Basis expansions and Kernel methods

김지수 (Jisu KIM)

통계적 기계학습(Statistical Machine Learning), 2024 1st semester

The lecture note is a minor modification of the lecture notes from Prof. Yongdai Kim's "Statistical Machine Learning", and Prof Larry Wasserman and Ryan Tibshirani's "Statistical Machine Learning". Also, see Section 5 and 6 from [7].

1 Review

1.1 Basic Model for Supervised Learning

- Input(입력) / Covariate(설명 변수) : $x \in \mathbb{R}^p$, so $x = (x_1, \dots, x_p)$.
- Output(출력) / Response(반응 변수): $y \in \mathcal{Y}$. If y is categorical, then supervised learning is "classification", and if y is continuous, then supervised learning is "regression".
- Model(모형) :
$$y \approx f(x).$$
If we include the error ϵ to the model, then it can be also written as
$$y = \phi(f(x), \epsilon).$$
For many cases, we assume additive noise, so
$$y = f(x) + \epsilon.$$
- Assumption(가정): f belongs to a family of functions $\mathcal{M}$. This is the assumption of a model: a model can be still used when the corresponding assumption is not satisfied in your data.
- Loss function(손실 함수): $\ell(y, a)$. A loss function measures the difference between estimated and true values for an instance of data.
- Training data(학습 자료): $\mathcal{T} = \{(y_i, x_i), i = 1, \dots, n\}$, where (y_i, x_i) is a sample from a probability distribution P_i. For many cases we assume i.i.d., or x_i's are fixed and y_i's are i.i.d..
- Goal(목적): we want to find f that minimizes the expected prediction error,
$$f^0 = \arg\min_{f \in \mathcal{F}} \mathbb{E}_{(Y,X)\sim P}\left[\ell(Y, f(X))\right].$$
Here, $\mathcal{F}$ can be different from $\mathcal{M}$; $\mathcal{F}$ can be smaller then $\mathcal{M}$.
- Prediction model(예측 모형): f^0 is unknown, so we estimate f^0 by $\hat{f}$ using data. For many cases we minimizes on the empirical prediction error, that is taking the expectation on the empirical distribution $P_n = \frac{1}{n}\sum_{i=1}^n \delta_{(Y_i, X_i)}$.
$$\hat{f} = \arg\min_{f \in \mathcal{F}} \mathbb{E}_{P_n}\left[\ell(Y, f(X))\right] = \arg\min_{f \in \mathcal{F}} \frac{1}{n}\sum_{i=1}^n \ell(Y_i, f(X_i)).$$
- Prediction(예측): if $\hat{f}$ is a predicted function, and x is a new input, then we predict unknown y by $\hat{f}(x)$.

1.2 Linear Regression

From the additive noise model

$$y = f(x) + \epsilon,\ f \in \mathcal{M},$$

Linear Regression Model (선형회귀모형) is that

$$\mathcal{M} = \mathcal{F} = \left\{ \beta_0 + \sum_{j=1}^{p} \beta_j x_j : \beta_j \in \mathbb{R} \right\}.$$

For estimating β, we use least squares: suppose the training data is $\{(y_i, x_{ij}) : 1 \leq i \leq n, 1 \leq j \leq p\}$. We use square loss

$$\ell(y, a) = (y - a)^2,$$

then the eimpirical loss becomes the residual sum of square (RSS) as

$$\begin{aligned} RSS(\beta) &= \sum_{i=1}^{n} (y_i - f(x_i))^2 \\ &= \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 . \end{aligned}$$

Let $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)$ be the nimimizor of RSS, then the predicted function is

$$\hat{f}(x) = \hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j x_j.$$

2 Introduction

For linear regression (선형 회귀) $y = x^\top \beta + \epsilon$, the mean response $\mathbb{E}[y|X = x] = x^\top \beta$ is linear with respect to x. But this linear relation may not hold: The nonparametric regression (비모수 회귀), or non-linear regression (비선형 회귀), we just assume

$$y = f(x) + \epsilon,\ f \in \mathcal{M}.$$

For training data $\mathcal{T} = \{(y_i, x_i), i = 1, \ldots, n\}$, where (y_i, x_i) is a sample from a probability distribution P_i. For many cases we assume i.i.d., or x_i's are fixed and y_i's are i.i.d. Then we can write

$$y_i = f(x_i) + \epsilon_i, \quad i = 1, \ldots, n.$$

For theoretical analysis, we assume ϵ_i, $i = 1, \ldots, n$ are i.i.d. random errors, with mean zero. $\mathbb{E}[\epsilon_i] = 0$ ensures that f is the minimizer of $\mathbb{E}_{(Y,X)\sim P}\left[(Y - f(X))^2\right]$ when x_i's are i.i.d.

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

Between x_i's being i.i.d. or fixed, theories are not completely the same (some theory is sharper when we assume fixed input points, especially evenly spaced input points), but for the most part the theory is quite similar. Therefore, in what follows, we won't be very precise about which setup we assume—random or fixed inputs—because it mostly doesn't matter when introducing nonparametric regression estimators and discussing basic properties

To solve low-dimensional non-linear regression (비선형 회귀) $y = f(x) + \epsilon$, there are two approaches:

- Basis expansions (기저 전개): Linear models on Take non-linear transformation (비선형 변환) of input X and then solve linear regression (선형 회귀).
    - regression function (회귀함수) becomes $f(x) = \sum_{j=1}^{q} \beta_j h_j(x)$. Each $h_j : \mathbb{R}^p \to \mathbb{R}$ is called basis function (기저함수)
    - polynomial regression (다항함수), regression spline (회귀스플라인), smoothing spline (평활스플라인)
- Kernel smoothing (핵평활): Solve linear regression (선형 회귀) locally (국소적) using kernel function (핵함수)
    - Nadaaya-Watson estimator, local linear regression (국소선형회귀)

