Chapter 4

Basis Expansions

4.1 Introduction

Many techniques will impose a (simple) structure for e.g. the decision boundary in classification or the regression function (next part). For instance, LDA and many of its extensions assume linear decision boundaries while many regression techniques consider linear regression. Such a relatively simple structure is often a convenient or even necessary restriction, but will in general not be the correct structure but only an approximation. As already mentioned, the restriction can be relaxed by increasing the predictor space by adding e.g. squares and cross-products of the measured predictors and using the technique in the enlarged/transformed predictor space.

In general, consider transformations $h_m(X) : \mathbb{R}^d \to \mathbb{R}$ of X, $m = 1, \ldots, M$. The new transformed predictor space is then given by $h(X) = (h_1(X), \ldots, h_M(X))$. A linear technique or *linear expansion in* X, $T(X) = \sum_{j=1}^{d} \beta_j X_j$ can consequently be extended to

$$\tilde{T}(X) = T(h(X)) = \sum_{m=1}^{M} \beta_m h_m(X),$$

a *linear basis expansion in* X.

Some standard transformations are

- $h_m(X) = X_m, m = 1 \ldots, d$ which uses the original measured predictors.
- Adding transformations $h_m(X) = X_j^2,\ h_m(X) = X_s X_t, \ldots$ allows to augment the feature space with polynomial terms as mentioned above. Note, that the number of variables grows exponentially in the degree of polynomials considered.

- $h_m(X) = \log(X_j)$, $h_m(X) = \sqrt{X_j}$, $h_m(X) = \|X\|, \ldots$ are standard nonlinear transformations of measured predictors.

The choice of transformations can be driven by the problem at hand or transformations can be used to achieve more flexibility. If the goal is flexibility, then instead of global transformations such as polynomials that can have strange effects in remote parts of the data space, it is more convenient to use local polynomial representations that fit well in some region of the space without adverse effects elsewhere. Useful local families are *piecewise-polynomials* and *splines*. The process of replacing an original set of measured features X by a transformed set of new features $X^\star$ is often called *filtering* or feature extraction and this process is often called the *preprocessing* phase of the data analysis.

4.2 Piecewise Polynomials

We assume that X is one-dimensional (and thus can be any variable in the feature space). To construct a piecewise polynomial function $\tilde{T}(X)$, the domain of X is divided in contiguous intervals. In each of these intervals, the function $\tilde{T}$ is represented by a separate polynomial. A basic example is obtained by using piecewise constant functions. Suppose that we split the domain of X into three intervals $X < \xi_1$, $\xi_1 \leq X < \xi_2$, $\xi_2 \leq X$. Consider the following piecewise constant basis functions

$$h_1(X) = I(X < \xi_1), \quad h_2(X) = I(\xi_1 \leq X < \xi_2), \quad h_3(X) = I(\xi_2 \leq X),$$

where $I(.)$ is the *indicator function* that takes the value 1 within the specified interval and zero elsewhere. The piecewise polynomial function $\tilde{T}(X)$ is now given by $\tilde{T}(X) = \sum_{m=1}^{3} \beta_m h_m(X)$, a linear function of the transformed predictors. Since only one predictor is different from zero in each of the intervals, the coefficients β_m can be estimated by $\hat{\beta}_m = \bar{Y}_m$ with $\bar{Y}_m$ the mean of the Y values in the mth region. In discriminant analysis the Y variables will be dummy variables corresponding to class membership for each of the classes. In regression Y will be the quantitative response variable.

Fitting piecewise linear functions is obtained by adding three additional basis functions

$$h_{m+3}(X) = h_m(X)X, \quad m = 1, 2, 3.$$

These functions thus equal X within the specified interval but are zero outside that interval. This leads to the approximation $\tilde{T}(X) = \sum_{m=1}^{6} \beta_m h_m(X)$.

Both of the above approximations have the drawback of not being continuous at the *knots*, that is the endpoints of the intervals. To introduce continuity at the knots we need two restrictions $\tilde{T}(\xi_1^-) = \tilde{T}(\xi_1^+)$ which implies

$\beta_1+\xi_1\beta_4 = \beta_2+\xi_1\beta_5$ and $\tilde{T}(\xi_2^-) = \tilde{T}(\xi_2^+)$ which implies $\beta_2+\xi_2\beta_5 = \beta_3+\xi_2\beta_6$. These two equations fix two parameter values and hence there are four free parameters that need to be estimated from the data.

4.3 Splines

A more direct approach to impose continuity and other desirable properties is to use basis functions that incorporate the necessary constraints. For the above example of a piecewise linear continuous fit we can use the following basis functions

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

where $(.)_+$ means that the resulting function equals the function between brackets when that function is positive and the resulting function is zero when the function between brackets is negative. Note that the number of basis functions now corresponds to the number of free parameters.

