Richard A. Berk

Statistical Learning from a Regression Perspective

Springer

2

Regression Splines and Regression Smoothers

2.1 Introduction

This chapter launches a more detailed examination of statistical learning within a regression framework. Once again, the focus is on conditional distributions. But now, the mean function for a response variable is central. How does the mean vary with different predictor values? The intent is to begin with procedures that have much the same look and feel as conventional linear regression and gradually move toward procedures that do not.

2.2 Regression Splines

A "spline" is a thin strip of wood that can be easily bent to follow a curved line (Green and Silverman, 1994: 4). Historically, it was used in drafting for drawing smooth curves. Regression splines, a statistical translation of this idea, are a way to represent non-linear, but unknown, mean functions.

Regression splines are not used a great deal in empirical work. As we show, there are usually better ways to proceed. Nevertheless, it is important to consider them, at least briefly. They provide an instructive transition between conventional parametric regression and the kinds of smoothers commonly seen in statistical learning.

2.2.1 Applying a Piecewise Linear Basis

For a piecewise linear basis, the goal is to fit the data with a broken line (or hyperplane) such that at each break point the left-hand edge meets the right-hand edge. When there is a single predictor, for instance, the fit is a set of straight line segments, connected end to end, sometimes called "piecewise linear." Figure 2.1 is a simple illustration using three straight lines joined end to end. There is a response variable represented by y and a predictor represented by x. For now, only the fitted values are shown.

R.A. Berk, *Statistical Learning from a Regression Perspective*,
DOI: 10.1007/978-0-387-77501-2_2,

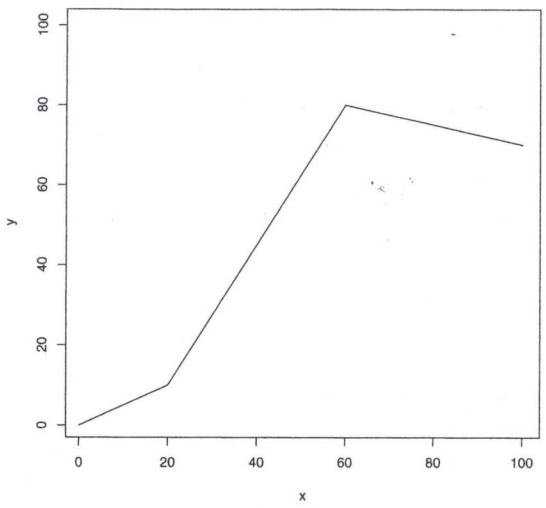

Fig. 2.1. An illustration of piecewise linear function with two knots.

Constructing such a function for the conditional means is straightforward in principle. First, one decides where the break points on x will be. If there is a single predictor, as in this illustration, the break points might be chosen after examining a scatter plot of y on x. When possible, subject matter expertise should also be used to help determine the break points. For example, x might be years, and then the break points might be determined by specific historical events. Thus, y might be a measure of a river's biodiversity, and x might be time in months, with one breakpoint representing the removal of a major dam and another breakpoint representing a toxic chemical spill. Let the break points here be defined at $x = a$ and $x = b$ (with $b > a$). In Figure 2.1, $a = 20$ and $b = 60$. Such break points are often called "knots."

The second step is to define two indicator variables to represent the break points. Here, the first (I_a) is equal to 1 if x is greater 20 and equal to 0 otherwise. The second (I_b) is equal to 1 if x is greater than 60 and equal to 0 otherwise. We let x_a be the value of x at the first break point, and x_b be the value of x at the second break point.

The third step is to define the mean function. Because at this point description is the primary goal, the conditional mean of y is represented by $\bar{y}|x$ rather than by $E(y|x)$. The latter implies that Y is a random variable. For now, it does not matter whether Y is a random variable. Then,

$$\bar{y}|x = \beta_0 + \beta_1 x + \beta_2(x - x_a)I_a + \beta_3(x - x_b)I_b. \tag{2.1}$$

Looking back at Equation 1.15, it is apparent that there are four transformations of X, $h_m(X)$s, in which the first function of x is a constant.