2.1 Notation

- We will define an empirical norm $\|\cdot\|_n$ in terms of the training points x_i, $i = 1, \ldots, n$, acting on functions $f : \mathbb{R}^d \to \mathbb{R}$, by
$$\|f\|_n^2 = \frac{1}{n}\sum_{i=1}^{n} f^2(x_i).$$
This makes sense no matter if the inputs are fixed or random (but in the latter case, it is a random norm)

- When the inputs are considered random, we will write P_X for the distribution of X, and we will define the L_2 norm $\|\cdot\|_2$ in terms of P_X, acting on functions $f : \mathbb{R}^d \to \mathbb{R}$, by
$$\|f\|_2^2 = \mathbb{E}[f^2(X)] = \int f^2(x)\, dP_X(x).$$
So when you see $\|\cdot\|_2$ in use, it is a hint that the inputs are being treated as random

- A quantity of interest will be the (squared) error associated with an estimator $\hat{f}$ of f_0, which can be measured in either norm:
$$\|\hat{f} - f_0\|_n^2 \quad \text{or} \quad \|\hat{f} - f_0\|_2^2.$$
In either case, this is a random quantity (since $\hat{f}$ is itself random). We will study bounds in probability or in expectation. The expectation of the errors defined above, in terms of either norm (but more typically the L_2 norm) is most properly called the risk; but we will often be a bit loose in terms of our terminology and just call this the error

2.2 Holder Spaces and Sobolev Spaces

The class of Lipschitz functions $H(1, L)$ on $T \subset \mathbb{R}$ is the set of functions $g : T \to \mathbb{R}$ such that
$$|g(y) - g(x)| \le L|x - y| \ \text{ for all } x, y \in T.$$

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

Let $T \subset \mathbb{R}$ and let β be an integer. The Holder space $H(\beta, L)$ is the set of functions $g : T \to \mathbb{R}$ such that g is $\ell = \beta - 1$ times differentiable and satisfies
$$|g^{(\ell)}(y) - g^{(\ell)}(x)| \le L|x - y| \ \text{ for all } x, y \in T.$$

(There is an extension to real valued β but we will not need that.) If $g \in H(\beta, L)$ and $\ell = \beta - 1$, then we can define the Taylor approximation of g at x by
$$\tilde{g}(y) = g(y) + (y - x)g'(x) + \cdots + \frac{(y - x)^\ell}{\ell!} g^{(\ell)}(x)$$
and then $|g(y) - \tilde{g}(y)| \le |y - x|^\beta$.

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

Let β be a positive integer. Define the *Hölder class*
$$H_d(\beta, L) = \left\{ g : \ |D^s g(x) - D^s g(y)| \le L\|x - y\|, \quad \text{for all } s \text{ such that } |s| = \beta - 1, \text{ and all } x, y \right\}. \tag{1}$$

For example, if $d = 1$ and $\beta = 2$ this means that
$$|g'(x) - g'(y)| \le L\,|x - y|, \quad \text{for all } x, y.$$

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

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

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

where

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

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

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

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

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

Again this extends naturally to $\mathbb{R}^d$. Also, there is an extension to non-integer β. It can be shown that if T is bounded, then $H_d(\beta, L) \subset S_d(\beta, L')$ for appropriate L and L'.

3 Linear smoothers

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

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

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

the trace of the smooth matrix S.

4 Basis Expansions

Consider a mapping $h : \mathbb{R}^p \to \mathbb{R}^q$. Then, consider a linear model

$$\mathcal{F} = \left\{ h(x)^\top \beta : \beta \in \mathbb{R}^q \right\}.$$

When h is nonlinear, the resulting model is also nonlinear. Typically, q is much larger than p.
Examples are:

- Polynomial regression
- Special functions: $h(x) = \exp(x_1 + x_2)$
- locally constant function: $h_j(x) = I(l < x_j < u)$.
- Regression spline: locally polynomial functions
- Smoothing spline: penalized locally polynomial functions
- Wavelet (not covered!)

4.1 Polynomial regression

The m-th order polynomial model is given as

$$f(x) = \beta_0 + \sum_{j=1}^{p} \beta_j x_j + \sum_{j<k} \beta_{jk} x_j x_k \cdots + \sum_{j_1<j_2<\cdots<j_m} \beta_{j_1,j_2,\cdots,j_m} \prod_{k=1}^{m} x_{j_k}.$$

- Main effect terms: $\beta_j x_j$
- Second order interaction terms: $\beta_{jk} x_j x_k$
- mth order interaction terms: $\beta_{j_1,j_2,\cdots,j_m} \prod_{k=1}^{m} x_{j_k}$
- When p is small, we can use relatively large m. But, when p is large, we use small m due to difficulties in computation as well as interpretation.
- In practice, we are satisfied with $m = 2$ for high dimensional problems.

5 Regression splines, smoothing splines

5.1 Splines

- Regression splines and smoothing splines are motivated from a different perspective than kernels and local polynomials; in the latter case, we started off with a special kind of local averaging, and moved our way up to a higher-order local models. With regression splines and smoothing splines, we build up our estimate globally, from a set of select basis functions
- These basis functions, as you might guess, are *splines*. Let's assume that $d = 1$ for simplicity. (We'll stay in the univariate case, for the most part, in this section.) A kth-order spline f is a piecewise polynomial function of degree k that is continuous and has continuous derivatives of orders $1, \ldots, k-1$, at its knot points. Specifically, there are $t_1 < \ldots < t_p$ such that f is a polynomial of degree k on each of the intervals

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

  and $f^{(j)}$ is continuous at $t_1, \ldots, t_p$, for each $j = 0, 1, \ldots, k-1$
- Splines have some special (some might say: amazing!) properties, and they have been a topic of interest among statisticians and mathematicians for a very long time. See [1] for an in-depth coverage. Informally, a spline is a lot smoother than a piecewise polynomial, and so modeling with splines can serve as a way of reducing the variance of fitted estimators. See Figure 1
- A bit of statistical folklore: it is said that a cubic spline is so smooth, that one cannot detect the locations of its knots by eye!
- How can we parametrize the set of a splines with knots at $t_1, \ldots, t_p$? The most natural way is to use the *truncated power basis*, $g_1, \ldots, g_{p+k+1}$, defined as

  $$\begin{aligned} g_1(x) = 1,\ g_2(x) = x,\ \ldots\ g_{k+1}(x) = x^k, \\ g_{k+1+j}(x) = (x - t_j)_+^k, \quad j = 1, \ldots, p. \end{aligned} \tag{4}$$

