Notes on Statistical and Machine Learning

Basis Expansions and Regularization

Chapter: *8* **Prepared by:** *Chenxi Zhou*

This note is prepared based on *Chapter 5, Basis Expansions and Regularization* in Hastie, Tibshirani, and Friedman (2009). In this chapter, we extend the linear models into nonlinear ones and include the transformations of input variables.

I. Introduction

1. **Motivation:** All models discussed in earlier chapters are linear ones. Linear model is a convenient approximation to the underlying true function f (e.g., the regression function $\mathbb{E}[Y \mid X]$ or the Bayes rule classifier $\arg\max_y \mathbb{P}(Y = y \mid X)$). Benefits of a linear model include the following:

   (a) linear models are easy to interpret;

   (b) linear models are the first-order Taylor approximation to f.

   In practice, it is extremely *unlikely* that the underlying true function f is *not* linear.

2. **General Idea of Nonlinear Additive Models:** Let $h_m : \mathbb{R}^p \to \mathbb{R}$ denote the m-th transformation of X, for $m = 1, \cdots, M$, and fit the model

$$f(\mathbf{x}) = \sum_{m=1}^{M} \beta_m h_m(\mathbf{x}).$$

   Remark. Note that f may *not* be linear in $\mathbf{x}$ but is linear in $h_1(\mathbf{x}), \cdots, h_M(\mathbf{x})$.

3. **Examples of h_m:**

   - $h_m(\mathbf{x}) = x_m$ for $m = 1, \cdots, p$, the original input variables;
   - $h_m(\mathbf{x}) = x_j^k$ for $k \in \mathbb{N}$, the polynomial terms;
   - $h_m(\mathbf{x}) = x_j \cdot x_k$, the interactions;
   - $h_m(\mathbf{x}) = \log(x_j)$ or $h_m(\mathbf{x}) = \sqrt{x_j}$, nonlinear transformations of $\mathbf{x}$;
   - $h_m(\mathbf{x}) = \mathbb{1}(L_m \leq x_j < U_m)$, an indicator for a region of x_j.

4. **Advantage and Disadvantage of Using Basis Expansions:**

   (a) *Advantages:* Achieve more flexibility representation;

   (b) *Disadvantages:*

- Can increase the complexity of a model;
- May lead to overfitting.

5. **Methods of Controlling the Complexity of a Model:** There are three common approaches to control the complexity of a model:

   (a) *Restriction Method:* Decide before-hand to limit the class of functions. For example, limit the number of basis functions;

   (b) *Selection Method:* Adaptively scan the set of basis functions and include only those that contribute *significantly* to the fit of the model. For example, forward and backward selection, and stagewise greedy approaches such as CART and MARS;

   (c) *Regularization Method:* Use all basis functions but restrict the coefficients. For example, ridge and lasso regression.

II. Piecewise Polynomials and Splines

1. **Setup:** Assume X is one-dimensional. Partition the range of X into several discontiguous intervals and assume there are M breaking points and, hence, $(M+1)$ disjoint intervals. We represent f by a separate polynomial in each sub-interval. Hence, the model we fit is

$$f(x) = \sum_{m=1}^{M+1} \beta_m h_m(x).$$

   In this section, we assume $M = 2$ and let ξ_1, ξ_2 be the breaking points.

2. **Piecewise Constant Function:** The basis functions are of the form

$$\begin{aligned} h_1(x) &= \mathbb{1}(x < \xi_1), \\ h_2(x) &= \mathbb{1}(\xi_1 \le x < \xi_2), \\ h_3(x) &= \mathbb{1}(\xi_2 \le x). \end{aligned}$$

   Then, the estimate of β_m on each interval is just the mean of the response in that particular interval.

3. **Piecewise Linear Function:** The basis functions are of the form

$$\begin{aligned} h_1(x) &= \mathbb{1}(x < \xi_1) \cdot (\beta_{1,0} + \beta_{1,1}x), \\ h_2(x) &= \mathbb{1}(\xi_1 \le x < \xi_2) \cdot (\beta_{2,0} + \beta_{2,1}x), \\ h_3(x) &= \mathbb{1}(\xi_2 \le X) \cdot (\beta_{3,0} + \beta_{3,1}x), \end{aligned}$$

   where we have 6 parameters to estimate in total.

4. **Continuous Piecewise Linear Function:** We use the basis functions in the piecewise linear function case and assume that f is continuous at the breaking points. This leads to

$$f(\xi_i^+) = f(\xi_i^-) \qquad \text{for all } i = 1, 2,$$

that is,

$$\beta_{i,0} + \beta_{i,1}\xi_i = \beta_{i+1,0} + \beta_{i+1,1}\xi_i, \qquad \text{for } i = 1, 2.$$

Since we have 2 restrictions (1 restriction at each breaking point) here, we have 4 parameters to estimate in total.

5. **Piecewise Linear Basis Function:** We require the basis functions of the form

$$h_1(x) = 1, \qquad h_2(x) = x, \qquad h_3(x) = (x - \xi_1)_+, \qquad h_4(x) = (x - \xi_2)_+,$$

where $(x)_+ = \max\{x, 0\}$.

6. **Cubic Spline:** We further extend the case of piecewise linear basis function above and wish to obtain *smoother* function estimation by *increasing* the order of the local polynomial. We require the continuity at the knots, and the continuous first- and second-order derivatives at the breaking points. This leads to the *cubic spline*. The basis functions with two knots at ξ_1 and ξ_2 are

$$\begin{aligned} h_1(x) &= 1, & h_2(x) &= x, & h_3(x) &= x^2, \\ h_4(x) &= x^3, & h_5(x) &= (x - \xi_1)_+^3, & h_6(x) &= (x - \xi_2)_+^3. \end{aligned}$$

In such a case, we need to estimate 6 parameters (due to the constraints), since

$$(3 \text{ regions}) \times (4 \text{ parameters per region}) - (2 \text{ knots}) \times (3 \text{ constraints per knot}) = 6.$$

7. **M-th Order Spline:** Suppose the knots are fixed at $\{\xi_i\}_{i=1}^K$ and the *M-th order spline* is a piecewise polynomial of order M with continuous derivatives up to order $(M-2)$. The basis functions are

$$\begin{aligned} h_j(x) &= x^{j-1}, & j &= 1, \cdots, M, \\ h_{M+l}(x) &= (x - \xi_l)_+^{M-1}, & l &= 1, \cdots, K, \end{aligned}$$

where we use the truncated power basis set.

Remarks.

- The cubic spline has order $M = 4$.
- It is claimed that cubic splines are the *lowest* order spline for which the knot-discontinuity is *not* visible to the human eye.
- The fixed-knot splines are also known as *regression splines*. One needs to select
  - the order of the spline,

  - the number of knots, and
  - the placement of knots.

  One approach is to parametrize a family of splines by the number of basis functions or degrees of freedom and have the knots at the observations.

- Since the space of spline functions of a particular order and knot sequence is a *vector space*, there are many equivalent bases for representation.

8. Natural Cubic Splines:

(a) *Motivation:* At the boundaries, the polynomial fits can be erratic and *extrapolation* can be dangerous. The spline models can behave bad at the boundaries. To remedy this problem, we introduce the *natural cubic spline*.

(b) *Definition: Natural cubic spline* requires that the function is *linear* beyond the boundary knots.

