaussian or Epanechnikov), the kernel smoothing estimator is a continuous moving average of the responses. Compared to the kNN regression estimator (1), which can be thought of as a raw (discontinuous) moving average of nearby responses, the kernel estimator in (5) is a smooth moving. See Figure 2 for an example.

Of course, with a spherical kernel, there is a strong similarity between kernel smoothing and kNN regression. The difference is that the former performs averages over fixed neighborhoods, whereas the latter uses adaptive neighborhoods—whose radius at x is defined by the distance to the k^{th}-nearest neighbor.

In fact, under suitable conditions, it can be shown that kNN regression acts like spherical kernel smoothing with a *density-dependent local bandwidth*; that is, if $w(x) = (w_1(x), \ldots, w_n(x)) \in \mathbb{R}^n$ denotes the kNN weight function, and we write $w_i(x) = w(x, x_i)$ to emphasize that this weight gets attributed to (x_i, y_i) in the weighted sum $w(x)^\mathsf{T} Y = \sum_{i=1}^n w(x, x_i) y_i$ which gives us the kNN prediction at x, then for large n,

$$w(x, z) \approx \frac{1}{k} \cdot 1\{\|x - z\|_2 \leq h(x)\}$$

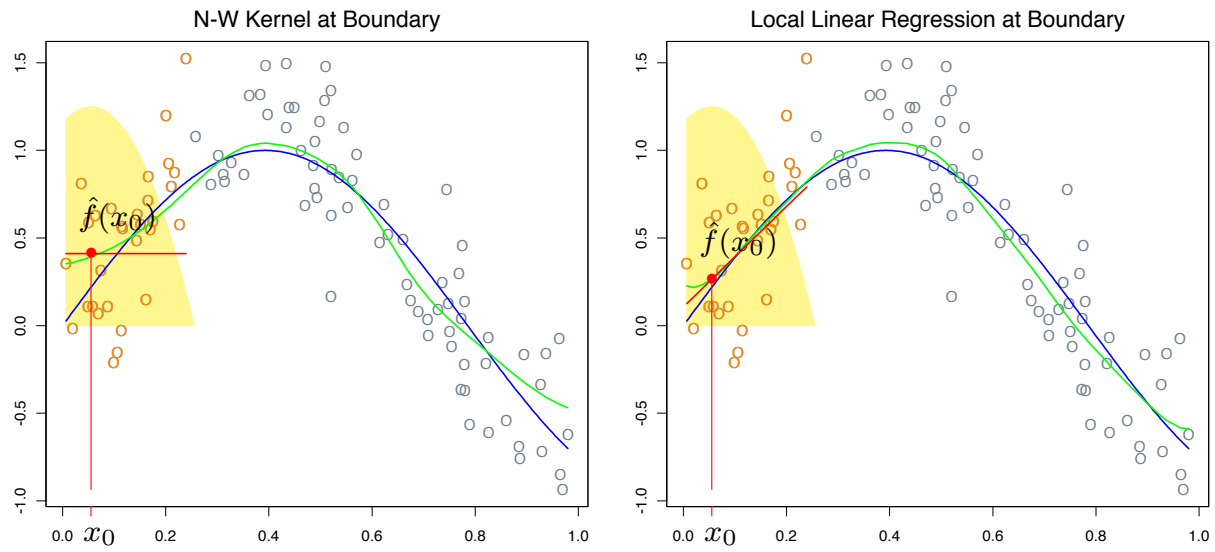

Figure 3: *Comparing kernel smoothing to local linear regression; the former is biased at the boundary, and the latter is unbiased (to first-order). Credit: Chapter 6.1 of Hastie et al. (2009).*

where

$$h(x) = \left(\frac{1}{\nu_d \, p(x)} \cdot \frac{k}{n}\right)^{1/d},$$

with $p(x)$ denoting density of the input distribution at x, and $\nu_d = \mathrm{vol}(B_d(0,1))$ the Lebesgue measure of the unit ball in $\mathbb{R}^d$. (Note that this sharpens the earlier claim about k^{th}-nearest neighbor distances having the asymptotic scaling $(k/n)^{1/d}$.) This is called the *equivalent kernel* for k-nearest neighbor regression.

3.1 Local linear regression

A shortcoming of kernel regression is that it suffers from poor bias at the boundary of the domain of the input points $x_1, \ldots, x_n$. This essentially happens because of the asymmetry of the kernel weights in such regions. See Figure 3 for an illustration.