  (Here x_+ denotes the positive part of x, i.e., $x_+ = \max\{x, 0\}$.) From this we can see that the space of kth-order splines with knots at $t_1, \ldots, t_p$ has dimension $p + k + 1$
- While these basis functions are natural, a much better computational choice, both for speed and numerical accuracy, is the *B-spline* basis. This was a major development in spline theory and is now pretty much the standard in software. The key idea: B-splines have local support, so a basis matrix that we form with them (to be defined below) is banded. See [1] or the Appendix of Chapter 5 in [7] for details

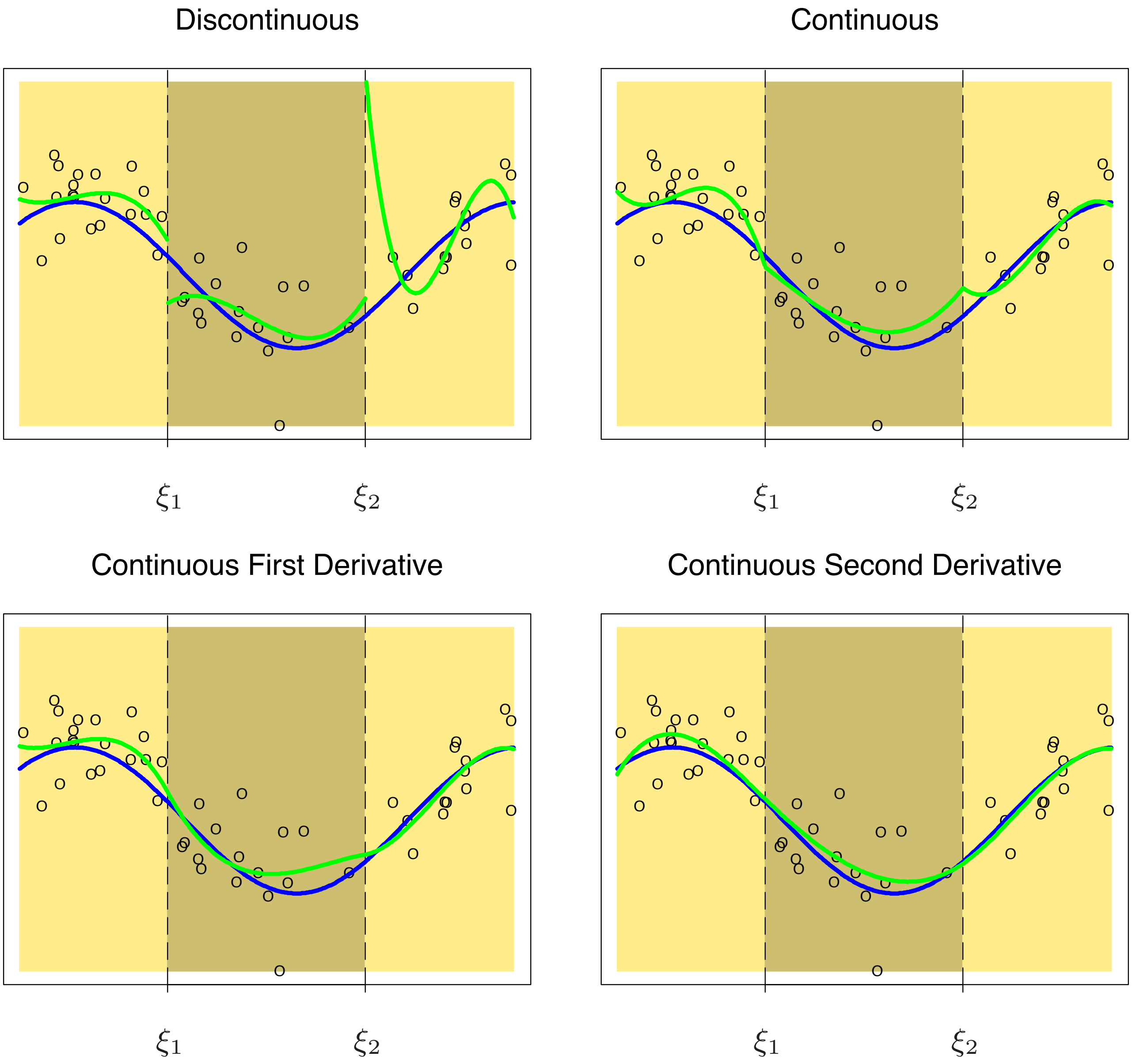

Figure 1: *Illustration of the effects of enforcing continuity at the knots, across various orders of the derivative, for a cubic piecewise polynomial. From Figure 5.1 of* [7]

5.2 Regression splines

- A first idea: let's perform regression on a spline basis. In other words, given inputs $x_1, \dots, x_n$ and responses $y_1, \dots, y_n$, we consider fitting functions f that are kth-order splines with knots at some chosen locations $t_1, \dots t_p$. This means expressing f as
$$f(x) = \sum_{j=1}^{p+k+1} \beta_j g_j(x),$$
where $\beta_1, \dots, \beta_{p+k+1}$ are coefficients and $g_1, \dots, g_{p+k+1}$, are basis functions for order k splines over the knots $t_1, \dots, t_p$ (e.g., the truncated power basis or B-spline basis)

- Letting $y = (y_1, \dots, y_n) \in \mathbb{R}^n$, and defining the basis matrix $G \in \mathbb{R}^{n \times (p+k+1)}$ by
$$G_{ij} = g_j(x_i), \quad i = 1, \dots, n, \; j = 1, \dots, p+k+1,$$
we can just use least squares to determine the optimal coefficients $\hat{\beta} = (\hat{\beta}_1, \dots, \hat{\beta}_{p+k+1})$,
$$\hat{\beta} = \mathrm{argmin}_{\beta \in \mathbb{R}^{p+k+1}} \; \|y - G\beta\|_2^2,$$
which then leaves us with the fitted *regression spline* $\hat{f}(x) = \sum_{j=1}^{p+k+1} \hat{\beta}_j g_j(x)$

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

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

- For the computation of regression splines, we fix the order of local polynomial, typically less than or equal to 3 (The cubic spline is a popular one). The parameters we have to estimate is the number of knots, the knot locations and regression coefficients. Once p and t_l's are fixed, the regression coefficients can be estimated easily. Again, the knot selection is computationally demanding and unstable. A better approach is smoothing spline.