(c) *Basis Functions:* A natural cubic spline with K knots $\{\xi_k\}_{k=1}^K$ is represented by K basis functions

$$\begin{aligned} N_1(x) &= 1, \\ N_2(x) &= x, \\ N_{k+2}(x) &= d_k(x) - d_{K-1}(x) \qquad \text{for } k = 1, \cdots, K-2, \end{aligned} \tag{1}$$

where

$$d_k(x) = \frac{(x-\xi_k)_+^3 - (x-\xi_K)_+^3}{\xi_K - \xi_k}. \tag{2}$$

(d) *Derivation of (1):* We start from the truncated power series representation for cubic splines with K interior knots, and let

$$f(x) = \sum_{j=0}^{3} \beta_j x^j + \sum_{k=1}^{K} \theta_k (x-\xi_k)_+^3. \tag{3}$$

We show that the natural boundary conditions for natural cubic splines imply

$$\beta_2 = 0, \qquad \beta_3 = 0, \qquad \sum_{k=1}^{K} \theta_k = 0, \qquad \text{and} \qquad \sum_{k=1}^{K} \xi_k \theta_k = 0. \tag{4}$$

To start with, for all $x \in (-\infty, \xi_1)$, we have

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3.$$

To ensure f is linear over $(-\infty, \xi_1)$, we must have $\beta_2 = 0$ and $\beta_3 = 0$.
For all $x \in (\xi_K, \infty)$, we have

$$\begin{aligned} f(x) &= \beta_0 + \beta_1 x + \sum_{k=1}^{K} \theta_k (x-\xi_k)^3 \\ &= \beta_0 + \beta_1 x + \left(\sum_{k=1}^{K} \theta_k\right) x^3 - 3\left(\sum_{k=1}^{K} \theta_k \xi_k\right) x^2 + 3\left(\sum_{k=1}^{K} \theta_k \xi_k^2\right) x - \left(\sum_{k=1}^{K} \theta_k \xi_k^3\right). \end{aligned}$$

To ensure f is linear over (ξ_K, ∞), we must have

$$\sum_{k=1}^{K} \theta_k = 0 \qquad \text{and} \qquad \sum_{k=1}^{K} \theta_k \xi_k = 0.$$

Now, let

$$S_1 := \Big\{ f \;\Big|\; f \text{ is spanned by (1)} \Big\},$$
$$S_2 := \Big\{ f \;\Big|\; f \text{ is of the form (3) with (4) satisfied} \Big\}.$$

We show $S_1 = S_2$ by showing $S_1 \subseteq S_2$ and $S_2 \subseteq S_1$.
To show $S_1 \subseteq S_2$: Let $f(x) = \beta_1 + \beta_2 x + \sum_{k=1}^{K-2} \gamma_k N_k(x) \in S_1$. It is obvious that such an f can be written in the form of (3). We need to show that f satisfies (4). Note that it is obvious $\beta_2 = \beta_3 = 0$. In addition, note that

$$\begin{aligned} f(x) = \beta_1 + \beta_2 x &+ \sum_{k=1}^{K-2} \frac{\gamma_k}{\xi_K - \xi_k} (x - \xi_k)_+^3 + \sum_{k=1}^{K-2} \frac{-\gamma_k}{\xi_K - \xi_{K-1}} (x - \xi_{K-1})_+^3 \\ &+ \sum_{k=1}^{K-2} \left(\frac{-\gamma_k}{\xi_K - \xi_k} + \frac{\gamma_k}{\xi_K - \xi_{K-1}} \right) (x - \xi_K)_+^3, \end{aligned}$$

that is,

$$\begin{aligned} \theta_k &= \frac{\gamma_k}{\xi_K - \xi_k}, \qquad \text{for all } k = 1, \cdots, K-2, \\ \theta_{K-1} &= \sum_{k=1}^{K-2} \frac{-\gamma_k}{\xi_K - \xi_{K-1}}, \\ \theta_K &= \sum_{k=1}^{K-2} \left(\frac{-\gamma_k}{\xi_K - \xi_k} + \frac{\gamma_k}{\xi_K - \xi_{K-1}} \right). \end{aligned}$$

Thus,

$$\begin{aligned} \sum_{k=1}^{K} \theta_k &= \sum_{k=1}^{K-2} \frac{\gamma_k}{\xi_K - \xi_k} + \sum_{k=1}^{K-2} \frac{-\gamma_k}{\xi_K - \xi_{K-1}} + \sum_{k=1}^{K-2} \left(\frac{-\gamma_k}{\xi_K - \xi_k} + \frac{\gamma_k}{\xi_K - \xi_{K-1}} \right) = 0, \\ \sum_{k=1}^{K} \xi_k \theta_k &= \sum_{k=1}^{K-2} \frac{\xi_k \gamma_k}{\xi_K - \xi_k} + \xi_{K-1} \sum_{k=1}^{K-2} \frac{-\gamma_k}{\xi_K - \xi_{K-1}} + \xi_K \sum_{k=1}^{K-2} \left(\frac{-\gamma_k}{\xi_K - \xi_k} + \frac{\gamma_k}{\xi_K - \xi_{K-1}} \right) \\ &= \sum_{k=1}^{K-2} \frac{1}{\xi_K - \xi_k} (\xi_k \gamma_k - \xi_K \gamma_k) + \sum_{k=1}^{K-2} \frac{1}{\xi_K - \xi_{K-1}} (-\xi_{K-1} \gamma_k + \xi_K \gamma_k) \\ &= -\sum_{k=1}^{K-2} \gamma_k + \sum_{k=1}^{K-2} \gamma_k \\ &= 0. \end{aligned}$$

This shows $S_1 \subseteq S_2$.
To show $S_2 \subseteq S_1$: Conversely, let $f(x) = \beta_0 + \beta_1 x + \sum_{k=1}^{K} \theta_k (x - \xi_k)_+^3 \in S_2$ with coefficients satisfying (4). From (2), we know, for all $k = 1, \cdots, K-2$,

$$(x - \xi_k)_+^3 = \big(N_{k+2}(x) + d_{K-1}(x)\big)(\xi_K - \xi_k) + (x - \xi_K)_+^3.$$

Thus, we can write f as

$$\begin{aligned}
f(x) =& \beta_0 N_1(x) + \beta_1 N_2(x) \\
&+ \sum_{k=1}^{K-2} \theta_k \Big(\big(N_{k+2}(x) + d_{K-1}(x)\big)(\xi_K - \xi_k) + (x - \xi_K)_+^3 \Big) \\
&\quad + \theta_{K-1}(x - \xi_{K-1})_+^3 + \theta_K (x - \xi_K)_+^3 \\
=& \beta_0 N_1(x) + \beta_1 N_2(x) + \sum_{k=1}^{K-2} \theta_k (\xi_K - \xi_k) N_{k+2}(x) \\
&+ \sum_{k=1}^{K-2} \theta_k \Big((\xi_K - \xi_k) d_{K-1}(x) + (x - \xi_K)_+^3 \Big) \\
&\quad + \theta_{K-1}(x - \xi_{K-1})_+^3 + \theta_K (x - \xi_K)_+^3.
\end{aligned}$$

To complete the proof, it is sufficient to show, for all x,

$$\sum_{k=1}^{K-2} \theta_k \Big((\xi_K - \xi_k) d_{K-1}(x) + (x - \xi_K)_+^3 \Big) + \theta_{K-1}(x - \xi_{K-1})_+^3 + \theta_K (x - \xi_K)_+^3 = 0. \tag{5}$$

In (5), the coefficient associated with $(x - \xi_{K-1})_+^3$ is

$$\begin{aligned}
&\sum_{k=1}^{K-2} \theta_k \frac{\xi_K - \xi_k}{\xi_K - \xi_{K-1}} + \theta_{K-1} \\
=& \frac{1}{\xi_K - \xi_{K-1}} \sum_{k=1}^{K-2} \Big(\theta_k \xi_K - \theta_k \xi_k + \theta_{K-1} \xi_K - \theta_{K-1} \xi_{K-1} \Big) \\
=& \frac{1}{\xi_K - \xi_{K-1}} \left[-(\theta_{K-1} + \theta_K) \xi_K - \left(\sum_{k=1}^{K-2} \theta_k \xi_k \right) + \theta_{K-1} \xi_K - \theta_{K-1} \xi_{K-1} \right] \\
=& \frac{1}{\xi_K - \xi_{K-1}} \sum_{k=1}^{K} \theta_k \xi_k \\
=& 0.
\end{aligned}$$

In addition, the coefficient associated with $(x-\xi_K)_+^3$ is

$$
\begin{aligned}
&-\sum_{k=1}^{K-2}\theta_k\frac{\xi_K-\xi_k}{\xi_K-\xi_{K-1}}+\sum_{k=1}^{K-2}\theta_k+\theta_K\\
=&\frac{1}{\xi_K-\xi_{K-1}}\sum_{k=1}^{K-2}\Big(-\theta_k(\xi_K-\xi_k)\Big)+\sum_{k=1}^{K-2}\theta_k+\theta_K\\
=&\frac{1}{\xi_K-\xi_{K-1}}\left[-\xi_K\sum_{k=1}^{K-2}\theta_k+\sum_{k=1}^{K-2}\theta_k\xi_k\right]+\sum_{k=1}^{K-2}\theta_k+\theta_K\\
=&\frac{1}{\xi_K-\xi_{K-1}}\Big[-\xi_K(-\theta_{K-1}-\theta_K)-(\theta_{K-1}\xi_{K-1}-\theta_K\xi_K)\Big]+\sum_{k=1}^{K-2}\theta_k+\theta_K\\
=&\frac{1}{\xi_K-\xi_{K-1}}\Big[\xi_K\theta_{K-1}+\xi_{K-1}\theta_{K-1}\Big]+\sum_{k=1}^{K-2}\theta_k+\theta_K\\
=&\theta_{K-1}+\sum_{k=1}^{K-2}\theta_k+\theta_K\\
=&0.
\end{aligned}
$$

Thus, $S_2\subseteq S_1$. The proof is complete.