4.3.1 Cubic splines

We often prefer to use approximations that are smoother than the piecewise linear approximation above. This can be achieved by using local higher order polynomials. A continuous approximation that has continuous first and second order derivatives also at the knots is obtained by using the following basis functions which is called the *cubic spline with knots at* ξ_1 *and* ξ_2.

$$\begin{aligned} h_1(X) = 1, \quad h_3(X) = X^2, \quad h_5(X) = (X-\xi_1)_+^3, \\ h_2(X) = X, \quad h_4(X) = X^3, \quad h_6(X) = (X-\xi_2)_+^3 \end{aligned} \tag{4.1}$$

Hence, we fit local cubic functions with constraints at the knots that ensure that at the knots the values of the left and right parts have the same function value and the same values of the first and second derivatives.

In theory splines can be defined with constraints beyond second derivatives but in practice there is seldom any need to go beyond cubic splines. Splines with fixed knots introduced here are often called *regression splines*. In practice, one needs to select the order of the spline, the number of knots, and their location. Often the observations are used as the knots.

4.3.2 Natural cubic splines

Higher order polynomial fits tend to be very sensitive, which can lead to erratic results near the boundaries and beyond. This makes extrapolation

extremely dangerous. Cubic regression splines tend to behave even more wildly than global polynomial fits around and beyond the boundary. To reduce this adverse effect, *natural cubic splines* require that the resulting function is linear beyond the boundary knots. As usual, there will be a price in bias near the boundaries of the domain, but assuming that the function is linear near the boundary (where we have the least information) is often reasonable. A natural cubic spline can be represented by the cubic spline basis (4.1) together with the linearity constraints at the boundary. If there are q knots $\xi_1, \ldots, \xi_q$, this leads to the basis functions

$$\begin{aligned} h_1^N(X) &= 1,\ h_2^N(X) = X,\ h_j^N(X) = d_{j-2}(X) - d_{q-1}(X);\ j = 3, \ldots, q \\ \text{with } d_l(X) &= \frac{(X-\xi_l)_+^3 - (X-\xi_q)_+^3}{\xi_q - \xi_l}. \end{aligned}$$

4.4 Smoothing Splines

Smoothing splines produce a spline basis with automatic knot selection and thus avoid the knot selection problem. Smoothing splines are obtained as the solution of the following optimization problem. Among all functions with second order continuous derivatives, find the function that minimizes the penalized residual sum of squares.

$$\mathrm{RSS}(f, \lambda) = \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int (f''(t))^2 \, dt \tag{4.2}$$

The first term of (4.2) measures the goodness-of-fit, that is the closeness of the fit to the data. The second term of (4.2) is the *penalty term* that penalizes curvature in the function. Functions with curvature have second order derivatives different from zero and thus get penalized. Hence, the criterion shows a preference for simpler (linear) fits if they fit the data reasonably well. The parameter λ is the *smoothing parameter* which controls the trade-off between goodness-of-fit and curvature. A larger value of λ means a bigger preference for simple linear functions, a smaller λ value means less penalization for more complex functions with curvature. The two limit cases are

$\lambda = 0$: No penalty. f can be any function that fits the data exactly leading to a RSS equal to zero.

$\lambda = \infty$: No curvature is allowed. f will be the best fitting linear fit. Since the first term in (4.2) is a least squares criterion, the solution will be the least squares regression line.

The solution thus varies from very rough ($\lambda = 0$) to very smooth ($\lambda = \infty$) and the idea is to select a value of λ that leads to a good compromise.

It can be shown that for any fixed λ, the minimization of (4.2) leads to a unique solution which is a *natural cubic spline* with knots at the x_i values. Without penalty ($\lambda = 0$) we would thus find a solution that exactly fits the data. However, the penalty term forces that some of the coefficients β_m in the fit $\tilde{T}(X) = \sum_{m=1}^{n} \beta_m h_m^N(X)$ are shrunken towards zero such that the resulting fit is pushed towards a linear fit.

4.4.1 Splines as linear operators and effective degrees of freedom

Let the $n \times n$ matrix $\boldsymbol{N}$ denote the transformed feature variable, that is $\boldsymbol{N}_{ij} = h_j^N(x_i)$. Criterion (4.2) can now be rewritten as

$$RSS(\beta, \lambda) = (\mathbf{y} - \boldsymbol{N}\beta)^t(\mathbf{y} - \boldsymbol{N}\beta) + \lambda\beta^t\boldsymbol{\Omega}\beta, \tag{4.3}$$

where $\boldsymbol{\Omega}_{ij} = \int N_i''(t)N_j''(t)\,dt$ with $N(t) = (h_1^N(t), \ldots, h_n^N(t))^t$. The solution can be shown to be

$$\hat{\beta} = (\boldsymbol{N}^t\boldsymbol{N} + \lambda\boldsymbol{\Omega})^{-1}\boldsymbol{N}^t\mathbf{y}.$$

Let $\hat{\mathbf{f}} = (\hat{f}(x_1), \ldots, \hat{f}(x_n))^t$ denote the vector of fitted values at the training data, then

$$\begin{aligned} \hat{\mathbf{f}} &= \boldsymbol{N}\hat{\beta} = \boldsymbol{N}(\boldsymbol{N}^t\boldsymbol{N} + \lambda\boldsymbol{\Omega})^{-1}\boldsymbol{N}^t\mathbf{y} \\ &= \boldsymbol{S}_\lambda\mathbf{y}. \end{aligned} \tag{4.4}$$

This means that the fitted values are a linear function of the responses $\mathbf{y}$. The linear operator $\boldsymbol{S}_\lambda$ is known as the *smoother matrix*. Note that this is similar to least squares regression where we have $\hat{\mathbf{f}}_{LS} = \boldsymbol{H}\mathbf{y}$ with $\boldsymbol{H} = \boldsymbol{X}(\boldsymbol{X}^t\boldsymbol{X})^{-1}\boldsymbol{X}^t$ the hat matrix. As for least squares, the linear operator $\boldsymbol{S}_\lambda$ does not depend on $\mathbf{y}$ but only on the x_i observations and λ.