The mean function for x less than a is

$$\bar{y}|x = \beta_0 + \beta_1 x. \tag{2.2}$$

In Figure 2.1, β_0 is zero, and β_1 is positive.

For values of x greater than a but smaller than b, the mean function becomes

$$\bar{y}|x = (\beta_0 - \beta_2 x_a) + (\beta_1 + \beta_2)x. \tag{2.3}$$

For a positive β_1 and β_2, the line beyond $x = a$ is steeper because the slope is $(\beta_1 + \beta_2)$. The intercept is lower because of the second term in $(\beta_0 - \beta_2 x_a)$. This too is consistent with Figure 2.1. If β_2 is negative, the reverse would apply.

For values of x greater than b, the mean function becomes,

$$\bar{y}|x = (\beta_0 - \beta_2 x_a - \beta_3 x_b) + (\beta_1 + \beta_2 + \beta_3)x. \tag{2.4}$$

For these values of x, the slope is altered by adding β_3 to the slope of the previous line segment. The intercept is altered by subtracting $\beta_3 x_b$. The sign and magnitude of β_3 determine if the slope of the new line segment is positive or negative and how steep it is. The intercept will shift accordingly. In Figure 2.1, β_3 is negative and large enough to make the slope negative. The intercept is increased substantially.

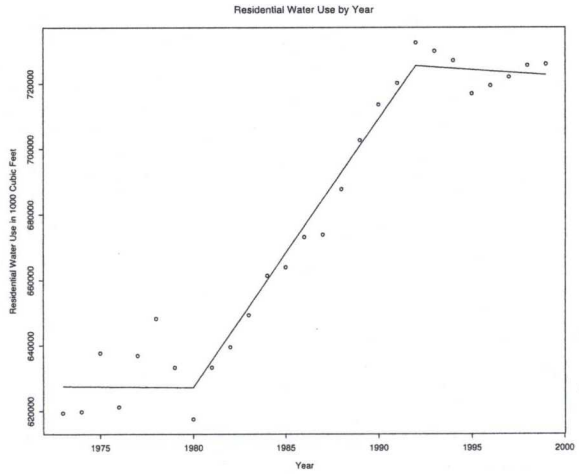

Fig. 2.2. A piecewise linear basis applied to water use by year.

Figure 2.2 shows a three-piece linear regression spline applied to water use data from Tokyo over a period of 27 years. Residential water use in 1000s of

cubic feet is on the vertical axis. Year is on the horizontal axis. The locations of the break points were chosen after inspecting the scatterplot, with some reliance on subject matter expertise about residential water use in Japan.

It is clear that water use was flat until about 1980, then increased linearly until about 1996, and then flattened out again. The first break point may correspond to a transition toward much faster economic and population growth. The second break point may correspond to the introduction of more water-efficient technology. But why the transitions are so sharp is mysterious. One possibility is that the break points correspond in part to changes in how the water use data were collected or reported.

It is perhaps most common to see regression splines fit to data in which time is used as the sole predictor. The end-to-end connections between line segments lead naturally to processes that unfold over time. The line segment on the right side of a knot begins where the line segment on the left side of the knot ends. But there is nothing about linear regression splines requiring that time be a predictor. For example, the response could be crop production per acre and the sole predictor could be the amount of phosphorus fertilizer applied to the soil. Crop production might increase in approximately a linear fashion until an excess of phosphorus caused other kinds of nutritional difficulties. At that point, crop yields might decline in roughly a linear manner.

Fitting line segments to data provides an example of "smoothing" a scatterplot, or applying a "smoother." The line segments are used in place of the data to characterize how x and y are related. The intent is to highlight key features of any association while removing unimportant details. This can often be accomplished by constructing fitted values in a manner that makes them more homogeneous than the set of conditional means of y computed for each unique value of x.