5.3 Natural splines

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

- A way to remedy this problem is to force the piecewise polynomial function to have a lower degree to the left of the leftmost knot, and to the right of the rightmost knot—this is exactly what *natural splines* do. A natural spline of order k, with knots at $t_1 < \ldots < t_p$, is a piecewise polynomial function f such that
  - f is a polynomial of degree k on each of $[t_1, t_2], \dots, [t_{p-1}, t_p]$,
  - 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 $1, \dots, k-1$ at $t_1, \dots, t_p$.

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

- What is the dimension of the span of kth order natural splines with knots at $t_1, \dots, t_p$? Recall for splines, this was $p + k + 1$ (the number of truncated power basis functions). For natural splines, we can compute this dimension by counting:
$$\underbrace{(k+1) \cdot (p-1)}_{a} + \underbrace{\left(\frac{(k-1)}{2} + 1\right) \cdot 2}_{b} - \underbrace{k \cdot p}_{c} = p.$$
Above, a is the number of free parameters in the interior intervals $[t_1, t_2], \dots, [t_{p-1}, t_p]$, b is the number of free parameters in the exterior intervals $(-\infty, t_1]$, $[t_p, \infty)$, and c is the number of constraints at the knots $t_1, \dots, t_p$. The fact that the total dimension is p is amazing; this is independent of k!

- Note that there is a variant of the truncated power basis for natural splines, and a variant of the B-spline basis for natural splines. Again, B-splines are the preferred parametrization for computational speed and stability

- Natural splines of cubic order is the most common special case: these are smooth piecewise cubic functions, that are simply linear beyond the leftmost and rightmost knots

5.4 Smoothing splines

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

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

$$\hat{f} = \mathrm{argmin}_f \; \sum_{i=1}^n \big(y_i - f(x_i)\big)^2 + \lambda \int_0^1 \big(f^{(m)}(x)\big)^2 \, dx, \quad \text{where } m = (k+1)/2. \tag{5}$$

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

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

$$\hat{f} = \mathrm{argmin}_f \; \sum_{i=1}^n \big(y_i - f(x_i)\big)^2 + \lambda \int_0^1 f''(x)^2 \, dx \tag{6}$$

- Remarkably, it so happens that the minimizer in the general smoothing spline problem (5) is unique, and is a natural kth-order spline with knots at the input points $x_1, \ldots, x_n$! Here we give a proof for the cubic case, $k = 3$, from [5] (see also Exercise 5.7 in [7])

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

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

  Note that this would in fact prove that we can restrict our attention in (6) to natural splines with knots at $x_1, \ldots, x_n$

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

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

  Consider

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

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

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

From this, it follows that

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

and therefore

$$\int_0^1 f''(x)^2 \, dx \leq \int_0^1 \tilde{f}''(x)^2 \, dx, \tag{7}$$

with equality if and only if $h''(x) = 0$ for all $x \in [0, 1]$. Note that $h'' = 0$ implies that h must be linear, and since we already know that $h(x_j) = 0$ for all $j = 1, \ldots, n$, this is equivalent to $h = 0$. In other words, the inequality (7) holds strictly except when $\tilde{f} = f$, so the solution in (6) is uniquely a natural spline with knots at the inputs

5.5 Finite-dimensional form

- The key result presented above tells us that we can choose a basis $\eta_1, \ldots, \eta_n$ for the set of kth-order natural splines with knots over $x_1, \ldots, x_n$, and reparametrize the problem (5) as

$$\hat{\beta} = \mathrm{argmin}_{\beta \in \mathbb{R}^n} \sum_{i=1}^n \Big(y_i - \sum_{j=1}^n \beta_j \eta_j(x_i)\Big)^2 + \lambda \int_0^1 \Big(\sum_{j=1}^n \beta_j \eta_j^{(m)}(x)\Big)^2 \, dx. \tag{8}$$

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

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

$$N_{ij} = \eta_j(x_i) \quad \text{and} \quad \Omega_{ij} = \int_0^1 \eta_i^{(m)}(x) \eta_j^{(m)}(x) \, dx \quad \text{for } i, j = 1, \ldots, n, \tag{9}$$

  the problem in (8) can be written more succintly as

$$\hat{\beta} = \mathrm{argmin}_{\beta \in \mathbb{R}^n} \|y - N\beta\|_2^2 + \lambda \beta \Omega \beta, \tag{10}$$

  showing the smoothing spline problem to be a type of generalized ridge regression problem. In fact, the solution in (10) has the explicit form

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

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

$$\hat{\mu} = N(N^T N + \lambda \Omega)^{-1} N^T y. \tag{11}$$

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

- A special property of smoothing splines: the fitted values in (11) can be computed in $O(n)$ operations. This is achieved by forming N from the B-spline basis (for natural splines), and in this case the matrix $N^T N + \Omega I$ ends up being banded (with a bandwidth that only depends on the polynomial order k). In practice, smoothing spline computations are extremely fast

5.6 Reinsch form

- It is informative to rewrite the fitted values in (11) is what is called Reinsch form,
$$\begin{aligned}\hat{\mu} &= N(N^TN + \lambda\Omega)^{-1}N^Ty \\ &= N\Big(N^T\big(I + \lambda(N^T)^{-1}\Omega N^{-1}\big)N\Big)^{-1}N^Ty \\ &= (I + \lambda Q)^{-1}y, \qquad (12)\end{aligned}$$
where $Q = (N^T)^{-1}\Omega N^{-1}$