When using regression splines (e.g. cubic splines), we first transform the predictor variable X to an M-dimensional ($M < n$) spline basis space leading to an $n \times M$ matrix $\boldsymbol{B}_\xi$ for the transformed feature space where ξ is the knot sequence used by the regression spline. Applying least squares to find the regression coefficients yields the fitted values

$$\begin{aligned} \hat{\mathbf{f}} &= \boldsymbol{B}_\xi\hat{\beta} = \boldsymbol{B}_\xi(\boldsymbol{B}_\xi^t\boldsymbol{B}_\xi)^{-1}\boldsymbol{B}_\xi^t\mathbf{y} \\ &= \boldsymbol{H}_\xi\mathbf{y}. \end{aligned}$$

The hat matrix $\boldsymbol{H}_\xi$ is now a projection matrix that projects the response vector $\mathbf{y}$ onto the M-dimensional space spanned by the columns of the new feature matrix $\boldsymbol{B}_\xi$.

In both cases (smoothing splines or regression splines) the fitted values are obtained as a linear combination of the observed response vector $\mathbf{y}$. There are some important similarities between the smoother matrix $\boldsymbol{S}_\lambda$ and the projection matrix $\boldsymbol{H}_\xi$, but also some important differences.

- Both matrices are symmetric, positive semidefinite matrices, but $\boldsymbol{S}_\lambda$ is not idempotent.
- $\boldsymbol{H}_\xi$ has rank $M < n$, while $\boldsymbol{S}_\lambda$ has rank n. That is, while $\boldsymbol{H}_\xi$ projects $\mathbf{y}$ on the M-dimensional space spanned by the columns of $\boldsymbol{B}_\xi$, the smoother matrix $\boldsymbol{S}_\lambda$ does not project $\mathbf{y}$ onto a space of lower dimension, but instead finds an approximation in the full n-dimensional space that satisfies the conditions imposed by the penalty term.
- $\text{trace}(\boldsymbol{H}_\xi) = M$ the dimension of the projection space which corresponds with the number of (free) parameters in the fit. Note that the trace of a square matrix is defined as the sum of its diagonal elements. Analogously, the *effective degrees of freedom* of a smoothing spline fit is defined as

$$\text{df}_\lambda = \text{trace}(\boldsymbol{S}_\lambda). \tag{4.5}$$

This definition of the effective degrees of freedom can be used to select the value of the parameter λ. A smaller effective degrees of freedom means a larger λ and vice versa. This is explained below in more detail.

The smoother matrix $\boldsymbol{S}_\lambda$ can be rewritten as

$$\begin{aligned}
\boldsymbol{S}_\lambda &= \boldsymbol{N}(\boldsymbol{N}^t\boldsymbol{N} + \lambda\boldsymbol{\Omega})^{-1}\boldsymbol{N}^t \\
&= \boldsymbol{N}[\boldsymbol{N}^t(\boldsymbol{N} + \lambda(\boldsymbol{N}^t)^{-1}\boldsymbol{\Omega})]^{-1}\boldsymbol{N}^t \\
&= \boldsymbol{N}[\boldsymbol{N}^t(\boldsymbol{I} + \lambda(\boldsymbol{N}^t)^{-1}\boldsymbol{\Omega}\boldsymbol{N}^{-1})\boldsymbol{N}]^{-1}\boldsymbol{N}^t \\
&= \boldsymbol{N}\boldsymbol{N}^{-1}[\boldsymbol{I} + \lambda(\boldsymbol{N}^t)^{-1}\boldsymbol{\Omega}\boldsymbol{N}^{-1}]^{-1}(\boldsymbol{N}^t)^{-1}\boldsymbol{N}^t \\
&= [\boldsymbol{I} + \lambda(\boldsymbol{N}^t)^{-1}\boldsymbol{\Omega}\boldsymbol{N}^{-1}]^{-1} \\
&= [\boldsymbol{I} + \lambda\boldsymbol{K}]^{-1}
\end{aligned} \tag{4.6}$$

with $\boldsymbol{K} = (\boldsymbol{N}^t)^{-1}\boldsymbol{\Omega}\boldsymbol{N}^{-1}$. Note that $\boldsymbol{K}$ does not depend on λ. Using $\mathbf{f} = \boldsymbol{N}\beta$, criterion (4.3) can be rewritten as

$$\min_{\mathbf{f}}(\mathbf{y} - \mathbf{f})^t(\mathbf{y} - \mathbf{f}) + \lambda\mathbf{f}^t\boldsymbol{K}\mathbf{f}, \tag{4.7}$$