We can alleviate this boundary bias issue by moving from a local constant fit to a local linear fit. To build intuition, another way to view the kernel smoothing estimator in (5) is as follows: at each input point x, it employs the estimate $\hat{f}(x) = \hat{\theta}_x$, which solves

$$\underset{\theta}{\text{minimize}} \sum_{i=1}^{n} K\left(\frac{\|x - x_i\|_2}{h}\right)(y_i - \theta)^2,$$

Instead we could consider forming the local estimate $\hat{f}(x) = \hat{\alpha}_x + \hat{\beta}_x^{\mathsf{T}} x$, where $\hat{\alpha}_x, \hat{\beta}_x$ solves

$$\underset{\alpha,\beta}{\text{minimize}} \sum_{i=1}^{n} K\left(\frac{\|x - x_i\|_2}{h}\right)(y_i - \alpha - \beta^{\mathsf{T}} x_i)^2. \tag{6}$$

This is called *local linear regression*.

We can rewrite the local linear regression prediction $\hat{f}(x)$ in a more evocative way. This is just given by a weighted least squares, so we can write

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

where $b(x) = (1, x) \in \mathbb{R}^{d+1}$, $B \in \mathbb{R}^{n \times (d+1)}$ is the matrix whose i^{th} row is $b(x_i)$, and $\Omega \in \mathbb{R}^{n \times n}$ is a diagonal matrix whose i^{th} element is $K(\|x - x_i\|_2/h)$.

We can write the local linear regression more concisely as a linear smoother (2): $\hat{f}(x) = w(x)^\mathsf{T} Y$, where

$$w(x) = \Omega B (B^\mathsf{T} \Omega B)^{-1} b(x).$$

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

$$\hat{Y} = B (B^\mathsf{T} \Omega B)^{-1} B^\mathsf{T} \Omega Y,$$

which should look familiar to you from weighted least squares.

3.2 Boundary bias calculation

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

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

At each x_i, using a Taylor expansion of f_0 about x,

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

where the remainder term R contains quadratic and higher-order terms, and under regularity conditions, is small. One can check (via direct algebra) that

$$\sum_{i=1}^n w_i(x) = 1 \quad \text{for both kernel smoothing and local linear regression.}$$

On the other hand

$$\sum_{i=1}^n (x_i - x) w_i(x) \neq 0 \quad \text{for kernel smoothing, and}$$
$$\sum_{i=1}^n (x_i - x) w_i(x) = 0 \quad \text{for local linear regression.}$$

Indeed the first-order bias term $\nabla f_0(x)^\mathsf{T} (\sum_{i=1}^n (x_i - x) w_i(x))$ will be generally large for kernel smoothing for x near the boundary. Meanwhile, for local linear regression, $\mathbb{E}[\hat{f}(x)] = f_0(x) + R$, which means that it is unbiased to first-order (at any x, including x near the boundary).

3.3 Universal consistency

Like kNN regression, the kernel smoothing estimator is universally consistent, which means $\mathbb{E}\|\hat{f} - f_0\|_2^2 \to 0$ as $n \to \infty$, under essentially no assumptions, provided that we use a compactly supported kernel K and a bandwidth $h = h_n$ such that $h_n \to 0$ and $n h_n^d \to \infty$ as $n \to \infty$. See Chapter 5.2 of Gyorfi et al. (2002).

Unfortunately, local linear regression does not share this property, and fails to be universally consistent. In theory, this can be rectified using a suitable truncation trick (restricting the domain in (6) to an ℓ_∞ ball whose radius diverges with n); see Chapter 5.3 of Gyorfi et al. (2002). However, this doesn't seem to be in common use in practice.

3.4 Rate of convergence

Assuming that $f_0 \in C^1(L; [0,1]^d)$, the underlying regression function f_0 is Lipschitz continuous on $[0,1]^d$ with some Lipschitz constant $L > 0$, and we place mild conditions on the input distribution, the kernel smoothing estimator with a spherical kernel and bandwidth $h \asymp n^{-1/(2+d)}$ satisfies

$$\mathbb{E}\big[\|\hat{f} - f_0\|_2^2 \,\big|\, x_{1:n}\big] \lesssim n^{-2/(2+d)} \quad \text{in probability,} \tag{7}$$

just like kNN regression. Similar results hold for more general compactly supported kernels.

Proof sketch: as usual, denote $\sigma^2 = \mathrm{Var}[\epsilon_0]$, condition on the inputs $x_{1:n}$, which we omit notationally for simplicity, and condition on x_0. We'll compute the bias and variance separately. In fact we already did the bias calculation, using a first-order Taylor expansion of f_0:

$$\mathbb{E}[\hat{f}(x_0)] = f_0(x_0) + \nabla f_0(x_0)^\mathsf{T} \bigg[\sum_{i=1}^n (x_i - x_0) w_i(x_0) \bigg] + R,$$

where R is a small remainder term that we'll ignore. The Lipschitz condition actually implies that f_0 is differentiable almost everywhere with $\|\nabla f_0(x)\|_2 \leq L$ for almost every x (by Rademacher's theorem). Hence for the squared bias,

$$\mathrm{Bias}^2(\hat{f}(x_0)|x_0) \leq L^2 \bigg\| \sum_{i=1}^n (x_i - x_0) w_i(x_0) \bigg\|_2^2 \leq L^2 \bigg[\sum_{i=1}^n \|x_i - x_0\|_2 w_i(x_0) \bigg]^2.$$