- Note that this matrix Q does not depend on λ. If we compute an eigendecomposition $Q = UDU^T$, then the eigendecomposition of $S = N(N^TN + \lambda\Omega)^{-1} = (I + \lambda Q)^{-1}$ is
$$S = \sum_{j=1}^{n} \frac{1}{1 + \lambda d_j} u_j u_j^T,$$
where $D = \text{diag}(d_1, \ldots, d_n)$

- Therefore the smoothing spline fitted values are $\hat{\mu} = Sy$, i.e.,
$$\hat{\mu} = \sum_{j=1}^{n} \frac{u_j^Ty}{1 + \lambda d_j} u_j. \qquad (13)$$
Interpretation: smoothing splines perform a regression on the orthonormal basis $u_1, \ldots, u_n \in \mathbb{R}^n$, yet they shrink the coefficients in this regression, with more shrinkage assigned to eigenvectors u_j that correspond to large eigenvalues d_j

- So what exactly are these basis vectors $u_1, \ldots, u_n$? These are known as the *Demmler-Reinsch basis*, and a lot of their properties can be worked out analytically [2]. Basically: the eigenvectors u_j that correspond to smaller eigenvalues d_j are smoother, and so with smoothing splines, we shrink less in their direction. Said differently, by increasing λ in the smoothing spline estimator, we are tuning out the more wiggly components. See Figure 2

5.7 Error rates

- Recall the *Sobolev class* of functions $S_1(m, C)$: for an integer $m \geq 0$ and $C > 0$, to contain all m times differentiable functions $f : \mathbb{R} \to \mathbb{R}$ such that
$$\int \big(f^{(m)}(x)\big)^2 \, dx \leq C^2.$$
(The Sobolev class $S_d(m, C)$ in d dimensions can be defined similarly, where we sum over all partial derivatives of order m.)

- Suppose $y = f_0(x) + \epsilon$, and assume $f_0 \in S_1(m, C)$ for the underlying regression function, where $C > 0$ is a constant. The smoothing spline estimator $\hat{f}$ in (5) of polynomial order $k = 2m - 1$ with tuning parameter $\lambda \asymp n^{1/(2m+1)} \asymp n^{1/(k+2)}$ satisfies
$$\|\hat{f} - f_0\|_n^2 \lesssim n^{-2m/(2m+1)} \quad \text{in probability.}$$
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 [10]

- This rate is seen to be minimax optimal over $S_1(m, C)$ (e.g., [8]). Also, it is worth noting that the Sobolev $S_1(m, C)$ and Holder $S_1(m, L)$ classes are *equivalent* in the following sense: given $S_1(m, C)$ for a constant $C > 0$, there are $L_0, L_1 > 0$ such that
$$H_1(m, L_0) \subseteq S_1(m, C) \subseteq H_1(m, L_1).$$
The first containment is easy to show; the second is far more subtle, and is a consequence of the Sobolev embedding theorem. (The same equivalences hold for the d-dimensional versions of the Sobolev and Holder spaces.)

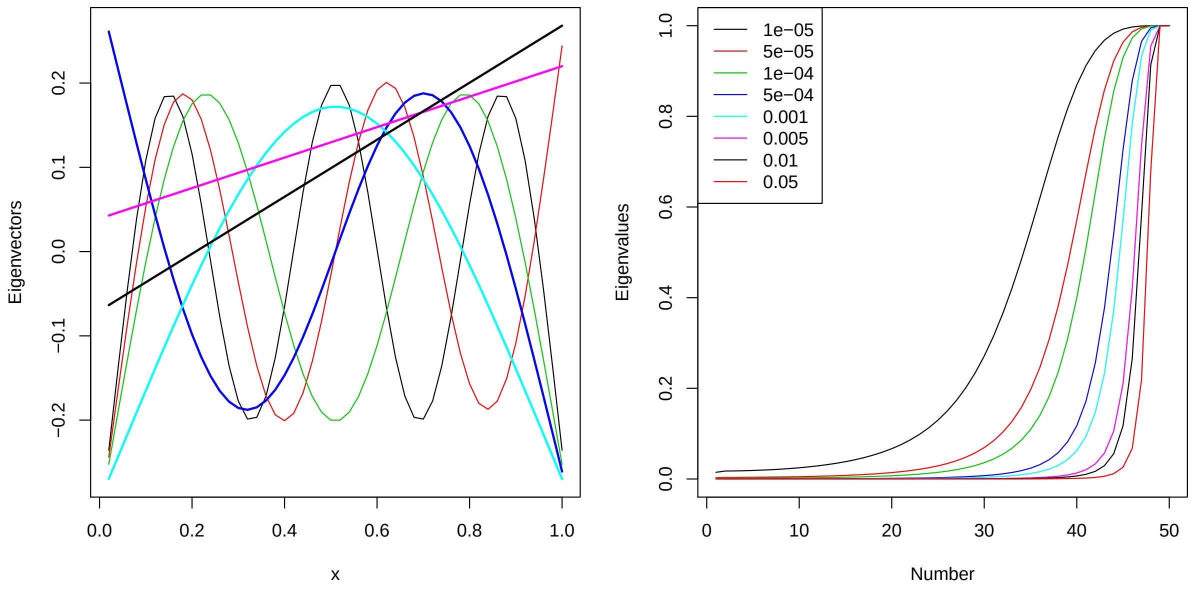

Figure 2: *Eigenvectors and eigenvalues for the Reinsch form of the cubic smoothing spline operator, defined over* $n = 50$ *evenly spaced inputs on* $[0, 1]$. *The left plot shows the bottom 7 eigenvectors of the Reinsch matrix* Q. *We can see that the smaller the eigenvalue, the "smoother" the eigenvector. The right plot shows the weights* $w_j = 1/(1+\lambda d_j)$, $j = 1, \ldots, n$ *implicitly used by the smoothing spline estimator (13), over 8 values of* λ. *We can see that when* λ *is larger, the weights decay faster, so the smoothing spline estimator places less weight on the "nonsmooth" eigenvectors*