Imagine a scatterplot in which the number of observations was large enough so that for each value of x there were at least several values of y. One could compute the mean of y for each x-value. If one then drew a straight line between each of the adjacent conditional means, the resulting smoother would be an interpolation of the conditional means and as rough as possible. At the other extreme, imposing a single linear fit on all of the means at once would produce the smoothest smoother possible. Figure 2.2 falls somewhere in between. How to think about the degree of smoothness more formally is addressed later.

For a piecewise linear basis, one can simply compute functions such as Equation 2.1 with ordinary least squares. With the regression coefficients in hand, fitted values are easily constructed. Indeed, many software packages compute and store fitted values on a routine basis. Also widely available are procedures to construct the matrix of regressors, although it is not hard to do so one term at a time using common transformation capabilities. For example, the library *spline* has a procedure bs() that constructs a B-spline basis (discussed later) that can be easily used to represent the predictor matrix for piecewise linear regression.

In contrast to most applications of conventional linear regression, there would typically be little interest in the regression coefficients themselves; they are but a means to an end. The point of the exercise is to superimpose the fitted values on a scatterplot so that the relationship between y and x can be more effectively visualized. As we show later, and as was briefly anticipated in the last chapter, model selection will not necessarily be the same as regressor selection.

2.2.2 Polynomial Regression Splines

Smoothing a scatterplot using a piecewise linear basis has the great advantage of simplicity in concept and implementation. And by increasing the number of break points, very complicated relationships can be approximated. However, in most applications there are good reasons to believe that the underlying relationship is much smoother than can be easily represented with a set of straight line segments.

Greater continuity can be achieved by using polynomials in x for each segment. Cubic functions of x are a popular choice because they strike a nice balance between flexibility and complexity. When used to construct regression splines, the fit is sometimes called "piecewise cubic." The cubic polynomial serves as a "truncated power series basis" in x.

Unfortunately, simply joining polynomial segments end to end is unlikely to result in a visually appealing fit where the polynomial segments meet. The slopes of the two lines will often appear to change abruptly even when that is inconsistent with the data. Far better visual continuity usually can be achieved by constraining the first and second derivatives on either side of each break point to be the same.

Putting this all together, one can generalize the piecewise linear approach and impose the continuity requirements. Suppose there are K interior break points, usually called "interior knots." These are located at $\xi_1 < \cdots < \xi_K$ with two boundary knots added at ξ_0 and ξ_{K+1}. Then, one can use piecewise cubic polynomials in the following regression formulation,

$$\bar{y}|x = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \sum_{j=1}^{K} \theta_j (x - x_j)_+^3, \tag{2.5}$$

where the "+" indicates the positive values from the expression inside the parentheses, and there are $K + 4$ parameters whose values need to be computed. This leads to a conventional regression formulation with a matrix of predictor terms having $K + 4$ columns and N rows. Each row would have the corresponding values of the piecewise cubic polynomial function evaluated at the single value of x for that case. There is still only a single predictor, but now there are $K + 4$ basis functions.

The output for the far-right term in Equation 2.5 may not be apparent at first. Suppose the values of the predictor are arranged in order from low

to high. For example, $x = [1, 2, 4, 5, 7, 8]$. Suppose also that x_j is located at an x-value of 4. Then, $(x - x_j)_+^3 = [0, 0, 0, 1, 27, 64]$. The knot-value of 4 is subtracted from each value of x, the negative numbers set to 0, and the others cubed. All that changes from knot to knot is the value of x_j that is subtracted. There are K such knots and K such terms in the regression model.

Figure 2.3 shows the water use data again, but with a piecewise cubic polynomial overlaid that imposes the two continuity constraints. The fit looks quite good to the eye and captures about 95% of the variance in water use. But, in all fairness, the scatterplot did not present a great challenge. The point is to compare Figure 2.2 to Figure 2.3 and note the visual difference. The linear piecewise fit also accounted for about 95% of the variance. Which plot would be more instructive in practice would depend on the use to be made of the fitted values and on prior information about what a sensible $f(X)$ might be. The regression coefficients ranged widely and, as to be expected, did not by themselves add any useful information. The story was primarily in the fitted values.