(e) *Disadvantage:* Assuming the function is linear near the boundaries leads to bias in those regions, but is often considered reasonable.

III. Smoothing Splines

1. **Motivation:** Assume we have a set of pairs of observations $\{(x_i,y_i)\}_{i=1}^n$. Instead of selecting the number and the locations of knots in cubic spline, we use a maximal set of knots and control the complexity of the fit by regularization. The resulting spline basis method is called *smoothing spline*.

2. **Problem Statement:** Among all functions f with two continuous derivatives, we wish to find the one that minimizes the penalized residual sum of squares

$$\mathrm{RSS}_\lambda(f):=\underbrace{\sum_{i=1}^n(y_i-f(x_i))^2}_{T_1}+\lambda\underbrace{\int\big(f''(t)\big)^2\mathrm{d}t}_{T_2},\tag{6}$$

where $\lambda>0$ is a smoothing parameter.

Remark. T_1 measures the goodness of fit of f to the data, and T_2 penalizes curvature in the function and controls the complexity of f (recall that the second-order derivative is a measure of roughness/curvature), and $\lambda>0$ establishes a tradeoff between the two.

3. **Effects of λ:** Consider the following two extreme cases:

- $\lambda = 0$: f can be any function that interpolates the data;
- $\lambda = \infty$: f is the simple least squares line fit, since no second derivative can be tolerated.

Hence, λ indexes an class of functions from very rough to very smooth.

4. **Characterizing the Solution to (6):** The problem (6) is defined on an infinite-dimensional function space — the Sobolev space of functions for which the second-order derivative is defined. We show that the solution is of finite dimension, is unique, and is a natural cubic spline with knots at the unique values of the x_i's, for $i = 1, \cdots, n$.

   Suppose that $n \geq 2$, and that g is the natural cubic spline interpolant to the pairs $\{(x_i, y_i)\}_{i=1}^n$, with $-\infty < a < x_1 < \cdots < x_n < b$. This is a natural spline with a knot at every x_i. Let $\tilde{g}$ be any other differentiable function on $[a, b]$ that interpolates the n pairs, and $h := \tilde{g} - g$.

   Let $x_0 = a$ and $x_{n+1} = b$. Using the piecewise nature of g and the integration by parts, we have

$$\begin{aligned}\int_a^b g''(x)h''(x)\mathrm{d}x &= \sum_{i=0}^{n} \int_{x_i}^{x_{i+1}} g''(x)h''(x)\mathrm{d}x \\ &= \sum_{i=0}^{n} \left[g''(x)h'(x)\Big|_{x_i}^{x_{i+1}} - \int_{x_i}^{x_{i+1}} g'''(x)h'(x)\mathrm{d}x \right].\end{aligned}$$

   Using the fact that g is a natural cubic spline that is twice continuously differentiable and is linear beyond x_1 and x_n, we have

$$\begin{aligned}\sum_{i=0}^{n} g''(x)h'(x)\Big|_{x_i}^{x_{i+1}} &= g''(x_1)h'(x_1) - g''(a)h'(a) + \sum_{i=1}^{n-1} \big(g''(x_{i+1})h'(x_{i+1}) - g''(x_i)h'(x_i)\big) \\ &\quad + g''(b)h'(b) - g''(x_n)h'(x_n) \\ &= g''(b)h'(b) - g''(a)h'(a) \\ &= 0.\end{aligned}$$

Applying the integration by parts again, we have

$$
\begin{aligned}
&\sum_{i=0}^{n} \int_{x_i}^{x_{i+1}} g'''(x)h'(x)\mathrm{d}x \\
=& \sum_{i=0}^{n} \left[g'''(x)h(x)\Big|_{x_i}^{x_{i+1}} - \int_{x_i}^{x_{i+1}} g^{(4)}(x)h(x)\mathrm{d}x \right] \\
\overset{\text{(i)}}{=}& \sum_{i=0}^{n} g'''(x)h(x)\Big|_{x_i}^{x_{i+1}} \\
=& g'''(x_1^-)h(x_1^-) - g'''(a^+)h(a^+) + \sum_{i=1}^{n-1} \big(g'''(x_{i+1}^-)h(x_{i+1}^-) - g'''(x_i^+)h(x_i^+) \big) \\
&\quad + g'''(b^-)h(b^-) - g'''(x_n^+)h(x_n^+) \\
\overset{\text{(ii)}}{=}& \sum_{i=1}^{n-1} \big(g'''(x_{i+1}^-)h(x_{i+1}) - g'''(x_i^+)h(x_i) \big) \\
\overset{\text{(iii)}}{=}& \sum_{i=1}^{n-1} \big(g'''(x_i^+)h(x_{i+1}) - g'''(x_i^+)h(x_i) \big) \\
=& \sum_{i=1}^{n-1} g'''(x_i^+) \big(h(x_{i+1}) - h(x_i) \big) \\
\overset{\text{(iv)}}{=}& 0.
\end{aligned}
$$

In the derivation above, we use the fact $\int_{x_i}^{x_{i+1}} g^{(4)}(x)h(x)\mathrm{d}x = 0$ for all $i = 0, \cdots, n$ in (i), since g is a cubic spline and has zero fourth derivative, use $g'''(x_1^-) = g'''(a^+) = g'''(b^-) = g'''(x_n^+) = 0$ in (ii), use g''' is a piecewise constant function and $g'''(x_i^+) = g'''(x_{i+1}^-)$ in (iii), and use $h = \tilde{g} - g$ and both g and $\tilde{g}$ interpolates the n pairs of data points in (iv).

Next, we show that $\int_a^b (\tilde{g}''(t))^2\mathrm{d}t \geq \int_a^b (g''(t))^2\mathrm{d}t$ and that the equality can only hold if h is identically zero on $[a, b]$. By the definition of h, we have $\tilde{g} = h + g$. Hence,

$$
\begin{aligned}
\int_a^b (g''(t))^2\mathrm{d}t &= \int_a^b \big(h''(t) + g(t) \big)^2\mathrm{d}t \\
&= \int_a^b (h''(t))^2\mathrm{d}t + 2\int_a^b h''(t)g''(t)\mathrm{d}t + \int_a^b (g''(t))^2\mathrm{d}t \\
&= \int_a^b (h''(t))^2\mathrm{d}t + \int_a^b (g''(t))^2\mathrm{d}t \\
&\geq \int_a^b (g''(t))^2\mathrm{d}t.
\end{aligned}
$$

The last inequality becomes an equality if and only if $\int_a^b (h''(t))^2\mathrm{d}t = 0$, which means $h''(x) = 0$ for all $x \in (a, b)$. This implies $h(x) = cx + d$ for some $c, d \in \mathbb{R}$. Since $n \geq 2$

and $h(x_i) = g(x_i) - \tilde{g}(x_i) = 0$ for all $i = 1, \cdots, n$, this implies that $c = d = 0$, i.e., h is identically zero.

Finally, we argue that, for a fixed $\lambda > 0$, the minimizer to (6) must be a cubic spline with knots at each of the x_i. Denote this minimizer to be $\hat{g}$. Suppose that we have a $\tilde{g} \neq \hat{g}$ that interpolates the data points. Then, we have

$$\sum_{i=1}^{n}(y_i - \hat{g}(x_i))^2 = \sum_{i=1}^{n}(y_i - \tilde{g}(x_i))^2.$$

In addition, $\lambda \int_a^b (\tilde{g}''(t))^2 \mathrm{d}t > \lambda \int_a^b (\hat{g}''(t))^2 \mathrm{d}t$. In other words, RSS_λ does *not* achieve the minimum at $\tilde{g}$. The desired result follows.

Remark. Consider to minimize the following function

$$\mathrm{RSS}_{\mathbf{w},\lambda}(f) := \sum_{i=1}^{n} w_i (y_i - f(x_i))^2 + \lambda \int \big(f''(t)\big)^2 \mathrm{d}t,$$

where $\mathbf{w} := (w_1, \cdots, w_n)^\top$ is the weight vector and w_i is the weight of the i-th observation. Using a similar argument as above, we can show the minimizer of $\mathrm{RSS}_{\mathbf{w},\lambda}$ is also a natural spline with knots at unique observations $x_1, \cdots, x_n$.