Now we use the fact that that $K(t) = 0$ for $|t| > 1$. Then we can further bound the sum above:

$$\mathrm{Bias}^2(\hat{f}(x_0)|x_0) \leq L^2 h^2 \bigg[\sum_{i=1}^n w_i(x_0) \bigg]_2^2 = L^2 h^2.$$

Now for the variance, using the fact that y_i, $i = 1, \ldots, n$ are independent with variance $\sigma^2 > 0$,

$$\mathrm{Var}(\hat{f}(x_0)|x_0) = \sigma^2 \sum_{i=1}^n w_i(x_0)^2 \leq \sigma^2 \bigg[\max_{i=1,\ldots,n} w_i(x_0) \bigg] \bigg[\sum_{i=1}^n w_i(x_0) \bigg] \leq \sigma^2 \bigg[\max_{i=1,\ldots,n} w_i(x_0) \bigg].$$

For the spherical kernel, using P_n for the empirical distribution of $x_1, \ldots, x_n$, we have

$$w_i(x_0) = \frac{1\{\|x_0 - x_i\|_2 \leq h\}}{n P_n(B_d(x_0, h))} \lesssim \frac{1}{nh^d},$$

where $B_d(x_0, h)$ is the ℓ_2 ball centered at x_0 with radius h. The inequality holds because it can be shown that with high probability, over the distribution of $x_{1:n}$, that $P_n(S) \geq c \cdot \mathrm{vol}(S)$ for any set S that is not "too small", where $c > 0$ is a constant that does not depend on S. Our bias-variance upper bound on the risk is hence:

$$L^2 h^2 + \frac{\sigma^2}{nh^d}.$$

Balancing the two terms so that they are equal gives $h^{2+d} \asymp n^{-1}$, i.e., $h \asymp n^{-1/(2+d)}$. And plugging this in gives the error rate of $n^{-2/(2+d)}$, as claimed.

3.5 Higher-order smoothness

To define and study higher-order smoothness classes, we'll need some more notation: given a multi-index $\alpha = (\alpha_1, \ldots, \alpha_d) \in \mathbb{Z}_+^d$, we write $|\alpha| = \alpha_1 + \cdots + \alpha_d$ and

$$D^\alpha f = \frac{\partial^{|\alpha|} f}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} \ldots \partial x_d^{\alpha_d}}.$$

For an integer $r \geq 0$, exponent $0 < \gamma \leq 1$, radius $L > 0$, and domain $U \subseteq \mathbb{R}^d$, we now define the Hölder class

$$C^{r+\gamma}(L; U) = \Big\{ f : U \to \mathbb{R} \,:\, |D^\alpha f(x) - D^\alpha f(y)| \leq L \|x - y\|_2^\gamma \text{ for all } |\alpha| = r, \text{ and } x, y \in U \Big\}.$$

Note that $C^1(L; U)$ is simply the space of all L-Lipschitz functions on U. Likewise, for an integer $k \geq 1$, $C^k(L; U)$ is the space of all functions on U whose order α^{th} derivative is L-Lipschitz, for all $\alpha = k - 1$.

It can be shown that a minimax lower bound over $C^s(L;[0,1]^d)$, for a constant $L>0$, is

$$\inf_{\hat{f}} \sup_{f_0 \in C^s(L;[0,1]^d)} \mathbb{E}\|\hat{f}-f_0\|_2^2 \gtrsim n^{-2s/(2s+d)}, \tag{8}$$

which generalizes the result we cited earlier for Lipschitz functions in (4).

So now let's think about rate optimal estimators. We saw from (3) and (9) that both kNN regression and kernel smoothing are minimax rate optimal over $C^1(L;[0,1]^d)$. But what about $C^s(L;[0,1]^d)$, for $s>1$? Can these estimators "track" the smoothness of f_0?

The answer is kind of both "yes" and "no". For the "yes" part, it turns out that kernel smoothing can still achieve the optimal convergence rate over $C^{1.5}(L;[0,1]^d)$, and the same is conjectured to be true of kNN. See Chapters 5.3 and 6.3 of Gyorfi et al. (2002).