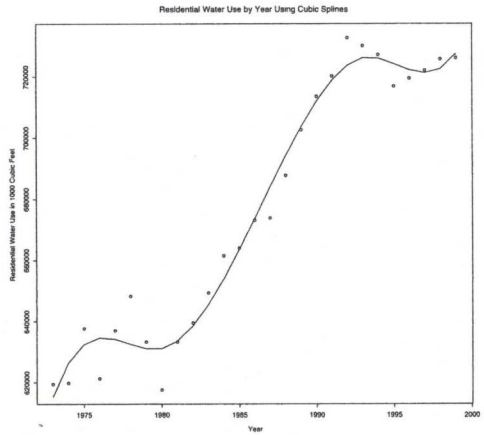

Fig. 2.3. A piecewise cubic polynomial applied to water use by year.

2.2.3 Natural Cubic Splines

Fitted values for piecewise cubic polynomials near the boundaries of x can be unstable because they fall at the ends of polynomial line segments where there are no continuity constraints, and where there may be little data. By "unstable" one means that a very few observations, which could vary over random samples from the same population, produce substantially different fitted values near the boundaries of x. As a result, the plot of the fitted values near the boundaries could look rather different from sample to sample.

Sometimes, constraints for behavior at the boundaries are added to increase stability. One common constraint imposes linearity on the fitted values beyond the boundaries of x. This introduces a bit of bias because it is very unlikely that if data beyond the current boundaries were available, their relationship with the response would be linear. However, the added stability is often worth it. When these constraints are added, the result is a "natural cubic spline."

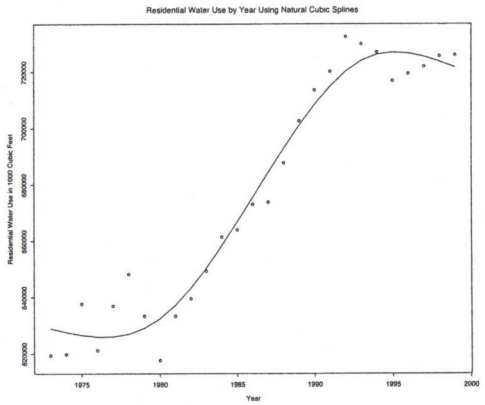

Fig. 2.4. Natural cubic regression splines applied to water use by year.

Figure 2.4 shows again a plot of the water use data on year, but now with a smoother constructed from natural cubic splines. One can see that the fitted values near the boundaries of x are somewhat different from the fitted values near the boundaries of x in Figure 2.3. The fitted values in Figure 2.4 are smoother, which is the desired result. There is one less bend near both boundaries. More generally, how one can formulate the boundary constraints is discussed in Hastie et al. (2001: Section 5.2.1).

The option of including extra constraints to help stabilize the fit provides an example of the bias–variance tradeoff. This is a topic to which we return many times in the pages ahead. For now, an informal overview for natural cubic splines may be useful.

The bias–variance tradeoff addresses some important properties of fitted values when the data are a random sample from a population or a realization of a stochastic process. Bias and variance refer to what can happen over a limitless number of hypothetical, independent, random samples or realizations; the context is the usual frequentist thought experiment. Therefore, the bias–variance tradeoff is only directly relevant when statistical inference is on the table and does not formally provide much insight when summary statistics are being used solely for description.

When constraints are imposed on a fitting process to make the fitted values less variable, bias in the fitted values can be introduced. The means of the fitted values over many samples or realizations will often be farther from the true conditional means of the response variable, which are the values one wants to estimate. However, in repeated independent random samples, or independent realizations of the data, the fitted values will vary less. When the fit is smoother, each fitted value is constructed, in effect, from a larger number of y-values. This increases stability in the same way that larger samples in general provide estimates with a smaller variance. Conversely, but using the same reasoning, a rougher fit can imply less bias but more variance over repeated samples or realizations. A tradeoff naturally follows.