5. **Algorithm:** Since the solution to (6) is a natural cubic spline, we know it must be of the form

$$f(x) = \sum_{j=1}^{n} \theta_j N_j(x), \tag{7}$$

where the $\{N_j\}_{j=1}^n$'s are an n-dimensional set of basis functions for representing this family of natural splines.

Plugging (7) into (6) yields the following criterion

$$\mathrm{RSS}_\lambda(\boldsymbol{\theta}) := (\mathbf{Y} - \mathbf{N}\boldsymbol{\theta})^\top (\mathbf{Y} - \mathbf{N}\boldsymbol{\theta}) + \lambda \boldsymbol{\theta}^\top \boldsymbol{\Omega} \boldsymbol{\theta},$$

where $[\mathbf{N}]_{i,j} = N_j(x_i)$ and

$$[\boldsymbol{\Omega}]_{j,k} = \int N_j''(t) N_k''(t)\, \mathrm{d}t.$$

Then,

$$\begin{aligned} \frac{\partial \mathrm{RSS}_\lambda(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} &= -2\mathbf{N}^\top(\mathbf{Y} - \mathbf{N}\boldsymbol{\theta}) + 2\lambda\boldsymbol{\Omega}\boldsymbol{\theta}, \\ \frac{\partial^2 \mathrm{RSS}_\lambda(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2} &= 2(\mathbf{N}^\top\mathbf{N} + \lambda\boldsymbol{\Omega}) \succ 0. \end{aligned}$$

We set the first-order derivative to be $\mathbf{0}_n$ and obtain the solution to $\boldsymbol{\theta}$

$$\widehat{\boldsymbol{\theta}} = (\mathbf{N}^\top\mathbf{N} + \lambda\boldsymbol{\Omega})^{-1}\mathbf{N}^\top\mathbf{Y},$$

Then, plug $\widehat{\boldsymbol{\theta}}$ into (7), we obtain the fitted smoothing spline is

$$\hat{f}_\lambda(x) = \sum_{j=1}^{N} \hat{\theta}_j N_j(x),$$

where $\hat{\theta}_j$ is the j-th component of $\widehat{\boldsymbol{\theta}}$.

6. **Smoother Matrix in Smoothing Spline:** We write the fitted values collectively in the matrix notation as

$$\widehat{\mathbf{Y}} = \underbrace{\mathbf{N}(\mathbf{N}^\top\mathbf{N} + \lambda\boldsymbol{\Omega})^{-1}\mathbf{N}^\top}_{\mathbf{S}_\lambda}\mathbf{Y}.$$

The matrix $\mathbf{S}_\lambda := \mathbf{N}(\mathbf{N}^\top\mathbf{N} + \lambda\boldsymbol{\Omega})^{-1}\mathbf{N}^\top$ is called the *smoother matrix*.

Remark. Note that the fitted value vector $\widehat{\mathbf{Y}}$ is linear in $\mathbf{Y}$ and, therefore, a smoothing spline with a pre-specified λ is an example of a *linear smoother*. Also note that the smoother matrix $\mathbf{S}_\lambda$ only depends on x_i and λ.

7. **Degrees of Freedom in Smoothing Spline:** The *effective degrees of freedom* a smoothing spline is

$$\mathrm{df}_\lambda = \mathrm{trace}(\mathbf{S}_\lambda).$$

8. **Smoother Matrix and Degrees of Freedom in Cubic Spline:** In the cubic spline setting, let $\mathbf{B}_{\boldsymbol{\xi}}$ be an $n \times M$ matrix of M cubic-spline basis functions evaluated at the n training points x_i, with knot sequence $\boldsymbol{\xi}$, and $M \ll n$. The fitted spline value vector is

$$\widehat{\mathbf{Y}} = \mathbf{B}_{\boldsymbol{\xi}}(\mathbf{B}_{\boldsymbol{\xi}}^\top\mathbf{B}_{\boldsymbol{\xi}})^{-1}\mathbf{B}_{\boldsymbol{\xi}}^\top\mathbf{Y} = \mathbf{H}_{\boldsymbol{\xi}}\mathbf{Y}.$$

Then, the linear operator $\mathbf{H}_{\boldsymbol{\xi}}$ is a *projection operator* or the *hat matrix*.

Here, $M = \mathrm{trace}(\mathbf{H}_{\boldsymbol{\xi}})$ is the dimension of the projection space, which is also the number of basis functions, and hence the number of parameters involved in the fit.

9. **A Comparison between $\mathbf{S}_\lambda$ and $\mathbf{H}_{\boldsymbol{\xi}}$:**
   (a) Both are symmetric, positive semidefinite matrices;
   (b) $\mathbf{H}_{\boldsymbol{\xi}}$ is idempotent, meaning that $\mathbf{H}_{\boldsymbol{\xi}} = \mathbf{H}_{\boldsymbol{\xi}}\mathbf{H}_{\boldsymbol{\xi}}$, while $\mathbf{S}_\lambda$ is *not* and $\mathbf{S}_\lambda\mathbf{S}_\lambda \preceq \mathbf{S}_\lambda$;
   (c) $\mathbf{H}_{\boldsymbol{\xi}}$ has rank M, while $\mathbf{S}_\lambda$ has rank n.

10. **Further Analysis of the Smoother Matrix $\mathbf{S}_\lambda$:**
    (a) Assuming $\mathbf{N}$ is invertible, we can write $\mathbf{S}_\lambda$ in the *Reinsch form* as

$$\begin{aligned}\mathbf{S}_\lambda &= \mathbf{N}(\mathbf{N}^\top\mathbf{N} + \lambda\boldsymbol{\Omega})^{-1}\mathbf{N}^\top \\ &= \mathbf{N}\big(\mathbf{N}^\top(\mathbf{I} + \lambda(\mathbf{N}^\top)^{-1}\boldsymbol{\Omega}\mathbf{N}^{-1})\mathbf{N}\big)^{-1}\mathbf{N}^\top \\ &= (\mathbf{I} + \lambda\mathbf{K})^{-1},\end{aligned}$$

where $\mathbf{K} := (\mathbf{N}^\top)^{-1}\boldsymbol{\Omega}\mathbf{N}^{-1}$ does *not* depend on λ.

(b) We can re-write the optimization problem (7) as

$$\underset{\mathbf{f}}{\text{minimize}}\,(\mathbf{Y}-\mathbf{f})^\top(\mathbf{Y}-\mathbf{f})+\lambda\mathbf{f}^\top\mathbf{K}\mathbf{f},$$

where $\mathbf{K}$ is known as the *penalty matrix*. The minimizer is exactly

$$\widehat{\mathbf{Y}}=(\mathbf{I}+\lambda\mathbf{K})^{-1}\mathbf{Y}=\mathbf{S}_\lambda\mathbf{Y}.$$

(c) By the spectral decomposition theorem, we can write the smoother matrix $\mathbf{S}_\lambda$ as

$$\mathbf{S}_\lambda=\sum_{i=1}^{n}\rho_i(\lambda)\mathbf{u}_i\mathbf{u}_i^\top$$

where

$$\rho_i(\lambda)=\frac{1}{1+\lambda d_i}$$

and d_i's are the eigenvalues of $\mathbf{K}$. In addition, $\mathbf{S}_\lambda$ and $\mathbf{K}$ have the same set of eigenvectors. In other words, the eigenvectors are *not* affected by changes in $\lambda>0$.

(d) By the definition of the fitted values and the eigen-decomposition above, we have

$$\mathbf{S}_\lambda\mathbf{Y}=\sum_{i=1}^{n}\rho_i(\lambda)\langle\mathbf{u}_i,\mathbf{Y}\rangle\mathbf{u}_i.$$

The interpretation is that the smoothing spline decomposes $\mathbf{Y}$ with respect to the (complete) basis $\{\mathbf{u}_i\}_{i=1}^n$, and differentially shrinks the contributions according to $\rho_i(\lambda)$. Thus, the smoothing splines are referred to as *shrinking* smoothers.

Remark. In contrast, the projection matrix $\mathbf{H}_{\boldsymbol{\xi}}$ has M eigenvalues equal to 1 and the rest are 0. The regression splines (with fixed choices of knots) are *projection* smoothers.

(e) With the decrease in $\rho_i(\lambda)$, the sequence $\{\mathbf{u}_i\}_i$ increases in complexity. Due to the identity

$$\mathbf{S}_\lambda\mathbf{u}_i=\rho_i(\lambda)\mathbf{u}_i,$$

the higher the complexity is, the more they are shrunk.