5.8 Multivariate splines

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

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

  which is rotationally invariant. Both of these concepts are discussed in Chapter 7 of [5] (see also Chapters 15 and 20.4 of [6])

- The multivariate extensions (thin-plate and tensor-product) of splines 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's (argurably) easier to study reproducing kernel Hilbert space estimators. We'll see, in fact, that this covers smoothing splines (and thin-plate splines) as a special case

6 k-nearest-neighbors regression

To motivate kernel methods, we start from: *k-nearest-neighbors* regression. We fix an integer $k \geq 1$ and define

$$\hat{f}(x) = \frac{1}{k} \sum_{i \in \mathcal{N}_k(x)} Y_i, \tag{14}$$

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

This is not at all a bad estimator, and you will find it used in lots of applications, in many cases probably because of its simplicity. By varying the number of neighbors k, we can achieve a wide range of flexibility in the estimated function $\hat{f}$, with small k corresponding to a more flexible fit, and large k less flexible.

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

$$\hat{f}(x) = \sum_{i=1}^{n} w_i(x) Y_i, \tag{15}$$

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 $\hat{f}(x)$.

The representation (15) also reveals that the k-nearest-neighbors estimate is in a class of estimates we call *linear smoothers*, i.e., writing $Y = (Y_1, \ldots, Y_n) \in \mathbb{R}^n$, the vector of fitted values

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

can simply be expressed as $\hat{\mu} = SY$. (To be clear, this means that for fixed inputs $X_1, \ldots, X_n$, the vector of fitted values $\hat{\mu}$ is a linear function of Y; it does not mean that $\hat{f}(x)$ need behave linearly as a function of x.) This class is quite large, and contains many popular estimators, as we'll see in the coming sections.

The k-nearest-neighbors estimator is *universally consistent*, which means $\mathbb{E}\|\hat{f} - f_0\|_2^2 \to 0$ 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 [6].

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

$$\mathbb{E}\|\hat{f} - f_0\|_2^2 \lesssim n^{-2/(2+d)}. \tag{16}$$

See Chapter 6.3 of [6]. In fact this is optimal.

6.1 Curse of dimensionality

Note that the above error rate $n^{-2/(2+d)}$ exhibits a very poor dependence on the dimension d. To see it differently: given a small $\epsilon > 0$, think about how large we need to make n to ensure that $n^{-2/(2+d)} \leq \epsilon$. Rearranged, this says $n \geq \epsilon^{-(2+d)/2}$. That is, as we increase d, we require *exponentially more samples* n to achieve an error bound of ϵ. See Figure 3 for an illustration with $\epsilon = 0.1$

In fact, this phenomenon is not specific to k-nearest-neighbors, but a reflection of the *curse of dimensionality*, the principle that estimation becomes exponentially harder as the number of dimensions increases. This is made precise by minimax theory: we cannot hope to do better than the rate in(16) over $H_d(1, L)$, which we write for the space of L-Lipschitz functions in d dimensions, for a constant $L > 0$. It can be shown that

$$\inf_{\hat{f}} \sup_{f_0 \in H_d(1,L)} \mathbb{E}\|\hat{f} - f_0\|_2^2 \gtrsim n^{-2/(2+d)}, \tag{17}$$

where the infimum above is over all estimators $\hat{f}$. See Chapter 3.2 of [6].

So why can we sometimes predict well in high dimensional problems? Presumably, it is because f_0 often (approximately) satisfies stronger assumptions. This suggests we should look at classes of functions with more structure. One such example is the additive model.

7 Kernel Smoothing and Local Polynomials

7.1 Kernel smoothing

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

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

Three common examples are the box-car kernel:

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

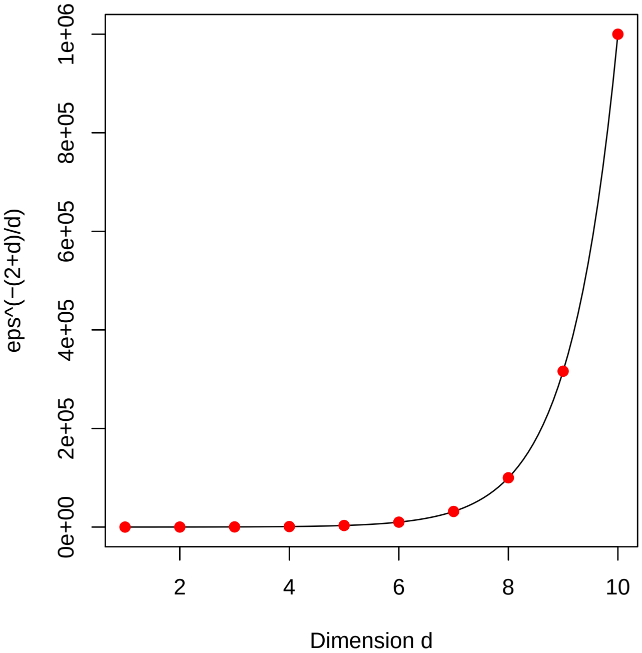

Figure 3: *The curse of dimensionality, with* $\epsilon = 0.1$

the Gaussian kernel:

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

and the Epanechnikov kernel:

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

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

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

$$(x) = \frac{\sum_{i=1}^{n} K\left(\frac{\|x - X_i\|_2}{h}\right) Y_i}{\sum_{i=1}^{n} K\left(\frac{\|x - X_i\|_2}{h}\right)} = \sum_i w_i(x) Y_i \tag{18}$$

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

In comparison to the k-nearest-neighbors estimator in (14), which can be thought of as a raw (discontinuous) moving average of nearby responses, the kernel estimator in (18) is a smooth moving average of responses. See Figure 4 for an example with $d = 1$.

7.2 Error Analysis

The kernel smoothing estimator is universally consistent ($\mathbb{E}\|\hat{f} - f_0\|_2^2 \to 0$ as $n \to \infty$, with no assumptions other than $\mathbb{E}(Y^2) \le \infty$), 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 [6]. We can say more.