Ideally, just the right amount of bias can be combined with just the right amount of variance so that over repeated random samples or realizations, the fitted values would be on the average as close to true response variable values as possible. "Close" can be operationalized in several ways, but it is often desirable to work with a test sample and then try to minimize the mean of the squared deviations between the fitted values and the observed values of the response variable (i.e., the mean squared error in a test sample).

For piecewise cubic polynomials and natural cubic splines, the degree of smoothness is primarily a function of the number of interior knots. In practice, the smaller the number of knots, the smoother are the fitted values. A smaller number of knots means that there are more constraints on the pattern of fitted values because there are fewer end-to-end, cubic line segments used in the fitting process. Consequently, less provision is made for potential twists and turns.

But placement matters too. Ideally, knots should be located where it is thought that the $f(X)$ is changing most rapidly. In some cases, inspection of the data, coupled with subject matter knowledge, can be used to determine the number and placement of knots. The water use data just considered were analyzed in this manner.

Alternatively, the number and placement of knots can be approached as a model selection problem. Any of the fit statistics discussed in the last chapter, such as the GCV, can be used to determine the number of knots, given a set of candidate locations. The number of knots translates into a penalty for the number of regression parameters whose values are being estimated from the data. The penalty increases with the number of knots, just as the penalty would normally increase with the number of regression parameters whose values were not known a priori. Then, the goal is to choose the knot number that minimizes the fit statistic. Knot selection is essentially regressor selection. In other words, a set of potential knots is specified, and fit statistics are used to determine which knots are really needed.

The fit statistics are largely silent on where to place the knots. Two models with the same number of knots can produce very different fitted values if the placement of the knots substantially differs. Two models with a very different number of knots may fit the data about the same, depending on

where the knots are placed. Moreover, absent subject matter information, knot placement has been long known to be a difficult technical problem, especially when there is more than one predictor (de Boors, 2001). The fitted values are related to where the knots are placed in a very complicated manner. Fortunately, methods discussed later sidestep the knot location problem.

Even if a good case for candidate knot locations can be made, one must be careful about taking any of the fit measures too literally. First, there will often be several models with rather similar values, whatever the kind of fit statistic used. Then, selecting a single model as "best" using the fit measure alone may amplify a small numerical superiority into a large difference in the results, especially if the goal is to interpret how the predictors are related to the response. Some jokingly call this "specious specificity." Second, the same issues can arise when comparing the models selected by the different kinds of fit statistics. The impact of very small differences in fit can lead to very large difference in the results. Third, one must be a very careful to not let small differences in the fit statistics automatically trump subject matter knowledge. The risk is arriving at a model that may be difficult to interpret, or effectively worthless.

In summary, for regression splines of the sort just discussed, there is no straightforward way to arrive at the best tradeoff between the bias and the variance because there is no straightforward way to determine knot location. A key implication is that it is very difficult to arrive at a model that is demonstrably the best. Fortunately, there are other approaches to smoothing that are more promising.

2.2.4 *B*-Splines

In practice, data analyses using piecewise cubic polynominals and natural cubic splines are rarely constructed directly from polynomials of x. They are commonly constructed using a B-spline basis, largely because of computational convenience. A serious discussion of B-splines would take us far afield and accessible summaries can be found in Gifi (1990) and Hastie et al. (2001). Nevertheless several observations are worth making.

The goal is to construct transformations of x that allow for a cubic piecewise fit but that have nice numerical properties and are easy to manipulate. B-splines do well by all three criteria. They are computed in a recursive manner from very simple functions to more complex ones, and consistent with the approach to basis functions taken here, can be represented as a linear basis expansion.