(f) The first two eigenvalues are always one, and they correspond to the two-dimensional eigen-space of functions linear in x, which are never shrunk.

(g) The eigenvalues of $\mathbf{S}_\lambda$,

$$\rho_i(\lambda)=\frac{1}{1+\lambda d_i},\qquad\text{for all } i=1,2,\cdots,n,$$

are an inverse function of the eigenvalues d_i of the penalty matrix $\mathbf{K}$, moderated by $\lambda>0$. Here, λ controls the rate at which $\rho_i(\lambda)$ decrease to 0. By the preceding remark, $d_1=d_2=0$ and, thus, linear functions are not penalized.

(h) *Reparametrization:* One can re-parametrize the smoothing spline using $\{\mathbf{u}_i\}_i$, called the *Demmler-Reinsch basis*. Then, (6) can be written as

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \ \|\mathbf{Y} - \mathbf{U}\boldsymbol{\theta}\|_2^2 + \lambda \boldsymbol{\theta}^\top \mathbf{D}\boldsymbol{\theta},$$

where $\mathbf{U}$ has columns $\mathbf{u}_i$ and $\mathbf{D}$ is a diagonal matrix with elements $\{d_i\}_{i=1}^n$.

(i) Notice that

$$\mathrm{df}_\lambda = \mathrm{trace}(\mathbf{S}_\lambda) = \sum_{k=1}^{n} \rho_k(\lambda).$$

IV. Automatic Selection of the Smoothing Parameters

1. **Parameters:**

   (a) *Regression spline:* the degree of the splines, the number of knots, and the placement of knots;

   (b) *Smoothing spline:* the penalty parameter $\lambda > 0$.

2. **Fix the Degrees of Freedom:** In smoothing splines, recall that

   $$\mathrm{df}_\lambda = \mathrm{trace}(\mathbf{S}_\lambda),$$

   which is monotonically decreasing in λ. One can invert this relationship and specify $\lambda > 0$ by fixing the degrees of freedom.

3. **The Bias-Variance Tradeoff Via EPE:** Suppose that the response variables are standardized so that $\mathrm{Cov}(\mathbf{Y}) = \mathbf{I}$. Then, the *covariance matrix* of the fitted value vector is

   $$\mathrm{Var}(\widehat{\mathbf{Y}}) = \mathbf{S}_\lambda \,\mathrm{Var}(\mathbf{Y})\mathbf{S}_\lambda^\top = \mathbf{S}_\lambda \mathbf{S}_\lambda^\top,$$

   where the diagonal elements are the pointwise variances at the training x_i.

   Also, the *bias* is

   $$\mathrm{Bias}(\widehat{\mathbf{Y}}) = \mathbf{f} - \mathbb{E}[\widehat{\mathbf{Y}}] = \mathbf{Y} - \mathbf{S}_\lambda \mathbf{Y} = (\mathbf{I} - \mathbf{S}_\lambda)\mathbf{Y},$$

   where $\mathbf{Y}$ is the *unknown* vector of evaluations of the true f at the training data points.

   We next calculate the integrated squared expected prediction error (EPE) at a new point X as follows:

   $$\begin{aligned}
   \mathrm{EPE}[\hat{f}_\lambda(X)] &= \mathbb{E}\big[(Y - \hat{f}_\lambda(X))^2\big] \\
   &= \mathrm{Var}(Y) + \mathbb{E}\big[\mathrm{Bias}^2(\hat{f}_\lambda(X)) + \mathrm{Var}[\hat{f}_\lambda(X)]\big] \\
   &= \sigma^2 + \mathrm{MSE}(\hat{f}_\lambda(X)).
   \end{aligned}$$

   Remark. The quantity EPE is averaged both over the training set (which produces $\hat{f}_\lambda$) and at a new point (X, Y) *independently* from the training set.

4. **The Bias-Variance Tradeoff via Cross-Validation:** We use the *n-fold cross-validation*, also known as *leave-one-out cross-validation*, defined by

$$\begin{aligned}\mathrm{CV}(\hat{f}_\lambda) &= \frac{1}{n}\sum_{i=1}^{n}\big(y_i - \hat{f}_\lambda^{(-i)}(x_i)\big)^2 \\ &= \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{f}_\lambda(x_i)}{1-\mathbf{S}_\lambda(i,i)}\right)^2, \end{aligned} \tag{8}$$

where $\mathbf{S}_\lambda(i,i)$ is the i-th diagonal element in the smoother matrix $\mathbf{S}_\lambda$. We show (8) holds.

Note that it is sufficient to show that

$$y_i - \hat{f}_\lambda^{(-i)}(x_i) = \frac{y_i - \hat{f}_\lambda(x_i)}{1-\mathbf{S}_\lambda(i,i)}, \qquad \text{for all } i = 1, \cdots, n,$$

and, without the loss of generality, is sufficient to show the case $i = 1$.
Let $\widetilde{\mathbf{Y}} := (\tilde{y}_1, \tilde{y}_2, \cdots, \tilde{y}_n)^\top = (\hat{f}_\lambda^{(-1)}(x_1), y_2, \cdots, y_n)^\top$. We show that $\hat{f}_\lambda^{(-1)}$ minimizes

$$\widetilde{\mathrm{RSS}}_\lambda(f) := \sum_{i=1}^{n}(\tilde{y}_i - f(x_i))^2 + \lambda\int_a^b (f''(t))^2 \mathrm{d}t.$$

Note that

$$\begin{aligned}\widetilde{\mathrm{RSS}}_\lambda(\hat{f}_\lambda^{(-1)}) &= \sum_{i=1}^{n}(\tilde{y}_i - \hat{f}_\lambda^{(-1)}(x_i))^2 + \lambda\int_a^b (\hat{f}_\lambda^{(-1)''}(t))^2 \mathrm{d}t \\ &= \sum_{i=2}^{n}(\tilde{y}_i - \hat{f}_\lambda^{(-1)}(x_i))^2 + \lambda\int_a^b (\hat{f}_\lambda^{(-1)''}(t))^2 \mathrm{d}t \\ &< \sum_{i=2}^{n}(\tilde{y}_i - f(x_i))^2 + \lambda\int_a^b (f''(t))^2 \mathrm{d}t \\ &\le \sum_{i=1}^{n}(\tilde{y}_i - f(x_i))^2 + \lambda\int_a^b (f''(t))^2 \mathrm{d}t \\ &= \widetilde{\mathrm{RSS}}_\lambda(f). \end{aligned}$$

Thus,

$$\begin{aligned}\tilde{y}_1 &= \sum_{i=1}^{n}\mathbf{S}_\lambda(1,i)\tilde{y}_i \\ &= \mathbf{S}_\lambda(1,1)\tilde{y}_1 + \sum_{i=2}^{n}\mathbf{S}_\lambda(1,i)y_i + \mathbf{S}_\lambda(1,1)y_1 - \mathbf{S}_\lambda(1,1)y_1 \\ &= \mathbf{S}_\lambda(1,1)(\tilde{y}_1 - y_1) + \sum_{i=1}^{n}\mathbf{S}_\lambda(1,i)y_i \\ &= \mathbf{S}_\lambda(1,1)(\tilde{y}_1 - y_1) + \hat{f}_\lambda(x_1). \end{aligned}$$

We plug $\tilde{y}_1 = \hat{f}_\lambda^{(-1)}(x_1)$ into the preceding equation and rearrange terms, yielding

$$\hat{f}_\lambda^{(-1)}(x_1) = \frac{\hat{f}_\lambda(x_1) - \mathbf{S}_\lambda(1,1)y_1}{1 - \mathbf{S}_\lambda(1,1)}.$$

Finally,

$$y_1 - \hat{f}_\lambda^{(-1)}(x_1) = y_1 - \frac{\hat{f}_\lambda(x_1) - \mathbf{S}_\lambda(1,1)y_1}{1 - \mathbf{S}_\lambda(1,1)} = \frac{y_1 - \hat{f}_\lambda(x_1)}{1 - \mathbf{S}_\lambda(1,1)}.$$

5. **A Comparison between EPE and CV:**
   - The EPE and CV curves have a similar shape;
   - Overall, the CV curve is approximately unbiased as an estimate of the EPE curve.

