Nonparametric Regression using Splines

Table of contents

1 Motivation and Overview

Consider the problem of approximating an unknown function f^* defined on $[a, b]$ from its values at n points

$$y_i = f^*(x_i) \text{ for } i = 1, .., n$$

This is a problem of interpolation. We may reduce it to a finite dimensional problem by choosing a finite basis for our search space. The optimum choice of basis will then depend upon the priori information we have about f^*.

In the absence of further information, suppose we start by approximating using polynomials. The Stone-Weierstrass theorem[5] guarantees the existence of a sequence of polynomials π_n which converge uniformly to r^*, *i.e.,*

$$\lim_{n\to\infty} \left\{ \sup_{x\in[a,b]} |f^*(x) - \pi_n(x)| \right\} = 0$$

We could define a sequence of interpolating polynomials whose order increases with the number of data points n and hope that the approximation error (as measured by the uniform norm) diminishes as n increases. However, since we do not know all the data points, the approximation error can increase without bound using this procedure[6].

We may mitigate this phenomenon by a choice of interpolating points (for instance, discard all the points except those close to roots of Chebyshev polynomials of first kind). The approximation error then will decreases to zero as n increases, but it is still a function of $(b-a)^n$. This suggests we can improve our approximation by breaking up the domain into a number of smaller bins and fit polynomials in these bins. We study the rate at which approximation error decreases with binwidth and the effect of function smoothness for a piecewise polynomial approximation in the next section.

Although we can adapt to functions of higher smoothness (a notion we will soon make more precise) by choosing polynomials of a suitably higher degree, the piecewise polynomials themselves are not smooth (they need not even be continuous). This adaptation only serves to lower the rate of decrease of the approximation error. We will look at splines as a possible solution to this problem.

If the samples we observe are noisy, merely interpolating through all the points using a spline may overfit the data. In the piecewise polynomial approximation, the bin width can be used as a controllable parameter; we would use larger bins if we have fewer data points, and smaller binwidths as we trust our data more. Similarly, we have the notion of a smoothing spline which has a controllable parameter λ that allows us to weigh goodness of fit and a smoothness measure as we please. We can then use the framework of complexity regularization to pick an optimal λ so that the average error decays at the fastest rate.

2 Smoothness and Approximation Theory

In the previous section, we saw that for function estimation, we were interested in studying how the approximation error decreases as we search in function spaces $\mathcal{F}_\lambda$ of increasing complexity λ.

Let $f^*\colon [a,b] \to \mathbb{R}$ be some function we wish to approximate and let $\{\mathcal{F}_\lambda\}$ be a parameterized family of function spaces whose complexity increases with λ. We assign a cost $A(f, f^*)$ to quantify the error in approximating f^* with f and then ask how the cost of the best fit in $\mathcal{F}_\lambda$ *viz.*

$$\inf_{f \in \mathcal{F}_\lambda} A(f, f^*)$$

decreases with increasing λ.

If the examples we use to learn f^* from are noisy, an optimal scheme would pick λ depending upon how much we trust our examples. We will see how to do this for a regression problem setup in the next section and for the moment, ignore the effect of noise and the associated estimation errors.

If we know that the function f^* is "smoother", we can choose an appropriately better family of function spaces such that the approximation error now decreases at a faster rate with increasing λ. Let us turn our attention to one way of characterizing the smoothness of functions.

2.1 Smoothness Spaces

Let C^k, $k \in \mathbb{N}$ be the space of real valued functions that have k continuous derivatives and supported on $[0,1] \subset \mathbb{R}$. Then,

$$C^0 \supset C^1 \supset C^2 \supset ... \supset C^\infty$$

are a nested sequence of function spaces where C^0 is the space of all continuous functions and C^∞ contains all infinitely differentiable functions (for instance, polynomials of any degree, bandlimited functions are in C^∞) and are called smooth functions or entire functions.

Suppose our goal is to approximate $f^* \in C^k$. Let $\mathcal{F}_m$ be the family of piecewise polynomials such with pieces at intervals $[\frac{l}{m}, \frac{l+1}{m})$ for $l = 0, ..., m-1$. Let us define the error in approximating f^* by f as

$$A(f, f^*) = \|f - f^*\|_\infty = \sup_{x \in [0,1]} |f(x) - f^*(x)|$$

We expect the approximation error to decrease with increasing m.