For a series of knots, which usually include several beyond the upper and lower boundaries of x, indicator variables are defined for each region marked off by the knots. If a value of x falls within a given region, the indicator variable for that region is coded 1, and coded 0 otherwise. For example, if there is a knot at an x-value of 2 and the next knot at an x-value of 3, the x-values between them form a region with its own indicator variable coded 1

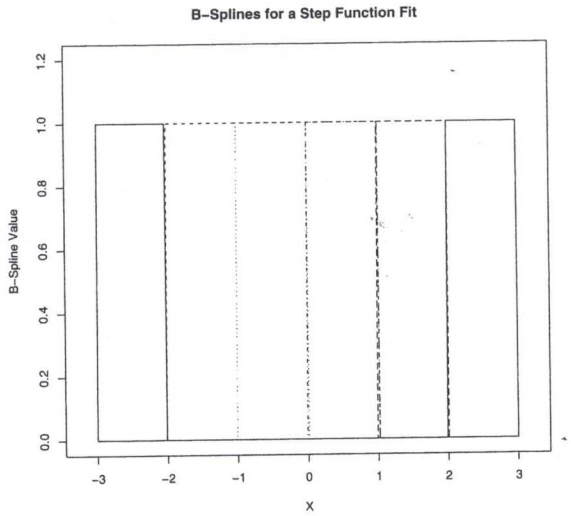

Fig. 2.5. Degree zero B-splines.

if the value of x falls in that region (e.g., $x = 2.3$), and coded 0 otherwise. The result is a set of indicator variables, with values of 1 or 0, for each region. These indicator variables define a set of degree zero B-splines.

Figure 2.5 is an illustration with interior knots at -2, -1, 0, 1, and 2. Using the indicator variables as regressors will produce a step function when y is regressed on x; they are the basis for a step function fit. The steps will be located at the knots.

Next a transformation can be applied to the degree zero B-splines. (See Hastie et al., 2001: 160–163). The result is a set of degree one B-splines. Figure 2.6 shows the set of degree one B-splines derived from the indicator variables shown in Figure 2.5. The triangular shape is characteristic of degree one B-splines, and implies that the values for each spline are no longer just 0 or 1, but proportions in between as well.

Degree one B-splines are the basis for linear piecewise fits. Here, the regressor matrix includes eight columns whose values appear in Figure 2.6. The content of each column is the B-spline values for each value of x. Regressing a response on that matrix will produce a linear piecewise fit with knots at -2, -1, 0, 1, and 2.

A transformation of the same form can now be applied to the degree one B-splines. This leads to a set of degree two B-splines that are the basis for a quadratic piecewise fit. For this illustration, there is now a matrix with nine columns that can serve as a regressor matrix. The set of such B-splines is shown in Figure 2.7 and as before, the shapes are characteristic.

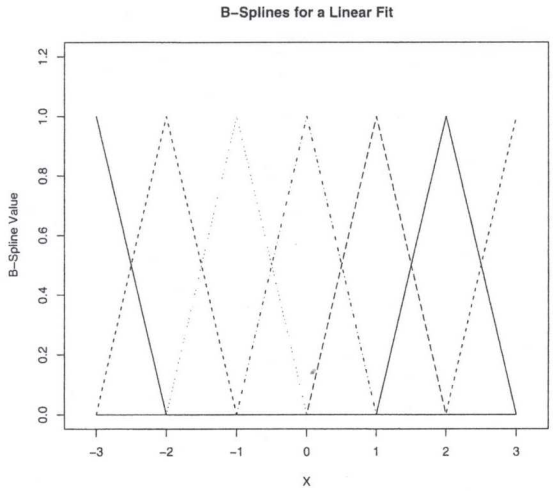

Fig. 2.6. Degree one B-splines.

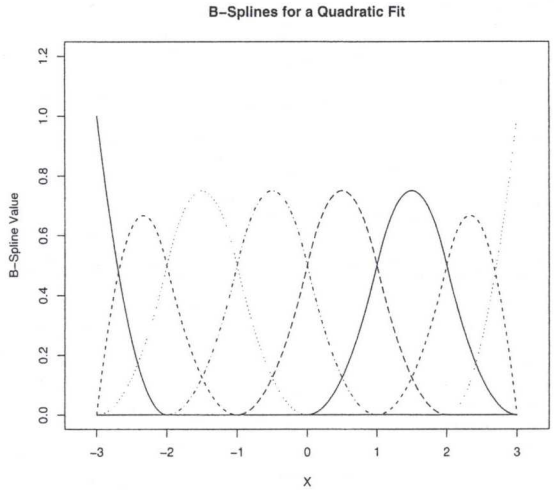

Fig. 2.7. Degree two B-splines.