For this reason $\boldsymbol{K}$ is called the *penalty matrix*. Since $\boldsymbol{S}_\lambda$ is a symmetric positive semidefinite matrix, it has a real eigen-decomposition

$$\boldsymbol{S}_\lambda = \sum_{j=1}^{n} \rho_j(\lambda)\mathbf{v}_j\mathbf{v}_j^t \tag{4.8}$$

$$\text{with } \rho_j(\lambda) = \frac{1}{1 + \lambda d_j} \tag{4.9}$$

where d_j are the corresponding eigenvalues of $\boldsymbol{K}$. From this representation we see that the eigenvectors do not change by changes in λ. This could intuitively be expected because $\boldsymbol{S}_\lambda$ is not a projection matrix but looks for an approximation in the full n-dimensional space, as explained before. The eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ are a basis for the full n-dimensional space. The smoother matrix $\boldsymbol{S}_\lambda$ transforms $\mathbf{y}$ into

$$\boldsymbol{S}_\lambda \mathbf{y} = \sum_{j=1}^{n} \rho_j(\lambda)(\mathbf{v}_j^t \mathbf{y})\mathbf{v}_j$$

The decomposition of $\mathbf{y}$ is $\mathbf{y} = \sum_{j=1}^{n}(\mathbf{v}_j^t \mathbf{y})\mathbf{v}_j$. Hence the linear smoother first decomposes the vector $\mathbf{y}$ w.r.t. the basis $\mathbf{v}_1, \ldots, \mathbf{v}_n$ and then adds factors $\rho_j(\lambda)$ to the weight $(\mathbf{v}_j^t \mathbf{y})$ of each basisvector in $\mathbf{y}$. Since $\boldsymbol{S}_\lambda$ has rank n, the factors $\rho_j(\lambda)$ are all larger than zero and also smaller than 1 as can be seen from (4.9). The size of a factor $\rho_j(\lambda)$ determines the amount of shrinkage that is used for the contribution of each of the basisvectors.

In contrast, a least squares regression spline method is based on a projection matrix $\boldsymbol{H}_\xi$ of rank M. The property that a projection matrix is idempotent implies that the eigenvalues $\rho_j(\boldsymbol{H}_\xi)$ of $\boldsymbol{H}_\xi$ can only take the values zero or one. M eigenvalues will be one and the remaining $n - M$ eigenvalues are zero. This means that the contribution of a basisvector to $\mathbf{y}$ is either fully used, or ignored (shrunk to zero). This contrast is often reflected by calling smoothing splines *shrinking smoothers* while calling regression splines *projection smoothers*.

It follows from (4.9) that the eigenvalues $\rho_j(\lambda)$ of $\boldsymbol{S}_\lambda$ are an inverse function of the eigenvalues d_j of $\boldsymbol{K}$. Hence, small eigenvalues of $\boldsymbol{S}_\lambda$ correspond to large eigenvalues of $\boldsymbol{K}$ and vice versa. The smoothing parameter λ controls the rate by which the eigenvalues $\rho_j(\lambda)$ of $\boldsymbol{S}_\lambda$ decrease to zero. The penalty matrix $\boldsymbol{K}$ depends on $\boldsymbol{\Omega}$, the matrix of second derivatives of the basis functions. Since the first two basis functions $h_1^N(X) = 1$ and $h_2^N(X) = X$ have second derivatives equal to zero, the corresponding eigenvalues of $\boldsymbol{K}$ will be zero. From (4.9) it then follows that the corresponding eigenvalues of $\boldsymbol{S}_\lambda$ are always equal to 1. Hence, the linear contributions to $\mathbf{y}$ are never shrunk.

Since the trace of a matrix also equals the sum of its eigenvalues, we have that

$$\mathrm{df}_\lambda = \mathrm{trace}(\boldsymbol{S}_\lambda) = \sum_{j=1}^{n} \rho_j(\lambda). \tag{4.10}$$

Larger values of λ lead to more severe penalization and yields smaller eigenvalues (according to (4.9)), and thus also a smaller degrees of freedom. On the other hand, smaller λ values correspond to larger degrees of freedom.

For the limiting case $\lambda = 0$ it follows from (4.6) that $\boldsymbol{S}_\lambda = \boldsymbol{I}$ which implies that $\rho_j(\lambda) = 1$ for $j = 1, \ldots, n$ and thus $\mathrm{df}_\lambda = n$. On the other hand, for $\lambda \to \infty$, all eigenvalues $\rho_j(\lambda) = 0$ except the first two for which $d_j = 0$ and thus $\rho_1(\lambda) = \rho_2(\lambda) = 1$ as always. Hence, $\mathrm{df}_\lambda = 2$ and in fact, since the eigenvalues of $\boldsymbol{S}_\lambda$ are all zero or one, $\boldsymbol{S}_\lambda$ now equals the projection matrix $\boldsymbol{H}$, the hat matrix for linear regression of $\mathbf{y}$ on X.