The following theorem tells us how to pick the degree of the polynomial pieces if $f^* \in C^k$.

Theorem 2.1. *Suppose $f^* \in C^k$ and $\mathcal{F}_m$ (defined above) has polynomial pieces of degree $d \leqslant k-1$. Then,*

$$\inf_{f \in \mathcal{F}_m} A(f, f^*) < \frac{M}{m^{1+d}}$$

Proof. *Since $d \leqslant k-1$, $f^{*(d+1)}$ exists and is also continuous. Let*

$$f_l(x) := \sum_{j=0}^{d} \frac{f^{*(j)}(\frac{l}{m})}{j!} \left(x - \frac{l}{m} \right)^j \text{ for } l = 0, ..., m-1$$

Then by Taylor's theorem[5], there exists $x_l \in \left[\frac{l}{m}, \frac{l+1}{m}\right)$ *such that*

$$f^*(x) = f_l(x) + \frac{f^{*(d+1)}(x_l)}{(d+1)!} m^{-k} \text{ for } x \in \left[\frac{l}{m}, \frac{l+1}{m}\right)$$

Now, define

$$\hat{f}(x) := \sum_{l=0}^{m-1} f_l(x) \cdot 1_{\left[\frac{l}{m}, \frac{l+1}{m}\right)}(x)$$

so that $\hat{f} \in \mathcal{F}_m$. *Since* $f^{*(d+1)}$ *is continuous and* $[0,1]$ *is compact,*

$$M := 1 + \sup_{x \in [0,1]} f^{*(d+1)}(x) < \infty$$

Then we have

$$A(f^*, \hat{f}) = \| f - \hat{f} \|_\infty = \sup_{x \in [0,1]} |\hat{f}(x) - f^*(x)| < \frac{M}{m^{1+d}}$$

Thus

$$\inf_{f \in \mathcal{F}_m} A(f, f^*) \leqslant A(\hat{f}, f^*) < \frac{M}{m^{1+d}}$$

□

From the previous theorem it is clear that the approximation error for a C^k-smooth function can decrease at a rate $O(m^{-k})$ in $\mathcal{F}_m$ by increasing the degree of the polynomial pieces to $d = k - 1$. One can show that no higher rate can be achieved by using polynomials of a higher degree. This shows how we would need to adapt the degree of the polynomial pieces according to function smoothness. Notice how the approximation error decreases with increasing smoothness (k) and increasing complexity ($\lambda \equiv m$).

In a parametric setting, we may consider a function smooth if it depends upon fewer parameters. For instance, polynomials of lower degree may be considered smoother and signals of lower bandwidth may be considered smoother. This is relevant when we study the estimation error in the presence of noisy samples. Note that for approximation, the notion of smoothness class of the function only quantifies how quickly we can decrease the approximation error as a $\lambda \to \infty$. In fact, smoother functions need to be approximated by more complex families of functions. We saw that the higher we scale in the smoothness class, the greater the degree of the polynomials we need to approximate it. It is important not to conflate the two notions.

3 Complexity Regularization

3.1 Regression and Bias-Variance Tradeoff

Suppose we have n examples $D_n := \{X_i, Y_i\}_{i=1}^n$ drawn *i.i.d* from the joint distribution of (X, Y) which are related as

$$Y = r^*(X) + \varepsilon$$

where ε has zero mean, a variance of σ^2 and is independent of Xwhich has density f. We wish to learn the regression function r^* by looking at D_n. We seek a function $\hat{r}_n$ that minimizes the mean square error. (Note that $\hat{r}_n$ is a random function since it depends upon the randomness in the data set).