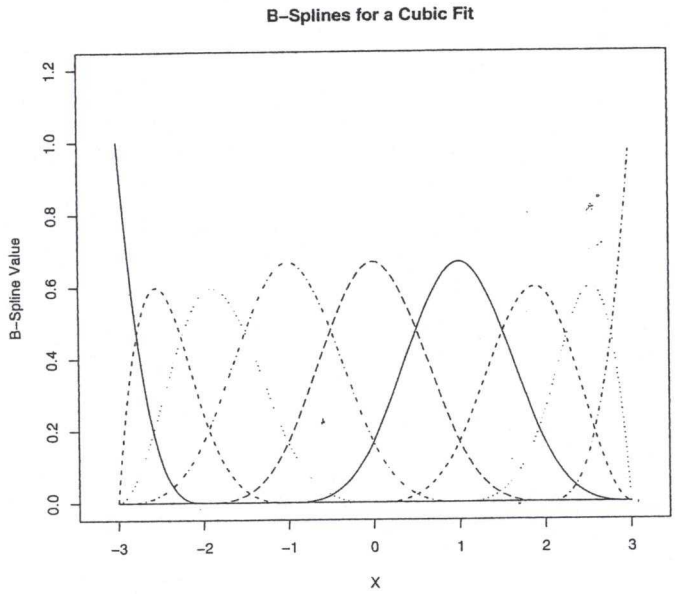

Fig. 2.8. Degree three B-splines.

The same kind of transformation can then be applied to the degree two B-splines. The result is a set of degree three B-splines that are the basis for a cubic piecewise fit. Figure 2.8 shows the set of degree three splines, whose shapes are, once again, characteristic. They can be used to construct a regressor matrix with nine columns.

All splines are a linear combinations of B-splines; B-splines are a basis for the space of all splines. They are also a well-conditioned basis because they are fairly close to orthogonal, and they can be computed in a stable and efficient manner. For our purposes, the main point is that B-splines are a computational device used to construct cubic piecewise fitted values. When such smoothers are employed, B-splines are doing the work behind the scenes.

2.3 Penalized Smoothing

The placement of knots, the number of knots, and the degree of the polynomial are subject to manipulation by a data analyst. All three can be used to construct a highly flexible fitting function that will track the data well. Because a good fit is typically considered desirable, there is sufficient reason in practice to worry about overfitting. The pull toward constructing a good fit can be very strong.

The fit statistics considered earlier can provide some protection against overfitting. They can help compensate for the amount of flexibility built into a given fitting function. However, they function indirectly. They are applied

the fitted values with the goal of reducing their variance. It is also important to look beneath the computer output and understand how the statistical inference was undertaken.

Third, overfitting can be a serious problem. The results from the data examined may not generalize well to other random samples from the same population. We consider overfitting in depth in later chapters. For now, *caveat emptor.*

Finally, for a wide range of problems, there are statistical learning techniques that arguably perform better than the procedures discussed in this chapter. They can fit the data better, are less subject to overfitting, and permit a wider range of information to be brought to bear. One price, however, is that the links to conventional regression analysis become far more tenuous. In the next chapter, we start down this path.

Exercises

Problem Set 1: Smoothers with a Single Predictor

1. Load the dataset called airquality using the command data(airquality). Attach the data with the command attach(airquality). Use gam() from the *gam* library with Ozone as the response and Temp as the sole predictor. Estimate the following three models assigning the output of each to its own name (e.g., output1 for the first model).

```
gam(Ozone ~ Temp)
gam(Ozone ~ as.factor(Temp) )
gam(Ozone ~ s(Temp) )
```

   The first model is the smoothest model possible. Why is that? The second model is the roughest model possible. Why is that? The third model is a compromise between the two in which the degree of smoothing is determined by the GCV statistic. (See the gam() documentation followed by the smoothing spline documentation.)