Since df_λ is a monotone function of the smoothing parameter λ, we can specify the degrees of freedom and derive λ from it. For instance, in R, the function *smooth.spline* takes the effective degrees of freedom as a possible input parameter to specify the amount of smoothing desired. Using the effective degrees of freedom allows to parallel model selection for spline methods with model selection for traditional parametric methods. Spline based fits corresponding to different effective degrees of freedom can be compared and investigated using residual plots and other criteria, to select an optimal fit.

As explained before, the selection of the smoothing parameter λ and thus the selection of the effective degrees of freedom df_λ is a trade-off between bias and variance. We illustrate this trade-off with the following example.

$$Y = f(X) + \epsilon \quad \text{with} \quad f(X) = \frac{\cos(10(X + 1/4))}{(X + 1/4)},$$

with $X \sim U[0, 1]$ and $\epsilon \sim N(0, 1)$. The training sample consists of $n = 100$ samples (x_i, y_i) generated independently from this model. We fit smoothing splines for three different effective degrees of freedom 3,6, and 12 to these data.

Note that from (4.4) it immediately follows that

$$\mathrm{Cov}(\hat{\mathbf{f}}) = \mathrm{Cov}(\boldsymbol{S}_\lambda \mathbf{y}) = \boldsymbol{S}_\lambda \mathrm{Cov}(\mathbf{y}) \boldsymbol{S}_\lambda^t.$$

Since $\mathrm{Cov}(\mathbf{y}) = \mathrm{Cov}(\epsilon) = \boldsymbol{I}$ in this example, we obtain that $\mathrm{Cov}(\hat{\mathbf{f}}) = \boldsymbol{S}_\lambda \boldsymbol{S}_\lambda^t$ and the diagonal elements of this matrix thus contain the variances of the fitted values.

Figure 4.1 shows the smoothing splines (red curves) with 3,6, and 12 effective degrees of freedom. The variance of the fits is represented by the pointwise confidence bands $\hat{f}(x_i) \pm 2\mathrm{se}(\hat{f}(x_i))$. The smoothing spline with $\mathrm{df}_\lambda = 3$ clearly underfits. The bias is most drastic in regions with high curvature. The confidence band is very narrow, so the biased curve is estimated with high precision. The smoothing spline with $\mathrm{df}_\lambda = 6$ fits much closer to the true function although there still is a small amount of bias. The variance of the fit is still small. The smoothing spline with $\mathrm{df}_\lambda = 12$ fits close to the true function but the fit is wiggly. This wiggliness is also visible

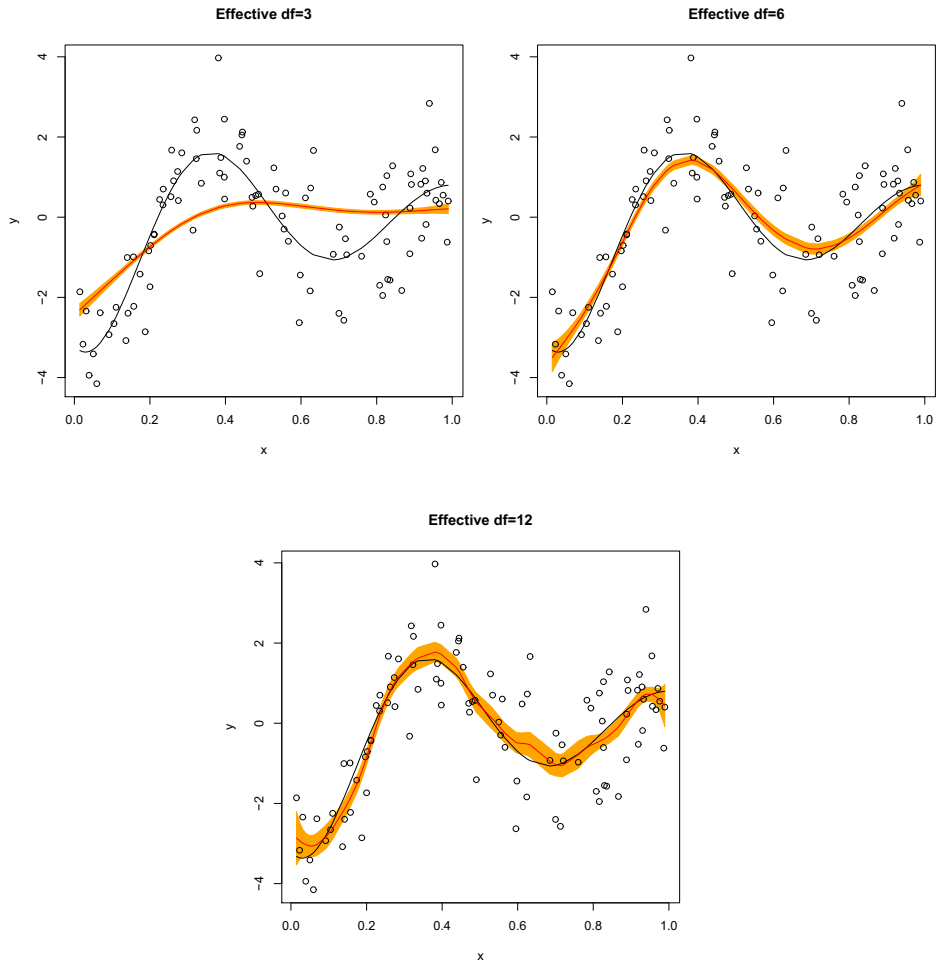

Figure 4.1: Nonlinear function f (black curve) with smoothing spline fits for three different values of the effective degrees of freedom.

in the confidence band that reveals the increased variance which now is the main cause for the difference between the fitted and true curves.