V. Nonparametric Logistic Regression

1. **Setup:** We consider the *binary logistic regression* with a single quantitative input. The model is

$$\log \frac{\mathbb{P}(Y=1 \,|\, X=x)}{\mathbb{P}(Y=0 \,|\, X=x)} = f(x),$$

for some f, and, hence,

$$\mathbb{P}(Y=1 \,|\, X=x) = \frac{\exp(f(x))}{1+\exp(f(x))}.$$

We fit f in a smooth fashion and this will lead to a smooth estimate of the conditional probability $\mathbb{P}(Y=1 \,|\, X=x)$, which can be used for *classification*.

2. **Penalized Log-likelihood Function:** Let $p(x) := \mathbb{P}(Y=1 \mid X=x)$. We consider to maximize the following *penalized log-likelihood function*

$$\begin{aligned}
\ell_\lambda(f) &= \sum_{i=1}^n \Big(y_i \log p(x_i) + (1-y_i)\log(1-p(x_i)) \Big) - \frac{1}{2}\lambda \int (f''(t))^2 \mathrm{d}t \\
&= \sum_{i=1}^n \Big(y_i \log p(x_i) - y_i \log(1-p(x_i)) + \log(1-p(x_i)) \Big) - \frac{1}{2}\lambda \int (f''(t))^2 \mathrm{d}t \\
&= \sum_{i=1}^n \left(y_i \log \frac{p(x_i)}{1-p(x_i)} + \log(1-p(x_i)) \right) - \frac{1}{2}\lambda \int (f''(t))^2 \mathrm{d}t \\
&= \sum_{i=1}^n \left(y_i \log(\exp(f(x_i))) + \log\left(\frac{1}{1+\exp(f(x))} \right) \right) - \frac{1}{2}\lambda \int (f''(t))^2 \mathrm{d}t \\
&= \sum_{i=1}^n \big[y_i f(x_i) - \log(1+\exp\big(f(x_i)\big)) \big] - \frac{1}{2}\lambda \int (f''(t))^2 \mathrm{d}t. \qquad (9)
\end{aligned}$$

The first term is the log-likelihood based on the binomial distribution.

3. **Characterizing the Solution to (9):** The maximizer to (9) is a finite-dimensional natural spline with knots at the unique values of x, and therefore, the solution is of the form

$$f(x) = \sum_{j=1}^{n} \theta_j N_j(x),$$

where $\{N_j\}_{j=1}^n$ is a set of basis functions spanning this function space.

4. **Applying the Newton-Raphson Algorithm:** With the characterization above, ℓ_λ is essentially a function of parameters $\boldsymbol{\theta} := (\theta_1, \theta_2, \cdots, \theta_n)^\top$.

We find the derivatives of ℓ_λ with respect to $\boldsymbol{\theta}$

$$\begin{aligned}\frac{\partial \ell_\lambda(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} &= \mathbf{N}^\top(\mathbf{y} - \mathbf{p}) - \lambda\boldsymbol{\Omega}\boldsymbol{\theta},\\ \frac{\partial^2 \ell_\lambda(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2} &= -\mathbf{N}^\top\mathbf{W}\mathbf{N} - \lambda\boldsymbol{\Omega},\end{aligned}$$

where $\mathbf{p}$ is the n-dimensional vector with the i-th element being $p(x_i)$, and $\mathbf{W}$ is a diagonal matrix of weights $p(x_i)(1 - p(x_i))$.

We use the Newton-Raphson algorithm to compute the maximizer. The update equation can be written as

$$\begin{aligned}\boldsymbol{\theta}^{(\text{new})} &= (\mathbf{N}^\top\mathbf{W}\mathbf{N} + \lambda\boldsymbol{\Omega})^{-1}\mathbf{N}^\top\mathbf{W}(\mathbf{N}\boldsymbol{\theta}^{(\text{old})} + \mathbf{W}^{-1}(\mathbf{y} - \mathbf{p}))\\ &= (\mathbf{N}^\top\mathbf{W}\mathbf{N})^{-1}\mathbf{N}^\top\mathbf{W}\mathbf{z},\end{aligned}$$

where $\mathbf{z} := \mathbf{N}\boldsymbol{\theta}^{(\text{old})} + \mathbf{W}^{-1}(\mathbf{y} - \mathbf{p})$ is the *working response*.

We can also express $\boldsymbol{\theta}^{(\text{new})}$ in terms of the fitted values. Note that

$$\begin{aligned}\mathbf{f}^{(\text{new})} = \mathbf{N}\boldsymbol{\theta}^{(\text{new})} &= \mathbf{N}(\mathbf{N}^\top\mathbf{W}\mathbf{N} + \lambda\boldsymbol{\Omega})^{-1}\mathbf{N}^\top\mathbf{W}(\mathbf{f}^{(\text{old})} + \mathbf{W}^{-1}(\mathbf{y} - \mathbf{p}))\\ &= \mathbf{S}_{\lambda,w}\mathbf{z}.\end{aligned}$$

VI. Multidimensional Splines

1. **Tensor Product Basis:** Let $x = (x_1, x_2) \in \mathbb{R}^2$. Suppose we have a set of basis functions $\{h_{1j}\}_{j=1}^{M_1}$ to represent functions of coordinate x_1 and a set of basis functions $\{h_{2k}\}_{k=1}^{M_2}$ for x_2. Then, the $M_1 \times M_2$ dimensional *tensor product basis* is defined to be

$$g_{jk}(x) = h_{1j}(x_1) \cdot h_{2k}(X_2), \qquad \text{for all } j = 1, \cdots, M_1 \text{ and } k = 1, \cdots, M_2.$$

This set of $M_1 \times M_2$ basis functions can be used for representing a two-dimensional function:

$$g(x) = \sum_{j=1}^{M_1}\sum_{k=1}^{M_2} \theta_{jk} g_{jk}(X),$$

where the coefficients θ_{jk} can be determined by using least squares.

Remark. There is nothing particular about $p = 2$, and one can generalize to dimensions larger than 2.

2. **General Setup for Higher-Dimensional Smoothing Splines via Regularization:** Suppose we have pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ with $\mathbf{x}_i \in \mathbb{R}^p$. We seek a p-dimensional regression function f in the smoothing spline setting. The associated optimization problem is

$$\underset{f}{\text{minimize}} \sum_{i=1}^{n} \big(y_i - f(\mathbf{x}_i)\big)^2 + \lambda J(f), \tag{10}$$

where $\lambda > 0$ and J is an appropriate penalty functional for stabilizing a function f in $\mathbb{R}^p$.

3. **Thin-Plate Spline:** Suppose $p = 2$. We can choose the roughness penalty to be

$$J(f) = \int_{\mathbb{R}^2} \left[\left(\frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} \right)^2 + 2 \left(\frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} \right)^2 + \left(\frac{\partial^2 f(\mathbf{x})}{\partial x_2^2} \right)^2 \right] \mathrm{d}x_1 \mathrm{d}x_2, \tag{11}$$

which is a natural generalization of $d = 1$ case. Optimizing (10) with this particular J leads to a smooth two-dimensional surface known as a *thin-plate* spline.

   (a) *Effects of λ:* The effect of λ is similar as before:
   - as $\lambda \to 0$, the solution approaches an interpolating function;
   - as $\lambda \to \infty$, the solution approaches the least squares plane;
   - for intermediate values of λ, the solution can be represented as a linear expansion of basis functions with coefficients obtained by a form of generalized ridge regression.

   (b) *Characterization of the Solution:* The solution to the (10) with the penalty defined by (11) is of the following form

$$f(\mathbf{x}) = \beta_0 + \boldsymbol{\beta}^\top \mathbf{x} + \sum_{j=1}^{n} \alpha_j h_j(\mathbf{x}), \tag{12}$$

   where $h_j(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_j\|^2 \log\|\mathbf{x} - \mathbf{x}_j\|$, an example of *radial basis functions.*
   To determine the coefficients, we plug (12) into (10) and obtain a finite-dimensional penalized least squares problem. The coefficients α_j's need to satisfy a set of linear constraints.

Remarks.

- Thin-plate splines can be defined more generally for arbitrary dimension p with an appropriately more general penalty J is used;

- The computational complexity for thin-plate splines is $\mathcal{O}(n^3)$. However, in practice, we need *not* use all n knots and, instead, work with a lattice of knots covering the domain. Using $K \ll n$ knots, the computational complexity reduces to $\mathcal{O}(nK^2 + K^3)$.

4. **Caution:** In general, one can represent $f \in \mathbb{R}^p$ as an expansion in any arbitrarily large collection of basis functions, and control the complexity by regularization. However, the number of basis functions can grow exponentially as the dimensionality increases, and we have to reduce the number of functions per coordinate accordingly.