For the "no" part: neither achieves the optimal rate over $C^2(L;[0,1]^d)$. See again Chapters 5.3 and 6.3 of Gyorfi et al. (2002). An important remark: here we see a big discrepancy between a pointwise analysis and L^2 theory. It can be shown that both kernel smoothing and kNN regression satisfy

$$\mathbb{E}\big[(\hat{f}(x_0)-f_0(x_0))^2\big] \lesssim n^{-4/(4+d)} \quad \text{for any fixed } x_0 \in (0,1)^d.$$

when $f_0 \in C^2(L;[0,1]^d)$. But the same is not true when we integrate over x_0, because the boundary bias inflates the error rate, for both methods.

Lastly, if you recall, we already talked about how to fix boundary bias ... local linear regression to the rescue! As one would hope, this is indeed rate optimal over $C^2(L;[0,1]^d)$, i.e., assuming that $f_0 \in C^2(L;[0,1]^d)$, and we place mild conditions on the input distribution as usual, the local linear regression estimator with bandwidth $h \asymp n^{-1/(4+d)}$ satisfies

$$\mathbb{E}\big[\|\hat{f}-f_0\|_2^2 \,\big|\, x_{1:n}\big] \lesssim n^{-4/(4+d)} \quad \text{in probability}, \tag{9}$$

for general compactly supported kernels. We can see this matches the rate in (8) for $s=2$.

3.6 Local polynomials and higher-order kernels

How can we get optimal error rates for even smoother functions, in $C^s(L;[0,1]^d)$ for $s>2$? With kernels there are basically two options: use local polynomials, or use higher-order kernels.

Local polynomials build on the idea behind local linear regression (as an extension of kernel smoothing). Consider $d=1$, for concreteness. Define $\hat{f}(x)=\hat{\alpha}_x+\sum_{j=1}^k \hat{\beta}_{x,j}x^j$, where the parameters $\hat{\alpha}_x,\hat{\beta}_{x,1},\ldots,\hat{\beta}_{x,k}$ now solve (cf. problem (6)):

$$\underset{\alpha,\beta_1,\ldots,\beta_k}{\text{minimize}} \sum_{i=1}^n K\bigg(\frac{\|x-x_i\|_2}{h}\bigg)\bigg(y_i-\alpha-\sum_{j=1}^k \beta_j x_i^j\bigg)^2. \tag{10}$$

This is called (k^{th} order) *local polynomial regression*. As before, we can express the prediction at x as

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

where now $b(x)=(1,x,\ldots,x^k)$, B is an $n\times(k+1)$, $B\in\mathbb{R}^{n\times(k+1)}$ is the matrix whose i^{th} row is $b(x_i)$, and $\Omega\in\mathbb{R}^{n\times n}$ is the same diagonal matrix with kernel weights as before. Hence again, local polynomial regression is a linear smoother.

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

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

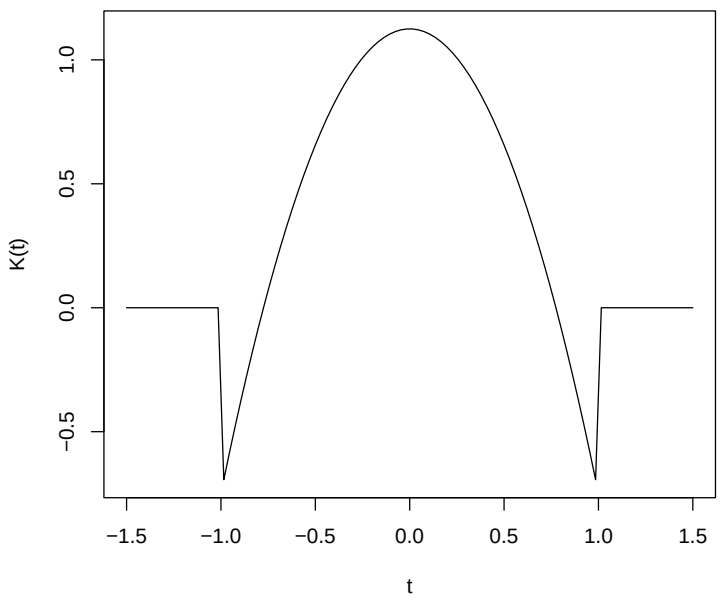

Figure 4: *A higher-order kernel function—specifically, a kernel of order 4.*

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! (Higher-order kernels, in fact, have an interesting connection to smoothing splines, which we'll learn a bit later on.)

Both local polynomials and higher-order kernels can achieve optimal rates over higher-order Hölder classes, where the order of the polynomial or kernel is adjusted with the order of smoothness. We do not give the details here (but see, e.g., Chapter 1.6.1 of Tsybakov (2009) for an analysis of local polynomials).

References

Laszlo Gyorfi, Michael Kohler, Adam Krzyzak, and Harro Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer, 2002.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer, 2009. Second edition.

Alexandre B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2009.

Larry Wasserman. *All of Nonparametric Statistics*. Springer, 2006.