Theorem. Suppose that $d = 1$ and that f_0'' is bounded. Also suppose that X has a non-zero, differentiable density p and that the support is unbounded. Then, the risk is

$$\begin{aligned} R_n = {} & \frac{h_n^4}{4} \left(\int x^2 K(x) dx \right)^2 \int \left(f_0''(x) + 2 f_0'(x) \frac{p'(x)}{p(x)} \right)^2 dx \\ & + \frac{\sigma^2 \int K^2(x) dx}{n h_n} \int \frac{dx}{p(x)} + o\left(\frac{1}{n h_n}\right) + o(h_n^4) \end{aligned}$$

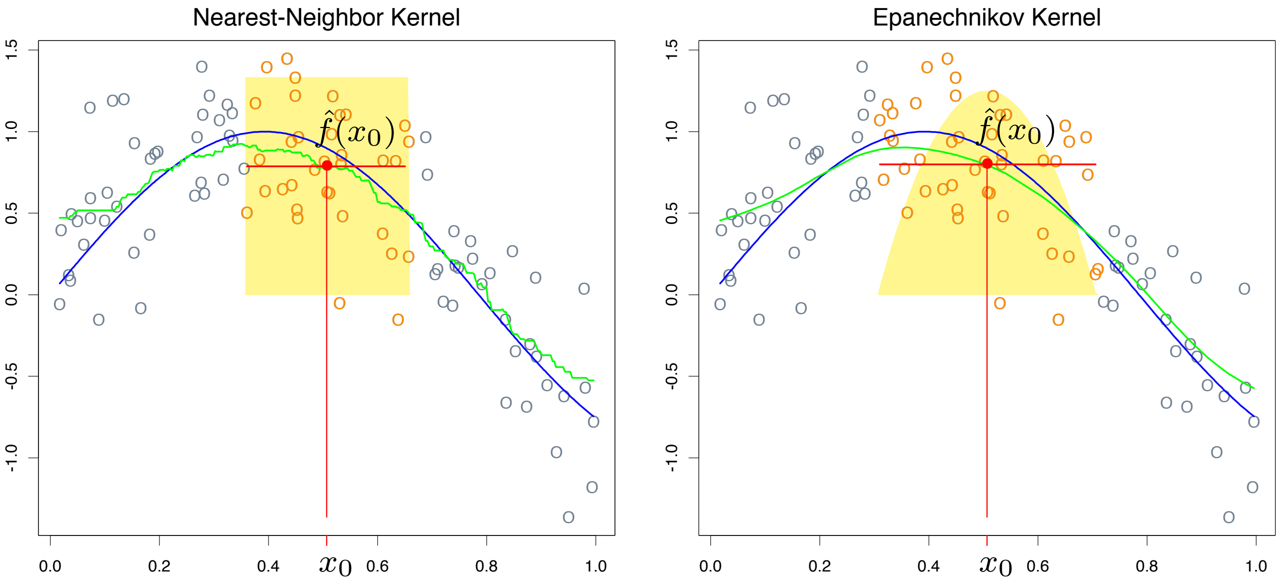

Figure 4: *Comparing k-nearest-neighbor and Epanechnikov kernels, when $d = 1$. From Figure 6.1 of* [7]

where p is the density of P_X.

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

If the support has boundaries then there is bias of order $O(h)$ near the boundary. This happens because of the asymmetry of the kernel weights in such regions. See Figure 5. Specifically, the bias is of order $O(h^2)$ in the interior but is of order $O(h)$ near the boundaries. The risk then becomes $O(h^3)$ instead of $O(h^4)$. We'll fix this problems using local linear smoothing. Also, the result above depends on assuming that P_X has a density. We can drop that assumption (and allow for boundaries) and get a slightly weaker result due to Gyorfi, Kohler, Krzyzak and Walk (2002).

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

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

where P_n is the empirical measure and $B(x, h) = \{u : \ \|x - u\| \leq h\}$. If the denominator is 0 we define $\hat{f}(x) = 0$. 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

$$\sup_{P \in H_d(1,L)} \mathbb{E}\|\hat{f} - f_0\|_P^2 \leq c_1 h^2 + \frac{c_2}{nh^d}. \tag{19}$$

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

$$\sup_{P \in H_d(1,L)} \mathbb{E}\|\hat{f} - f_0\|_P^2 \leq \frac{c}{n^{2/(d+2)}}. \tag{20}$$

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

Recall from (17) 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

$$\inf_{\hat{f}} \sup_{f_0 \in H_d(\alpha,L)} \mathbb{E}\|\hat{f} - f_0\|_2^2 \gtrsim n^{-2\alpha/(2\alpha+d)}, \tag{21}$$

see again Chapter 3.2 of [6]. However, as we saw above, with extra conditions, we got the rate $n^{-4/(4+d)}$ which is minimax for $H_d(2, L)$. We'll get that rate under weaker conditions with local linear regression.

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

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

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

7.3 Local Linear Regression

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

To build intuition, another way to view the kernel estimator in (18) is the following: at each input x, define the estimate $\hat{f}(x) = \hat{\theta}_x$, where $\hat{\theta}_x$ is the minimizer of

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

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

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

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

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

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

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

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

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

which should look familiar to you from weighted least squares.