4.5 Multidimensional Splines

Here we briefly discuss some multivariate analogs of the univariate spline models introduced in the previous sections. Hence, instead of a one-dimensional predictor X, we now consider $X = (X_1, \ldots, X_d) \in \mathbb{R}^d$.

4.5.1 Multivariate regression splines

Consider $X = (X_1, \ldots, X_d) \in \mathbb{R}^d$ and suppose we have for each component X_j a (spline) basis of functions $h_{jm}(X_j)$; $m = 1, \ldots, M_j$. For simplicity we

show results for $d = 2$, but the generalization is straightforward.

The $M_1 + M_2$ dimensional *additive spline basis* is defined by

$$g_{jm}(X) = h_{jm}(X_j);\ j = 1, 2;\ m = 1, \ldots, M_j.$$

The function will be approximated by

$$f(X) = \sum_{j=1}^{2} \sum_{m=1}^{M_j} \beta_{jm} g_{jm}(X)$$

where the coefficients β_{jm} need to be estimated from the training data using e.g. least squares.

The $M_1 \times M_2$ dimensional *tensor product spline basis* is defined by

$$g_{jm}(X) = h_{1j}(X_1) h_{2m}(X_2);\ j = 1, \ldots, M_1;\ m = 1, \ldots, M_2.$$

The function will now be approximated by

$$f(X) = \sum_{j=1}^{M_1} \sum_{m=1}^{M_2} \beta_{jm} g_{jm}(X).$$

Again, the coefficients β_{jm} need to be estimated from the training data using e.g. least squares. Note that the dimension of this basis grows exponentially with the dimension d contrary to the additive basis. Hence, some regularization or selection will be necessary when using the tensor basis in higher dimensions.

4.5.2 Multivariate smoothing splines

Suppose we have observations (y_i, x_i) with $x_i \in \mathbb{R}^2$, then we look for the two-dimensional function $f(x)$ that minimizes

$$\mathrm{RSS}(f, \lambda) = \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda J(f) \tag{4.11}$$

where $J(f)$ is an appropriate penalty term that penalizes curvature in the bivariate surface. A straightforward generalization of the one-dimensional penalty in (4.2) is

$$J(f) = \int \int \left[\left(\frac{\partial^2 f(x)}{\partial x_1^2} \right)^2 + 2 \left(\frac{\partial^2 f(x)}{\partial x_1 \partial x_2} \right)^2 + \left(\frac{\partial^2 f(x)}{\partial x_2^2} \right)^2 \right] dx_1\, dx_2$$

The solution of (4.11) has the form

$$f(x) = \beta_0 + \beta^t x + \sum_{j=1}^{n} \alpha_j h_j(x),$$

where $h_j(x) = h(\|x - x_j\|)$ with $h(t) = t^2 \log(t^2)$. This is a smooth bivariate surface, which is often called the *thin-plate spline*. Note that the solution has the same form as in the one-dimensional case. The linear part is always present and the value of λ determines the amount of shrinking for the higher order terms. Higher values of λ imply more shrinkage. As for the one-dimensional smoothing spline it holds that when $\lambda \to \infty$, then the solution becomes the least squares plane. On the other hand, when $\lambda \to 0$, then the penalty term disappears and the solution is a function that fits the data exactly.

4.5.3 Hybrid splines

Some popular hybrid spline bases exist that try to allow sufficient flexibility for the fitted function but at the same time try to keep the dimension of the basis reasonable

Additive models assume that the true function can be approximated well by a function $f(X)$ of the form $f(X) = \beta_0 + f_1(X_1) + \cdots + f_d(X_d)$ where each of the functions f_j are univariate smoothing splines. A componentwise penalty is also imposed on the fit:

$$J(f) = \sum_{j=1}^{d} \int f_j''(t_j)^2 \, dt_j \tag{4.12}$$

A natural extension is to consider an approximation of the form

$$f(X) = \beta_0 + \sum_{j=1}^{d} f_j(X_j) + \sum_{j<m} f_{jm}(X_j, X_m)$$

where each component is a spline of the required dimension. Such approximations are called *ANOVA spline decompositions*. The splines used can be regression splines with the use of tensor products for the interactions, or smoothing splines with an appropriate penalty term as regularizer to keep the number of nonzero coefficients limited.