   For each model, examine the numerical output and plot the fitted values against the predictor. For example, if the results of the first model are assigned to the name "output1," use plot.gam (output1, residuals=TRUE).

   Which model has the best fit judging by the residual deviance? Which model has the best fit judging by the AIC? Why might the choice of the best model differ depending on which measure of fit is used? Which model seems to be most useful judging by the plots? Why is that?

2. Overlay a lowess smooth on a scatterplot with the variable Ozone on the vertical axis and the variable Temp on the horizontal axis. Vary three tuning parameters: span: .25, .50, .75; degree: 0, 1, 2; family as Gaussian or symmetric. Describe what happens to the fitted values as each tuning parameter is varied. Which tuning parameter seems to matter most?

3. The relationship between temperature and ozone concentrations should be positive and monotonic. From the question above, select a single set of tuning parameter values that produces a fit you like best. Explain why you like that fit best. If there are several sets of fitted values you like about equally, explain what it is about these fitted values that you like.

4. For the overlay of the fitted values you like best (or select a set from among those you like best) describe how temperature is related to ozone concentrations.

5. One can address the stability of the fitted values using the bootstrap percentile method. Load the library *simpleboot.* The procedure first requires that you run lowess and then you apply the bootstrap. For example: assign loess(Ozone ~ Temp) to a name such as "smooth." Then assign loess.boot (smooth) to a name such as"bo." Finally use plot(bo). The point-by-point interval is constructed by taking the standard deviations of the fitted values for each point over bootstrap samples, multiplying each by two, and adding that product to the fitted values at each point and subtracting that product from the fitted values at each point.

   For what values of temperature does the instability appear to be about the largest? For what values of temperature does the instability appear to be the smallest? What in the data accounts for these differences?

Problem Set 2: Smoothers with Two Predictors

1. From the library *assist* load the dataset TXtemp. Load the library *gam.* With mmtemp as the response and longitude and latitude as the predictors, apply gam(). Construct the fitted values using the sum of a 1-D lowess smooth of longitude and a 1-D smooth of latitude. Try several different values for the degrees of freedom of each. Try different values for the degree of the polynomial. You can learn how to vary these tuning parameters with help(gam) and help(lo). Use the summary() command to examine the output and the plot.gam() to plot the two partial response functions. To get both plots on the same page use par(mfrow=c(1,1)). How are longitude and latitude related to temperature?

2. Repeat the analysis in 1, but now construct the fitted values using a single 2-D smoother of longitude and latitude together. Again, try several different values for the span and degree of the polynomial. Examine the tabular output with summary() and the plot using plot.gam(). How do these results compare to those using two 1-D predictor smooths?

Problem Set 3: Smoothers with More Than Two Predictors

1. Now build an additive model for mmtemp with the predictors longitude, latitude, year, and month. Use a lowess smooth for each. Try different spans and polynomial degrees. Again use the summary() and plot.gam() command. To get all four graphs on the same page use par(mfrow=c(2,2)). How is temperature related to each of the four predictors?

2. Repeat the analysis done for 1, but with penalized smoothing splines. The operator in front of each predictor is now s and not lo. Read the help documentation for gam(), and s(). How is temperature related to each of the four predictors? How do the conclusions from 1 compare with the conclusions drawn here? Why?

Problem Set 4: Smoothers with a Binary Response Variable

1. From the *car* library, load the dataset Mroz. Using the glm(), regress labor force participation on age, income, and the log of wages. From the library *gam*, use gam() to repeat the analysis, smoothing each of the predictors. Note that labor force participation is a binary variable. Compare and contrast your conclusions from the two sets of results. Which procedure seems more appropriate here? Why?