For any candidate function r, with $\bar{r}(x) := \boldsymbol{E}r(x)$, the mean square error at x is

$$\begin{aligned}\boldsymbol{E}(r^*(x)-r(x))^2 &= \boldsymbol{E}\Big[(r^*(x)-\bar{r}(x))^2+(r(x)-\bar{r}(x))^2+2(r^*(x)-\bar{r}(x))(r(x)-\bar{r}(x))\Big]\\ &= (r^*(x)-\bar{r}(x))^2+\boldsymbol{E}(r(x)-\bar{r}(x))^2+2(r^*(x)-\bar{r}(x))(\boldsymbol{E}r(x)-\bar{r}(x))\\ &= [\boldsymbol{B}r(x)]^2+\boldsymbol{V}r(x)\end{aligned}$$

where $B(r)$ is the bias defined as

$$\boldsymbol{B}r(x) := \bar{r}(x) - r^*(x)$$

and $V(r)$ is the variance defined as

$$\boldsymbol{V}r(x) := \boldsymbol{E}(r(x)-\bar{r}(x))^2$$

We restrict our search for $\hat{r}_n$ in the parameterized family of function spaces $\{\mathcal{F}_\lambda\}$. Let $\mathcal{F} = \bigcup_\lambda \mathcal{F}_\lambda$ so that

$$\begin{aligned}\hat{r}_n &= \arg\inf_{r\in\mathcal{F}} \boldsymbol{E}(r^*(X)-r(X))^2\\ &= \arg\inf_{r\in\mathcal{F}} \int \Big([\boldsymbol{B}r(x)]^2+\boldsymbol{V}r(x)\Big)f(x)dx\\ &= \arg\inf_{r\in\mathcal{F}_{\lambda_n}} \boldsymbol{E}(\boldsymbol{B}r(X))^2+\Big(\boldsymbol{E}(r(X)-\bar{r}(X))^2\Big)\end{aligned}$$

where λ_n is chosen to minimize the argument above. Note that

$$\begin{aligned}\boldsymbol{E}(\boldsymbol{B}\hat{r}_n(X))^2 &= \int \boldsymbol{E}(r^*(x)-\boldsymbol{E}\hat{r}_n(x))^2 f(x)dx\\ &= \int \boldsymbol{E}(r^*(x)-r_\lambda(x))^2 f(x)dx)\end{aligned}$$

assuming that $\hat{r}_n(x)$ is chosen so that

$$\begin{aligned}\boldsymbol{E}\hat{r}_n = r_{\lambda_n} &= \arg\inf_{r\in\mathcal{F}_{\lambda_n}} \int \boldsymbol{E}(r^*(x)-r_\lambda(x))^2 f(x)dx)\\ &= \arg\inf_{r\in\mathcal{F}_{\lambda_n}} A_e(r,r^*)\end{aligned}$$

i.e., for each choice of λ_n, $\hat{r}_n$ on an average minimizes the approximation error

$$A_e(r,r^*) = \inf_{r\in\mathcal{F}_\lambda} \int (r(x)-r^*(x))^2 f(x)dx$$

in $\mathcal{F}_{\lambda_n}$.

With reasonable assumptions on $\hat{r}_n$, we see that the bias is simply the approximation error of the optimal estimator and hence decreases with increasing complexity λ_n. However, the variance of $\hat{r}_n$ would increase with complexity λ_n. This suggests the choice of picking λ_n so as to balance both the errors. For a fixed n, picking a larger than optimum λ will yield an estimator that overfits the data and a smaller than optimum would yield an oversmoothed estimate. Next, we recapitulate the results of our previous discussion about regression using piecewise polynomials and kernels.

3.2 Learning Regression using Piecewise Polynomials and Kernels

Note that the regression function is given by

$$r^*(x) \;=\; \boldsymbol{E}(Y\,|X=x)$$

which suggests averaging values near x if we assume that $r^*(x)$ is smooth enough. Depending upon the smoothness of r^* we may approximate by fitting polynomials of higher degree.