5. **Additive Spline Model:** Consider the case $p \geq 2$ and $\mathbf{x} := (x_1, \cdots, x_p)^\top \in \mathbb{R}^p$.

   (a) Assume $f(\mathbf{x}) = \alpha + f_1(x_1) + \cdots + f_p(x_p)$ is additive and impose a penalty on each of the component functions

   $$J(f) = J(f_1 + f_2 + \cdots + f_d) = \sum_{j=1}^{d} \int (f_j''(t_j))^2 \mathrm{d}t_j.$$

   (b) Assume f is the *ANOVA spline decomposition* of the form

   $$f(\mathbf{x}) = \alpha + \sum_j f_j(x_j) + \sum_{j<k} f_{jk}(x_j, x_k) + \cdots, \tag{13}$$

   where each of the components is a spline of the required dimension.

   Remark. In (13), we can choose

   (a) the maximum order of interactions;

   (b) which terms to include — not all of the main effects and interactions are needed;

   (c) the basis functions:

   - we can choose a relatively small number of basis functions per coordinate and use the tensor product for interactions; *or*
   - We can choose a complete basis and include appropriate regularizer for each term in the expansion.

VII. Regularization and Reproducing Kernel Hilbert Space

1. **Overview:** We consider a general class of regularization problem of the form

   $$\underset{f \in \mathcal{H}}{\text{minimize}} \left\{ \sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i)) + \lambda J(f) \right\}, \tag{14}$$

   where L is a loss function, $J : \mathcal{H} \to [0, \infty)$ is a penalty functional, and $\mathcal{H}$ is a space of functions on which $J(f)$ is defined.

2. A First Example: Consider the following penalty function

$$J(f) = \int_{\mathbb{R}^p} \frac{|\tilde{f}(\mathbf{s})|^2}{\widetilde{G}(\mathbf{s})} \mathrm{d}\mathbf{s},$$

where $\tilde{f}$ denotes the Fourier transform of f, and $\widetilde{G}$ is some positive function that falls off to zero as $\|\mathbf{s}\| \to \infty$. Then the solution is of the following form

$$f(\mathbf{x}) = \sum_{m=1}^{M} \alpha_m \phi_m(\mathbf{x}) + \sum_{i=1}^{n} \theta_i G(\mathbf{x} - \mathbf{x}_i),$$

where $\{\phi_m\}_{m=1}^M$ spans the null space of the penalty functional J, and G is the inverse Fourier transform of $\widetilde{G}$.

Remarks.

(a) Smoothing splines and thin-plate splines fall into this framework.

(b) The remarkable feature of this approach is that while the criterion (14) is defined over an infinite-dimensional space, its solution is finite-dimensional.

3. Space of Functions Generated by Kernels:

(a) *Kernel:* A function $K : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ is called a *kernel* if there exists a Hilbert space $\mathcal{H}$ and a map $\phi : \mathbb{R}^p \to \mathcal{H}$ such that, for all $x, x' \in \mathbb{R}^p$, we have

$$K(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle_{\mathcal{H}}.$$

We call $\phi : \mathbb{R}^p \to \mathcal{H}$ a *feature map* and $\mathcal{H}$ a *feature space* of K.

(b) *Reproducing Property:* A kernel K is said to have the *reproducing property* if, for any $f \in \mathcal{H}$,

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

Such a kernel function K is called a *reproducing kernel*, or the *representer of evaluation*. In particular, if $f = K(\mathbf{x}, \cdot)$, then,

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

(c) *(Strictly) Positive Definite Kernel:* A kernel function $K : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ is *positive definite* if $K(\mathbf{x}, \mathbf{y}) = K(\mathbf{y}, \mathbf{x})$ for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^p$ and K is positive definite

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j K(\mathbf{x}_i, \mathbf{x}_j) \geq 0. \tag{15}$$

for all $n \geq 1$, all $c_1, \cdots, c_n \in \mathbb{R}$, and all $\mathbf{x}_1, \cdots, \mathbf{x}_n \in \mathbb{R}^p$. It is said to be *strictly positive definite* if the equality in (15) holds only when $c_i = 0$ for all i.

(d) *Reproducing Kernel Hilbert Space (RKHS):* A *reproducing kernel Hilbert space (RKHS)*, $\mathcal{H}$, is a space of functions generated by a positive definite kernel.

(e) *Functions in $\mathcal{H}$:* Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^p$ and consider the space of functions generated by the linear span of $\{K(\mathbf{y}, \cdot) \,|\, y \in \mathbb{R}^p\}$, i.e., arbitrary linear combinations of the form

$$f(\mathbf{x}) = \sum_{m \in \mathbb{N}} \alpha_m K(\mathbf{x}, \mathbf{y}_m).$$

Suppose that K has an eigen-expansion

$$K(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{\infty} \gamma_i \psi_i(\mathbf{x}) \psi_i(\mathbf{y}),$$

where $\gamma_i \geq 0$ and $\sum_{i=1}^{\infty} \gamma_i^2 < \infty$. Then, functions in $\mathcal{H}$ can be written in terms of these eigenfunctions,

$$f(\mathbf{x}) = \sum_{i=1}^{\infty} c_i \psi_i(\mathbf{x})$$

with the constraint that

$$\|f\|_{\mathcal{H}}^2 := \sum_{i=1}^{\infty} \frac{c_i^2}{\gamma_i} < \infty,$$

where $\|f\|_{\mathcal{H}}$ is the norm induced by K.

4. **Equivalent Formulation of (14):** If we let $J(f)$ in (14) for $f \in \mathcal{H}$ be the squared norm of f, i.e.,

$$J(f) = \|f\|_{\mathcal{H}}^2,$$

which can be interpreted as a *generalized ridge penalty*, we can rewrite (14) as

$$\underset{f}{\text{minimize}} \left\{ \sum_{i=1}^{n} L\big(y_i, f(\mathbf{x}_i)\big) + \lambda \|f\|_{\mathcal{H}}^2 \right\}. \tag{16}$$

or, equivalently,

$$\underset{\{c_i\}_{i=1}^{\infty}}{\text{minimize}} \left\{ \sum_{i=1}^{n} L\left(y_i, \sum_{j=1}^{\infty} c_j \psi_j(\mathbf{x}_i)\right) + \lambda \sum_{j=1}^{\infty} \frac{c_j^2}{\gamma_j} \right\}.$$

5. **Characterizing the Solution to (16):** The solution to (16) is finite-dimensional and has the form

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}, \mathbf{x}_i). \tag{17}$$

Let $\tilde{f} = f + \rho$, where f is of the form (17) and belongs to the span of $\mathcal{H}_0 := \text{Span}\{K(\mathbf{x}_1, \cdot), K(\mathbf{x}_2, \cdot), \cdots, K(\mathbf{x}_n, \cdot)\}$ and $\rho \in \mathcal{H}$ belongs to the orthogonal complement of $\mathcal{H}_0$. Then, we have

$$\begin{aligned}\tilde{f}(\mathbf{x}_i) &= \langle \tilde{f}, K(\mathbf{x}_i, \cdot)\rangle_{\mathcal{H}} = \langle f + \rho, K(\mathbf{x}_i, \cdot)\rangle_{\mathcal{H}} \\ &= \langle f, K(\mathbf{x}_i, \cdot)\rangle_{\mathcal{H}} + \langle \rho, K(\mathbf{x}_i, \cdot)\rangle_{\mathcal{H}} = f(\mathbf{x}_i),\end{aligned}$$

and

$$J(\tilde{f}) = \|f + \rho\|_{\mathcal{H}}^2 = \|f\|_{\mathcal{H}}^2 + 2\langle f, \rho\rangle_{\mathcal{H}} + \|\rho\|_{\mathcal{H}}^2 = \|f\|_{\mathcal{H}}^2 + \|\rho\|_{\mathcal{H}}^2.$$

Thus,

$$\begin{aligned}\sum_{i=1}^n L\big(y_i, \tilde{f}(\mathbf{x}_i)\big) + \lambda\|\tilde{f}\|_{\mathcal{H}}^2 &= \sum_{i=1}^n L\big(y_i, f(\mathbf{x}_i)\big) + \lambda\Big(\|f\|_{\mathcal{H}}^2 + \|\rho\|_{\mathcal{H}}^2\Big) \\ &\geq \sum_{i=1}^n L\big(y_i, f(\mathbf{x}_i)\big) + \lambda\|f\|_{\mathcal{H}}^2,\end{aligned}$$

with the inequality becoming an equality if and only if $\rho = 0$. In other words, (16) is minimized by functions of the form (17).