4.6 Regularization with Kernels

A general formulation of regularization problems is

$$\min_{f \in \mathcal{H}} \left[\sum_{i=1}^{n} L(y_i, f(x_i)) + \lambda J(f) \right] \tag{4.13}$$

where $L(y, f(x))$ is a loss function, $J(f)$ is a penalty term, and $\mathcal{H}$ is an (infinite-dimensional) space of functions on which the penalty term $J(f)$ is

well-defined. Under general assumptions, it can be shown that this problem has a finite-dimensional solution.

We will consider the important subclass of problem (4.13) where the space of functions $\mathcal{H}$ is a *reproducing kernel Hilbert space* (RKHS) generated by a positive definite kernel $K(x, y)$. Such a space will be denoted by $\mathcal{H}_K$. Moreover, we assume that the penalty term $J(f)$ can be written in terms of the kernel as well.

A positive definite kernel $K(x, y)$ has an *eigen-expansion*

$$K(x, y) = \sum_{i=1}^{\infty} \gamma_i \phi_i(x) \phi_i(y) \tag{4.14}$$

with $\gamma_i \geq 0$ and $\sum_{i=1}^{\infty} \gamma_i^2 < \infty$. The basis functions $\phi_i(x)$ form a basis for the infinite-dimensional (Hilbert) space $\mathcal{H}_K$ spanned by the kernel function K. Functions in $\mathcal{H}_K$ thus have an expansion of the form

$$f(x) = \sum_{i=1}^{\infty} a_i \phi_i(x) \tag{4.15}$$

The kernel $K(x, y)$ allows to define an *inner product* in the Hilbert space. For any two functions $f(x) = \sum_{i=1}^{\infty} a_i \phi_i(x)$ and $g(x) = \sum_{i=1}^{\infty} b_i \phi_i(x)$ in $\mathcal{H}_K$, the inner product is defined as

$$< f, g >= \sum_{i=1}^{\infty} \frac{a_i b_i}{\gamma_i}. \tag{4.16}$$

This inner product leads to the definition of a norm in the Hilbert space $\mathcal{H}_K$ given by

$$\|f\|^2_{\mathcal{H}_K} =< f, f >= \sum_{i=1}^{\infty} \frac{a_i^2}{\gamma_i}. \tag{4.17}$$

Functions f in $\mathcal{H}_K$ satisfy the constraint

$$\|f\|^2_{\mathcal{H}_K} < \infty \tag{4.18}$$

Note that for any $z \in \mathbb{R}^d$ the function $g(x) = K(x, z) = \sum_{i=1}^{\infty} [\gamma_i \phi_i(z)] \phi_i(x)$ satisfies (4.15) with $\|g\|^2_{\mathcal{H}_K} = \sum_{i=1}^{\infty} \frac{\gamma_i^2 \phi_i(z)^2}{\gamma_i} = \sum_{i=1}^{\infty} \gamma_i \phi_i(z)^2 = K(z, z) < \infty$. Hence, $g(x) = K(x, z) \in \mathcal{H}_K$. For a fixed $z \in \mathbb{R}^d$, the function $g(x) = K(x, z)$ will be denoted by $K(., z)$ to indicate that we consider the kernel as a function of its first argument with the second argument fixed.

The penalty term $J(f)$ in (4.13) is taken equal to $J(f) = \|f\|^2_{\mathcal{H}_K}$. This penalty term has a similar interpretation as the smoothing spline penalty. The contributions to f of basis functions $\phi(x)$ with large eigenvalue γ_i will get penalized less while contributions of basis functions with small eigenvalue γ_i will get penalized more.

The minimization problem (4.13) now becomes

$$\min_{f \in \mathcal{H}_K} \left[\sum_{i=1}^{n} L(y_i, f(x_i)) + \lambda \|f\|^2_{\mathcal{H}_K} \right]. \tag{4.19}$$

Using (4.15) and (4.17), this can be rewritten as

$$\min_{a_j;\, j=1,\ldots,\infty} \left[\sum_{i=1}^{n} L(y_i, \sum_{j=1}^{\infty} a_j \phi_j(x_i)) + \lambda \sum_{j=1}^{\infty} \frac{a_j^2}{\gamma_j} \right]. \tag{4.20}$$

It has been shown that this problem has a finite-dimensional solution in $\mathcal{H}_K$ that can be written in the form

$$f(x) = \sum_{i=1}^{n} \alpha_i K(x, x_i). \tag{4.21}$$

For the kernel function K it holds that

$$\begin{aligned} < K(., x_i), K(., x_j) > &= \sum_{m=1}^{\infty} \frac{\gamma_m \phi_m(x_i) \gamma_m \phi_m(x_j)}{\gamma_m} \\ &= \sum_{m=1}^{\infty} \gamma_m \phi_m(x_i) \phi_m(x_j) \\ &= K(x_i, x_j). \end{aligned} \tag{4.22}$$

For any $g \in \mathcal{H}_K$ it holds that

$$< g, K(., x_i) > = \sum_{m=1}^{\infty} \frac{b_m \gamma_m \phi_m(x_i)}{\gamma_m} = \sum_{m=1}^{\infty} b_m \phi_m(x_i) = g(x_i). \tag{4.23}$$

This property is known as the *reproducing property of the kernel* K. It implies that any function $g \in \mathcal{H}_K$ can be approximated arbitrary well by a function f of form (4.21).