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

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

Using a Taylor expansion of f_0 about x,

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

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

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

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

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

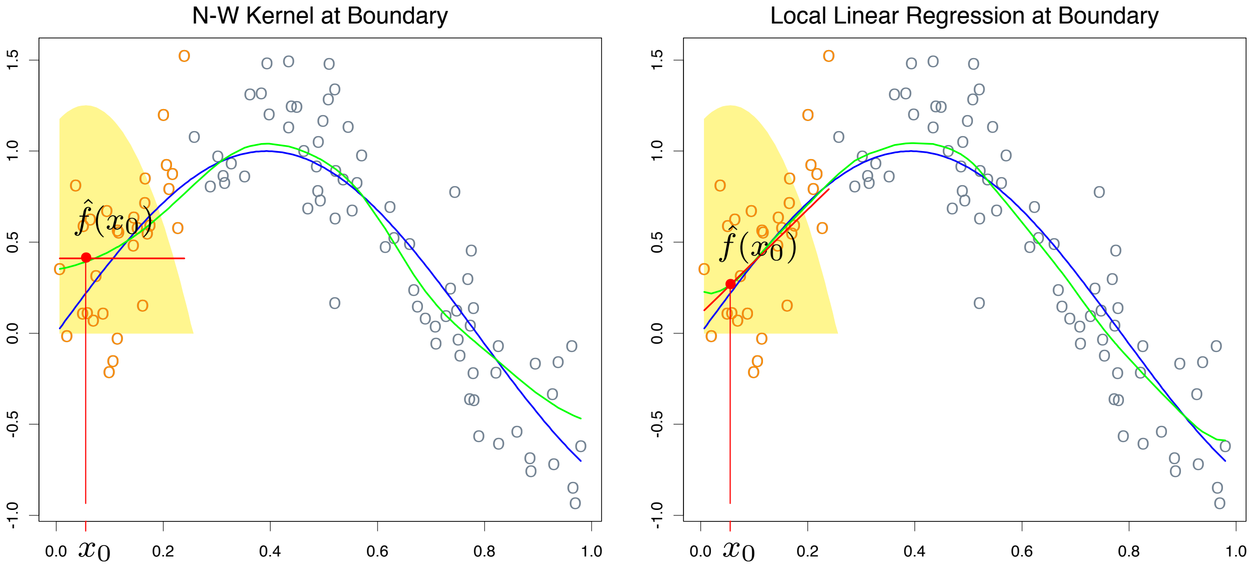

Figure 5: *Comparing (Nadaraya-Watson) kernel smoothing to local linear regression; the former is biased at the boundary, the latter is unbiased (to first-order). From Figure 6.3 of [?]*

Theorem. Under some regularity conditions, the risk of $\hat{f}$ is

$$\frac{h_n^4}{4}\int\left(\mathrm{tr}(f_0''(x)\int K(u)uu^T du)\right)^2 dP(x)+\frac{1}{nh_n^d}\int K^2(u)du\int\sigma^2(x)dP(x)+o(h_n^4+(nh_n^d)^{-1}).$$

For a proof, see [4]. For points near the boundary, the bias is $Ch^2 f_0''(x)+o(h^2)$ whereas, the bias is $Chf_0'(x)+o(h)$ for kernel estimators.

In fact, [3] shows a rather remarkable result. Let R_n be the minimax risk for estimating $f_0(x_0)$ over the class of functions with bounded second derivatives in a neighborhood of x_0. Let the maximum risk r_n of the local linear estimator with optimal bandwidth satisfies

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

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

7.4 Higher-order smoothness

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

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

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

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

Again we can express

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

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

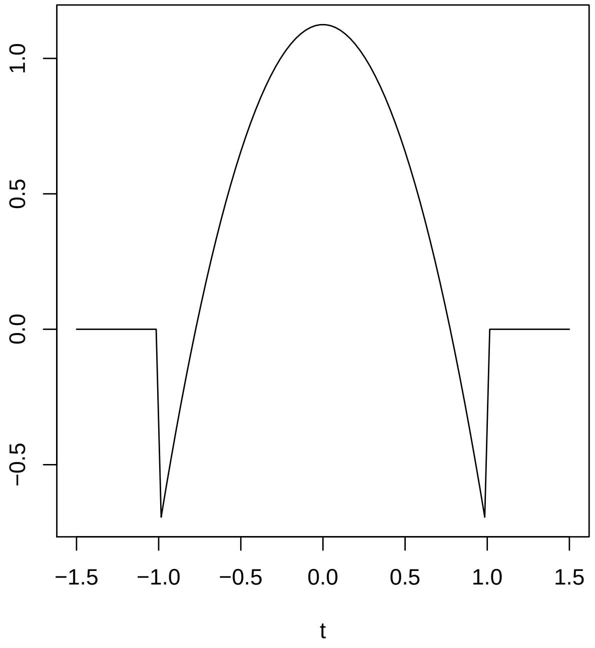

Figure 6: *A higher-order kernel function: specifically, a kernel of order 4*

Assuming that $m_0 \in H_1(\alpha, L)$ for a constant $L > 0$, a Taylor expansion shows that the local polynomial estimator $\hat{f}$ 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

$$\mathbb{E}\|\hat{f} - f_0\|_2^2 \lesssim n^{-2\alpha/(2\alpha+1)}.$$

See Chapter 1.6.1 of [9]. This matches the lower bound in (21) (when $d = 1$)

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

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

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

An example of a 4th-order kernel is $K(t) = \frac{3}{8}(3 - 5t^2)1\{|t| \le 1\}$, plotted in Figure 6. Notice that it takes negative values.

Lastly, while local polynomial regression and higher-order kernel smoothing can help "track" the derivatives of smooth functions $m_0 \in H_d(\alpha, L)$, $\alpha \ge 2$, 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 [6]

References

[1] Carl de Boor. *A Practical Guide to Splines.* Springer, 1978.

[2] A. Demmler and C. Reinsch. Oscillation matrices with spline smoothing. *Numerische Mathematik*, 24(5):375–382, 1975.

[3] Jianqing Fan. Local linear regression smoothers and their minimax efficiencies. *The Annals of Statistics*, pages 196–216, 1993.

[4] Jianqing Fan and Irene Gijbels. *Local polynomial modelling and its applications: monographs on statistics and applied probability 66*, volume 66. CRC Press, 1996.

[5] Peter Green and Bernard Silverman. *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach*. Chapman & Hall/CRC Press, 1994.

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

[7] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning*. Springer Series in Statistics. Springer, New York, second edition, 2009. Data mining, inference, and prediction.

[8] Michael Nussbaum. Spline smoothing in regression models and asymptotic efficiency in l_2. *Annals of Statistics*, 13(3):984–997, 1985.

[9] Alexandre Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2009.

[10] Sara van de Geer. *Empirical Processes in M-Estimation*. Cambdrige University Press, 2000.