6. **Optimization Problem in Matrix Form:** We can rewrite the optimization problem (16) in the following matrix form

$$\underset{\boldsymbol{\alpha}}{\text{minimize}} \;\; L(\mathbf{Y}, \mathbf{K}\boldsymbol{\alpha}) + \lambda\boldsymbol{\alpha}\mathbf{K}\boldsymbol{\alpha},$$

where $\boldsymbol{\alpha} := (\alpha_1, \cdots, \alpha_n)^\top \in \mathbb{R}^n$, and $\mathbf{K}$ is the $n \times n$ matrix with (i, j)-entry being $K(\mathbf{x}_i, \mathbf{x}_j)$.

7. **Partial Penalization:** Sometimes, we want to leave some components in $\mathcal{H}$ alone and do *not* penalize them. Decompose the space $\mathcal{H}$ as $\mathcal{H} = \mathcal{H}_0 \oplus \mathcal{H}_1$, with the null space $\mathcal{H}_0$ consisting of components that do not get penalized. The penalty becomes $J(f) = \|P_1 f\|_{\mathcal{H}}^2$, where P_1 is the orthogonal projection of f onto $\mathcal{H}_1$. The solution has the form

$$f(\mathbf{x}) = \sum_{j=1}^M \beta_j h_j(\mathbf{x}) + \sum_{i=1}^n \alpha_i K(\mathbf{x}, \mathbf{x}_i),$$

where the first term represents an expansion of functions in $\mathcal{H}_0$.

8. **Example — Penalized Least Squares:** We consider the penalized least squares problem where we solve the following (infinite-dimensional) optimization problem

$$\underset{\{c_i\}_{i=1}^\infty}{\text{minimize}} \sum_{i=1}^n \left(y_i - \sum_{j=1}^\infty c_j\psi_j(\mathbf{x}_i)\right)^2 + \lambda\sum_{j=1}^\infty \frac{c_j^2}{\gamma_j},$$

called the *generalized ridge regression problem.*

By the earlier argument, the solution is of the form $\sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, \cdot)$. We can rewrite the preceding infinite-dimensional optimization problem as the following finite-dimensional problem in the matrix form

$$\underset{\boldsymbol{\alpha}}{\text{minimize}}\ (\mathbf{Y} - \mathbf{K}\boldsymbol{\alpha})^\top (\mathbf{Y} - \mathbf{K}\boldsymbol{\alpha}) + \lambda \boldsymbol{\alpha}\mathbf{K}\boldsymbol{\alpha},$$

where the (i, j)-entry of $\mathbf{K} \in \mathbb{R}^{n\times n}$ is $K(\mathbf{x}_i, \mathbf{x}_j)$. The solution is

$$\widehat{\boldsymbol{\alpha}} := (\mathbf{K} + \lambda \mathbf{I})^{-1}\mathbf{Y},$$

and the fitted value at $\mathbf{x}$ is

$$\hat{f}(\mathbf{x}) = \sum_{j=1}^{n} \hat{\alpha}_j K(\mathbf{x}, \mathbf{x}_j),$$

where $\hat{\alpha}_j$ is the j-th component of $\widehat{\boldsymbol{\alpha}}$.

Collectively, the fitted value vector is

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

9. Example — Penalized Polynomial Regression: We consider the polynomial kernel of the form

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^\top \mathbf{y} + 1)^d, \qquad \text{for all } \mathbf{x}, \mathbf{y} \in \mathbb{R}^p.$$

It has $M := \binom{p+d}{d}$ eigen-functions that span the space of polynomials in $\mathbb{R}^p$ of total degree d.

(a) *Example:* With $p = 2$, $d = 2$ and $M = 6$, we have

$$\begin{aligned} K(\mathbf{x}, \mathbf{y}) &= 1 + 2x_1y_1 + 2x_2y_2 + x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2 \\ &= \sum_{m=1}^{M} h_m(\mathbf{x})h_m(\mathbf{y}) \end{aligned}$$

with

$$\mathbf{h}(\mathbf{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)^\top.$$

(b) One can represent $\mathbf{h}$ in terms of the M orthogonal eigenfunctions and eigenvalues of K as

$$\mathbf{h}(\mathbf{x}) = \mathbf{V}\mathbf{D}_\gamma^{1/2}\boldsymbol{\psi}(\mathbf{x}),$$

where $\mathbf{D}_\gamma = \text{diag}(\gamma_1, \gamma_2, \cdots, \gamma_M)$, $\mathbf{V}$ is an $M \times M$ orthogonal matrix, and $\boldsymbol{\psi}(\mathbf{x}) := (\psi_1(\mathbf{x}), \psi_2(\mathbf{x}), \cdots, \psi_M(\mathbf{x}))^\top$.

(c) We wish to solve the following penalized polynomial regression problem

$$\underset{\{\beta_m\}_{m=1}^M}{\text{minimize}} \sum_{i=1}^n \left(y_i - \sum_{m=1}^M \beta_m h_m(\mathbf{x}_i) \right)^2 + \lambda \sum_{m=1}^M \beta_m^2. \tag{18}$$

Let $\widehat{\boldsymbol{\beta}} := (\hat{\beta}_1, \cdots, \hat{\beta}_M)^\top$ be the minimizer of (18). Then,

$$\widehat{\boldsymbol{\beta}} = \mathbf{H}^\top (\mathbf{H}\mathbf{H}^\top + \lambda \mathbf{I})^{-1} \mathbf{Y},$$

where $\mathbf{H} \in \mathbb{R}^{n \times M}$ with rows given by $\mathbf{h}(\mathbf{x}_i)^\top$, and $\mathbf{Y} := (y_1, \cdots, y_n)^\top \in \mathbb{R}^n$.

(d) *Caution:* Notice that the number of basis functions $M = \binom{p+d}{d}$ can be very large!

10. Example — Gaussian Radial Basis Functions: The *Gaussian kernel* is $K(\mathbf{x}, \mathbf{y}) = e^{-\nu \|\mathbf{x}-\mathbf{y}\|_2^2}$. Then, the Gaussian radial basis functions are

$$k_m(\mathbf{x}) = e^{-\nu \|\mathbf{x}-\mathbf{x}_m\|_2^2}, \qquad \text{for } m = 1, \cdots, n,$$

where each one is centered at one of the training feature vectors $\mathbf{x}_m$.

- For a kernel matrix $\mathbf{K}$, where each entry is calculated as $\mathbf{K}_{ml} = k_m(\mathbf{x}_l)$ for all $m, l = 1, \cdots, n$, we compute its eigen-decomposition $\mathbf{K} = \boldsymbol{\Psi} \mathbf{D}_\gamma \boldsymbol{\Psi}^\top$. We can think of the columns of $\boldsymbol{\Psi}$ and the corresponding eigenvalues in $\mathbf{D}_\gamma$ as empirical estimates of the eigen-expansion $K(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^\infty \gamma_i \psi_i(\mathbf{x}) \psi_i(\mathbf{y})$.
- Although in principle the implicit feature space is *infinite* dimensional, the effective dimension is dramatically lower.
- The kernel scale parameter ν plays a role here as well: larger ν implies more local k_m functions, and increases the effective dimension of the feature space.

11. Example — Support Vector Machines: The support vector machines for a two-class classification problem have the form

$$f(\mathbf{x}) = \alpha_0 + \sum_{i=1}^n \alpha_i K(\mathbf{x}, \mathbf{x}_i),$$

where the parameters are chosen to minimize

$$\underset{\alpha_0, \boldsymbol{\alpha}}{\text{minimize}} \left\{ \sum_{i=1}^n \left[1 - y_i \cdot f(\mathbf{x}_i) \right]_+ + \frac{\lambda}{2} \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} \right\}, \tag{19}$$

with $y_i \in \{-1, 1\}$, $\boldsymbol{\alpha} := (\alpha_1, \alpha_2, \cdots, \alpha_n)^\top$, and $[z]_+ := \max\{z, 0\}$ denoting the positive part of z.

This can be viewed as a *quadratic* optimization problem with *linear* constraints.

The name *support vector* arises from the fact that typically many components of the $\widehat{\boldsymbol{\alpha}}$ equal to 0 due to the piecewise-zero nature of the loss function in (19), and so $\hat{f}$ is an expansion in a subset of the $\{K(\mathbf{x}_i, \cdot)\}_{i=1}^n$.

References

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The Elements of Statistical Learning*. Vol. 1. Springer Series in Statistics. New York, NY, USA: Springer New York Inc.