For a function f of the form (4.21) it follows further that

$$\begin{aligned} < f, K(., x_i) > &= < \sum_{j=1}^{n} \alpha_j K(., x_j), K(., x_i) > \\ &= \sum_{j=1}^{n} \alpha_j < K(., x_j), K(., x_i) > \\ &= \sum_{j=1}^{n} \alpha_j K(x_i, x_j). \end{aligned} \tag{4.24}$$

Using (4.21) and (4.22) the penalty term $J(f) = \|f\|^2_{\mathcal{H}_K}$ can be rewritten as

$$\begin{aligned} \|f\|^2_{\mathcal{H}_K} =< f, f > &= \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j < K(., x_i), K(., x_j) > \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K(x_i, x_j) \\ &= \boldsymbol{\alpha}^t \boldsymbol{K} \boldsymbol{\alpha} \end{aligned} \tag{4.25}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^t$ and $\boldsymbol{K}$ is an n by n symmetric matrix with elements $\boldsymbol{K}_{ij} = K(x_i, x_j)$, called the *kernel matrix*.

Using (4.23)-(4.25) the minimization problem (4.19) can be rewritten as

$$\min_{\boldsymbol{\alpha}} L(\mathbf{y}, \boldsymbol{K}\boldsymbol{\alpha}) + \lambda \boldsymbol{\alpha}^t \boldsymbol{K} \boldsymbol{\alpha} \tag{4.26}$$

which now is a finite-dimensional criterion that can be solved numerically. The fact that the infinite-dimensional problem (4.19) can be reduced to a finite-dimensional problem as in (4.26) is often called *the kernel property*.

In general there are two important choices in regularization using kernels. The choice of the kernel function K and the choice of loss function L. If we consider squared error loss, then (4.19) becomes a penalized least squares problem

$$\min_{\boldsymbol{\alpha}} (\mathbf{y} - \boldsymbol{K}\boldsymbol{\alpha})^t (\mathbf{y} - \boldsymbol{K}\boldsymbol{\alpha}) + \lambda \boldsymbol{\alpha}^t \boldsymbol{K} \boldsymbol{\alpha} \tag{4.27}$$

with solution

$$\hat{\boldsymbol{\alpha}} = (\boldsymbol{K} + \lambda \boldsymbol{I})^{-1} \mathbf{y}. \tag{4.28}$$

This yields

$$\hat{f}(x) = \sum_{i=1}^{n} \hat{\alpha}_i K(x, x_i)$$

or

$$\begin{aligned} \hat{\mathbf{f}} &= \boldsymbol{K}\hat{\boldsymbol{\alpha}} \\ &= \boldsymbol{K}(\boldsymbol{K} + \lambda \boldsymbol{I})^{-1} \mathbf{y} \\ &= (\boldsymbol{I} + \lambda \boldsymbol{K}^{-1})^{-1} \mathbf{y} \end{aligned}$$

This strongly resembles the smoothing spline solution (4.6) and shows that the fitted values again are a linear function of the responses $\mathbf{y}$.

Two possible choices for the kernel function $K(x, y)$ are introduced below.

4.6.1 Polynomial kernels

For $x, y \in \mathbb{R}^d$, the polynomial kernel $K(x, y) = (x^t y + 1)^m$ has $M = \binom{d+m}{m}$ eigen-functions that span the space of polynomials in $\mathbb{R}^d$ with maximal degree equal to m. For example, for degree 2 polynomials in $\mathbb{R}^2$ the kernel becomes

$$\begin{aligned} K(x, y) &= (x^t y + 1)^2 \\ &= 1 + 2x_1 y_1 + 2x_2 y_2 + x_1^2 y_1^2 + x_2^2 y_2^2 + 2x_1 x_2 y_1 y_2 \end{aligned}$$

and $M = 6$ in this case. Let us denote $h(x) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1 x_2)^t$. Note that $K(x, y) = h(x)^t h(y)$. A penalized polynomial fit of degree m is

obtained through

$$\min_{\beta_1,\ldots,\beta_M} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{M} \beta_j h_j(x_i) \right)^2 + \lambda \sum_{j=1}^{M} \beta_j^2$$

Using the polynomial kernel, this problem can be rewritten as in (4.27) and solved easily. Note that the number of eigen-functions soon becomes very large (even larger than n, however, from (4.28) it follows that the solution can be computed in the same computation time order ($O(n^3)$) independent of the degree m of the polynomial.

4.6.2 Radial basis function kernels

The Gaussian kernel $K(x, y) = \exp(-\frac{1}{2}(\|x-y\|/\sigma)^2)$ leads to a solution that is an expansion in Gaussian radial basis functions

$$h_j(x) = \exp(-\frac{1}{2}(\|x - x_j\|/\sigma)^2); \quad j = 1, \ldots, n$$

The radial kernel $K(x, y) = \|x-y\|^2 \log(\|x-y\|)$ leads to a solution using the same basis functions $h_j(x) = \|x - x_j\|^2 \log(\|x - x_j\|)$ as the thin-plate spline.