This intuition implicitly assumes that f (density of X) is uniform. If f takes really low values near x, then there may be too few points for the regression estimates to be correct. We could get a more uniform approximation by weighing it with the estimated density in the window appropriately. So, we observe that

$$\begin{aligned} r^*(x) &= \boldsymbol{E}(Y\,|X=x) \\ &= \int y f(y|x)\,dx \\ &= \int \frac{y f(x,y)}{f(x)} dx \end{aligned}$$

which suggests the an estimator of the form

$$\hat{r}_n(x) = \frac{\frac{1}{n}\sum_{i=1}^{n} \frac{1}{h} K\Big(\frac{x-X_i}{h}\Big) Y_i}{\frac{1}{n}\sum_{i=1}^{n} \frac{1}{h} K\Big(\frac{x-X_i}{h}\Big)}$$

for a constant fit near x. h is the bandwidth of the kernel. This is a weighted sum of responses Y_i near x. The Kernel serves to smooth the estimate by weighing points near x more than points further away. We can show (as we did last time) that this Kernel Regression estimator is consistent and has bias and variance given by

$$\text{Bias}^2 = \boldsymbol{E}\big(\boldsymbol{B}\hat{r}_n(X)\big)^2 \in O(h^4)$$

and

$$\text{Variance} = O\Big(\frac{1}{nh}\Big)$$

with mild assumptions on the kernel (integrates to unity, symmetric) and the assumption that $r^* \in C^2$. As expected the bias of the estimator (function of the approximation error) decreases with increasing complexity $\frac{1}{h}$ and the variance (function of the estimation error) increases with complexity $\frac{1}{h}$ but decreases with the number of data points n. By trading off these two errors, we saw that a choice of $h \in O\Big(n^{-1/5}\Big)$ gives the optimal rate of $O\Big(n^{-4/5}\Big)$ for the mean square error.

Although we do change the degree of the polynomial and choose suitably higher order kernels to adapt to regression functions of higher smoothness, this has an effect only on the rate at which the mean square error decays and not the smoothness of $\hat{r}_n$ itself. For example, if K were a boxcar kernel, $\hat{r}_n$ is only piecewise polynomial which need not be even continuous. So, the continuity and smoothness of $\hat{r}_n$ only comes from the choice of the kernel.

In the following section, we look at an alternative regression technique instead of the kernel regression estimator, so we can easily guarantee a suitable smoothness for $\hat{r}_n$.

4 Splines

Let us ignore the effect of noise for a while and concern ourselves solely with approximation errors again. We will soon return to learning from noisy examples. We want to interpolate through the data points while guaranteeing a particular smoothness for the interpolant. A natural way of extending piecewise polynomial fits is to splice the pieces together such that the higher order derivatives match up at the interpolating points. This is called a spline.

Definition 4.1. *A piecewise polynomial $s_n^m(x)$ of order $2m$ (or degree $2m-1$) is called a **spline** with **knots** at $x_1, ..., x_n$ if it is a polynomial of degree $2m-1$ in each of the intervals $[x_1, x_2], ...,$ $[x_{n-1}, x_n]$ and has smoothness of C^{2m-2}.*

Note that the loss of continuity only occurs at the knots. For $s \in C^{2m-2}$, it must be $2m-2$ times differentiable everywhere and each derivative must be continuous. This is already satisfied between the knots. At the knots, we must therefore have

$$s^{(j)}(x_k-) = s^{(j)}(x_k+) \text{ for each } k = 2, ..., n-1 \text{ and each } j = 0, ..., 2m-2$$

Definition 4.2. *A spline $s_n^m(x)$ on $[a, b]$ of order $2m$ and n knots at $a < x_1, ..., x_n < b$ is said to be a **natural spline** if it is an order $2m$ spline with knots at $x_1, ..., x_n$ with the further constraint that the polynomial pieces at the two end intervals $[a, x_1]$ and $[x_n, b]$ be of degree $m-1$*

Note that the constraint that the spline be a polynomial of a degree $m-1$ (or order m) can also be written as

$$s^{(j)}(a) = s^{(j)}(b) = 0 \text{ for } j \geqslant m$$

Now, given n data points $\{(x_1, y_1), ..., (x_n, y_n)\}$, Carl de Boor[2] showed the existence of a unique natural cubic spline on $[a, b]$ with knots at $a < x_1, ..., x_n < b$ that interpolates through these points for any $n > m$. This is called an *interpolating spline*. We reproduce here a simple counting argument due to Grace Wahba[9]:

To determine a natural spline of order $2m$ and n knots, we need $2m$ coefficients for a $2m-1$ degreee polynomial in each of the $n-1$ intervals $[x_1, x_2], ..., [x_{n-1}, x_n]$ and m coefficients each at the end intervals. This gives us $2m(n-1)+m+m = 2mn$ coefficents that we need to determine. Now, we have at each of the n knots, $2m-1$ linear equations because we want the $2m-2$ derivatives of the polynomials to the left and right of each knot to match up so that the spline is in C^{2m-2} and we have a continuity requirement so that the polynomials meet at the knot. We also have n more linear equations since we want the spline to interpolate through n points. So, we have $(2m-1)n + n = 2mn$ conditions that would allow us to determine a unique natural cubic spline. This means we have $2m$ extra degrees of freedom to satisfy additional constraints if we do not need a natural spline.

We looked at splines merely as a patchwork to rectify a deficiency of piecewise polynomial fits. However the following variational properties that splines satisfy that may encourage the choice of splines for interpolation based on priors on the unknown function. The following result due to Schoenberg[7] establishes a minimal property of a natural spline.

4.1 A Minimal Property

Of all the C^{2m-2} functions on $[a, b]$ that also interpolate through the n points $\{(x_i, y_i)\}_{i=1}^n$, the natural interpolating spline s of order $2m$ minimizes

$$\int_a^b \Big[f^{(m)}(x) \Big]^2 dx$$

The integrated m^{th} derivative may be construed as a smoothness measure. We do an adaptation of a proof due to John Holladay[3] (who showed this for $m = 2$)

Put $(x_0, y_0) := (a, 0)$ and $(x_{n+1}, y_{n+1}) = (b, 0)$. Let s be the interpolating spline and let $g \in C^{2m-2}$ be another function that also interpolates through all the data points.

$$g(x_k) = s(x_k) = y_k \text{ for } k = 0, ..., n+1 \tag{4.1}$$

$$g^{(j)}(x_k -) = g^{(j)}(x_k +) \text{ for } k = 1, ..., n; j = 1, ...2m-2 \tag{4.2}$$

$$s^{(j)}(a) = s^{(j)}(b) = 0; j \geqslant m \tag{4.3}$$

The last equality is the natural spline condition which forces it to be a polynomial of order m and degree $m-1$. Now, using integration by parts for $k=0,...,m-3$, we have

$$\begin{aligned}\int_a^b s^{(m+k)}(x)\Big(g^{(m-k)}(x)-s^{(m-k)}(x)\Big)dx &= \Big[s^{(m+k)}(x)\Big(g^{(m-k-1)}(x)-s^{(m-k-1)}(x)\Big)\Big]_a^b - \\ &\quad \int_a^b s^{(m+k+1)}(x)\Big(g^{(m-k-1)}(x)-s^{(m-k-1)}(x)\Big)dx \\ &= \int_a^b s^{(m+k+1)}(x)\Big(s^{(m-k-1)}(x)-g^{(m-k-1)}(x)\Big)dx \\ &\quad \text{(using 4.3)} \qquad (4.4)\end{aligned}$$

By repeated application of 4.4, we

$$\begin{aligned}\int_a^b s^{(m)}(x)\Big(g^{(m)}(x)-s^{(m)}(x)\Big)dx &= (-1)^{m-3}\int_a^b s^{(2m-2)}(x)\Big(g^{(2)}(x)-s^{(2)}(x)\Big)dx \\ &= (-1)^{m-3}\Big[s^{(2m-2)}(x)(g'(x)-s'(x))\Big]_a^b \\ &\quad -(-1)^{m-3}\sum_{i=0}^{n}\int_{x_i}^{x_{i+1}} s^{2m-1}(x)(g'(x)-s'(x))dx \\ &= (-1)^m\sum_{i=0}^{n} c_i\int_{x_i}^{x_{i+1}}(g'(x)-s'(x))dx \\ &= (-1)^m\sum_{i=0}^{n} c_i[g(x)-s(x)]_{x_i}^{x_{i+1}} \\ &= 0 \qquad (4.5)\end{aligned}$$

where we use the fact that $(2m-1)^{\text{th}}$ derivative of s exists between the knots and is a constant, say c_i in $[x_i, x_{i+1}]$.

Finally,

$$\begin{aligned}\int_a^b \Big[g^{(m)}(x)\Big]^2dx &= \int \Big(g^{(m)}(x)-s^{(m)}(x)\Big)^2dx + \int \Big[s^{(m)}(x)\Big]^2dx \\ &\quad \text{(cross term vanishes by 4.5)} \\ &\geqslant \int \Big[s^{(m)}(x)\Big]^2dx \text{ (note that the equality holds only if } g=s!) \qquad (4.6)\end{aligned}$$

This has a nice interpretation when $m=2$, which corresponds to a order 4 (degree 3) spline called the natural cubic spline. The natural cubic spline minimizes

$$\int [f''(x)]^2dx \cong \int \left[\frac{f''(x)}{(1+f'(x))^{3/2}}\right]^2 dx = \text{Curvature}(f)$$

I have no idea why that approximation is reasonable! But if it is reasonable, we may think of natural cubic spline as the function of minimum curvature that runs through a given set of points.

4.2 Noisy Samples and Regularization

When we have noisy samples, it may not be a good idea to interpolate through all the points. We may wish to have a parameter λ (like the bandwidth in the kernel regression problem, or the window size in the piecewise polynomial interpolation problem) that we can vary so that as we trust our data more, we can penalize the m^{th} derivative (smoothness constraint) less. This can be done by a classic regularization scheme as follows.

Among all functions in C^{2m-2}, find an f that minimizes

$$J(f) = E(f) + \frac{1}{\lambda} P(f)$$

where

$$E(f) = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

encourages a better fit to the data and

$$P(f) = \int_a^b \left[f^{(m)}(x) \right]^2 dx$$

is the penalty term that punishes a large m^{th} derivative.

The parameter λ controls the trade-off between goodness of fit to the data and the smoothness as measured by the m^{th} derivative. If $\lambda \to \infty$, any function that interpolates through the data minimizes J and as $\lambda \to 0$, the m^{th} derivative is penalized so heavily that a $m-1$ degree polynomial fit through the data will minimize J. For very large values of λ, it is desirable for f to almost interpolate through the data while ensuring that the penalty term is not too large compared to functions that similarly interpolate, which seems to favour splines. Schoenberg[8] proved that this is indeed the case. For a proof using Calculus of Variations, see Reinsch[4].

Here is a simple argument to show that it is sufficient to search within splines at knots at $\{x_i\}_{i=1}^n$. Suppose $\hat{f}$ is not a spline. Now construct a spline s with knots at $\{x_i\}_{i=1}^n$ which interpolates at these knots to the corresponding $\hat{f}$ values. Then by 4.6,

$$\frac{1}{\lambda} \int_a^b \left[\hat{f}^{(m)}(x) \right]^2 dx > \frac{1}{\lambda} \int_a^b \left[s^{(m)}(x) \right]^2 dx$$

Also

$$\sum_{i=1}^{n} (y_i - \hat{f}(x_i))^2 = \sum_{i=1}^{n} (y_i - s(x_i))^2$$

Thus

$$J(s) < J(\hat{f})$$

Hence, for every $f \in C^{2m-2}$ that is not a spline, there exists a spline with a strictly smaller penalized fitness cost compared to $J(f)$. It is sufficient to search for a minimum among splines. Further more, the minimum, if it exists, is unique since J is convex:

Proposition 4.1. *J is convex*

Proof. *Let $\boldsymbol{L}$ be any linear functional and let $c \in \mathbb{R}$. Define*

$$Q(f) = (\boldsymbol{L}f + c)^2$$

Then for any $0 < \alpha < 1$,

$$\begin{aligned} Q((1-\alpha)f_1 + \alpha f_2) &= [(1-\alpha)(\boldsymbol{L}f_1 + c) + \alpha(\boldsymbol{L}f_2 + c)]^2 \\ &= (1-\alpha)^2(\boldsymbol{L}f_1 + c)^2 + \alpha^2(\boldsymbol{L}f_2 + c)^2 + 2\alpha(1-\alpha)(\boldsymbol{L}f_1 + c)(\boldsymbol{L}f_2 + c) \\ &= (1-\alpha)(\boldsymbol{L}f_1 + c)^2 + \alpha(\boldsymbol{L}f_2 + c)^2 \\ &\quad - \alpha(1-\alpha)\big[(\boldsymbol{L}f_1 + c)^2 + (\boldsymbol{L}f_2 + c)^2 - 2(\boldsymbol{L}f_1 + c)(\boldsymbol{L}f_2 + c)\big] \\ &= (1-\alpha)(\boldsymbol{L}f_1 + c)^2 + \alpha(\boldsymbol{L}f_2 + c)^2 - \alpha(1-\alpha)[\boldsymbol{L}(f_1 - f_2)]^2 \\ &\leqslant (1-\alpha)Q(f_1) + \alpha Q(f_2) \end{aligned}$$

Thus Q is convex. Now, point evaluation is a linear functional and hence $E(f)$ is convex. Similarly the square of the m^{th} derivative is convex and so also the intergral of it (Linear transformations of convex functions are convex). Since J is a sum of convex functions, J is also convex. □

Thus, we see that splines solve a regularized least squares problem with a smoothing paramater λ determining a tradeoff between smoothness as measured by the m^{th} derivative and fidelity to data. We know that for splines that interpolate through the data, we can determine the coefficients of the polynomial pieces by solving a set of linear equations at the knots. However, the splines that minimize the penalized least squares do not interpolate through the data points: We pick the the knots at the abscissa of the datapoints, but the ordinates can be arbitrary. These splines are called *smoothing splines*.

In the next section, we see that splines have a finite basis. So, the search for the optimal spline becomes a finite dimensional matrix problem. This will allow us to determine the smoothing spline for a given λ

4.3 Determining Smoothing Spline Coefficients

Consider the space of all natural splines that have n knots at $x_1, ..., x_n$ and order $2m$. This is a linear space of dimension $n + 2m$. Recall that we have $2m$ independent coefficients to pick for the 2 polynomials in the end intervals. After picking them, n more coefficients which would correspond to points we want the spline to interpolate through, will uniquely determine the spline. Then it is easy to see that $1, x, ..., x^{2m-1}, (x - x_1)_+^{2m-1}, ..., (x - x_n)_+^{2m-1}$ is a sequence of $n + 2m$ independent splines of order $2m$ and knots at x_1, ..., x_n and must hence form a basis. This is called the **truncated spline basis**.

Another basis called the **B-spline** basis proposed by Schoenberg consists of splines of the minimal support of a given degree, passing through a given set of knots. Carl de Boor showed that the B-spline basis can be generated easily using recursion. For the case of uniformly spaced knots they are just shifted versions of a primary B-spline.

Let $B_1, ..., B_{n+2m}$ be a set of splines of order $2m$ with knots at $x_1, ..., x_n$. Suppose we wish to find a smoothing spline $\hat{s}(t)$ of order $2m$ and knots at $x_1, ..., x_n$ that minimizes

$$J(f) = \sum_{i=1}^{n} (y_i - f(x_i))^2 + \frac{1}{\lambda} \int_a^b \left[f^{(m)}(x) \right]^2 dx$$

We write f as

$$f(t) = \sum_{i=1}^{n+2m} \beta_i B_i(t)$$

If we let B be the matrix whose i^{th} column is $[B_i(x_1), ..., B_i(x_n)]^T$, and define

$$\beta := [\beta_1, ..., \beta_{n+2m}]^T$$

$$Y := [y_1, ..., y_n]^T$$

$$E(f) = \sum_{i=1}^{n} (y_i - f(x_i))^2 = (Y - B\beta)^T (Y - B\beta)(Y - B\beta)^T (Y - B\beta)$$

Also,

$$\begin{aligned} P(f) &= \int_a^b \left[f^{(m)}(x) \right]^2 dx \\ &= \int_a^b \left[\sum_{i=1}^{n+2m} \beta_i B_i^{(m)}(t) \right]^2 dt \\ &= \int_a^b \left[\sum_{i=1}^{n+2m} \sum_{j=1}^{n+2m} \beta_i \beta_j B_i^{(m)}(t) B_j^{(m)}(t) \right] dt \\ &= \sum_{i=1}^{n+2m} \sum_{j=1}^{n+2m} \beta_i \beta_j \int_a^b B_i^{(m)}(t) B_j^{(m)}(t) dt \end{aligned}$$

If we define a (symmetric) matrix Ω with $(i, j)^{\text{th}}$ entry

$$\Omega_{ij} = \int_a^b B_i^{(m)}(t) B_j^{(m)}(t) dt \quad (= \Omega_{ji}!)$$

we see that Ω is positive definite since for any $\alpha = [\alpha_1, ..., \alpha_{n+2m}]^T \neq \mathbf{0}$,

$$\alpha^T \Omega \alpha = \int_a^b \left[\sum_{i=1}^{n+2m} \alpha_i B_i^{(m)}(t) \right]^2 dt > 0$$

Now $P(f)$ reduces to a Quadratic form given by

$$P(f) = \beta^T \Omega \beta$$

Thus, to determine $\hat{\beta}$, we just have to determine β that minimizes

$$\tilde{J}(\beta) = (Y - B\beta)^T(Y - B\beta) + \frac{1}{\lambda}\beta^T\Omega\beta$$

This is a well-posed matrix regularization problem. Since we already showed that J is a convex function of f, $\tilde{J}$ can be shown to be a convex function of β by simply setting f_1 and f_2 in the previous proof to be linear combinations of basis functions with coefficients β_1 and β_2 in the previous proof. Therefore if there exists $\hat{\beta}$ with

$$\nabla\tilde{J}(\hat{\beta}) = 0$$

then $\hat{\beta}$ corresponds to the unique minimizer. Here, by vector differentiation, we obtain

$$\nabla\tilde{J}(\beta) = -2(Y - B\beta)^T B + \frac{2}{\lambda}\beta^T\Omega$$

Thus

$$\begin{aligned}\nabla\tilde{J}(\hat{\beta})^T = 0 \;&\Rightarrow\; B^T(Y - B\hat{\beta}) = \frac{1}{\lambda}\Omega\hat{\beta} \\ &\Rightarrow\; \hat{\beta} = \left(B^TB + \frac{1}{\lambda}\Omega\right)^{-1} B^TY\end{aligned}$$

The matrix $\left(B^TB + \frac{1}{\lambda}\Omega\right)$ is invertible since it has positive eigen values being the sum of non-negative definite B^TB and a positive definite Ω/λ. So, we have a scheme for determining the coefficients of the smoothing spline.

4.4 Choice of λ

As in other nonparametric methods we have seen, we can pick our smoothing parameter so as to balance overfitting and oversmoothing. The bias and variance of spline estimator can be hard to compute. Peter Craven and Grace Wahba[1] suggests the use of leave-one-out validation and generalized cross validation schemes to pick a λ.

Bibliography

[1] Peter Craven and Grace Wahba, *Smoothing noisy data with spline functions*, Numerische Mathematik **31** (1979), 377–403.

[2] Carl de Boor, *Best approximation properties of spline functions of odd degree*, J Math. Mech. **12** (1963), no. 5, 747–749.

[3] John C. Holladay, *A smoothest curve approximation*, Mathematical Tables and Other Aids to Computation **11** (1957), no. 60, 233–243.

[4] Christian Reinsch, *Smoothing by spline functions*, Numerische Mathematik **16** (1971), 451–454.

[5] Walter Rudin, *Principles of mathematical analysis*, third ed., McGraw-Hill, New York, 1976.

[6] C Runge, *Über empirische funktionen und die interpolation zwischen äquidistanten ordinaten"*, Zeitschrift Mathematische Physik **46** (1901), 224–243.

[7] Isaac Schoenberg, *On interpolation by spline functions and its minimal properties*, Proceedings of the Conference held in the Mathematical Research Institute at Oberwolfach, 1964, p. 109.

[8] Isaac Schoenberg, *Spline functions and the problem of graduation*, Annals of Mathematicals **52** (1964), 947–950.

[9] Grace Wahba, *Spline models for observational data*, Society for Industrial and Applied Mathematics, Philadelphia, 1990.