1
Introduction

This is a book about P-splines, our favorite smoother, and the best one you can find. We may be a bit biased, because we invented them, but we will try to convince you by showing many applications. We also provide the necessary theoretical details that are needed for a complete understanding. We have practical data analysts in mind, who want to extend their statistical toolbox in new directions, and we rely on many illustrations to stimulate an intuitive understanding of P-splines.

Why are P-splines so good? They start from (generalized linear) regression, the strongest workhorse of statistics. The estimation of curves or surfaces is our goal, so the building blocks of the regression are B-splines. We combine them with a less-familiar ingredient, a penalty, giving P-splines their name. A penalty is a device that, more or less gently, forces the fit of a statistical model in a desired direction, in our case smoothness. Figure 1.1 illustrates the core idea behind P-splines: (i) the smooth fit is the sum of many B-spline basis functions, each having local support and scaled by its own coefficient, and (ii) the penalty enforces further smoothness by discouraging adjacent B-spline coefficients from having values that are too different from each other.

Neither B-splines nor penalties are new ideas, and each has found many applications. B-splines are smoothly joining polynomial segments (of a chosen degree), and they are completely defined by the "knots," the places where the segments meet. You can try to optimize the number and the positions of the knots when fitting a curve to data, but this is a tricky nonlinear problem.

A step forward is to use evenly spaced knots, leaving only the number of B-splines as a design parameter. New problems now arise, especially when the locations of the data points are not evenly distributed. In such a case, some of the splines may have little or no support, meaning that coefficient estimates will be unstable or even impossible to obtain.

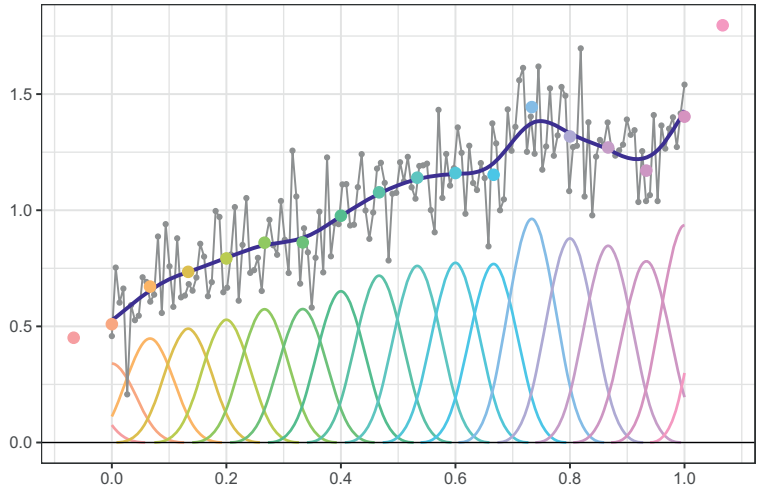

Figure 1.1 The core idea of P-splines: a sum of B-spline basis functions, with gradually changing heights. The small connected gray dots show simulated data. The blue curve shows the P-spline fit, and the large dots the B-spline coefficients (they have the same colors as the splines). `R` code in `f-ps-show.R`

O'Sullivan (1986) eliminated the instability problem by combining a relatively large number of B-splines with a roughness penalty. He was inspired by the classical smoothing spline that uses the integral of the squared second derivative of the fitted curve. Exploiting the fact that the curve consists of polynomial segments, he derived a matrix that forms the core of the penalty.

Let B be a regression matrix that contains the B-spline basis functions, and denote α as the corresponding vector of coefficients. The fitted values are $\mu = B\alpha$, and the objective function to be minimized is

$$S = ||y - B\alpha||^2 + \lambda \alpha' P \alpha. \tag{1.1}$$

In the first term we recognize the sum of squares of residuals, familiar from linear regression. The second term measures the roughness of the fitted curve, expressed in the B-spline coefficients. O'Sullivan derived the analytic form of the matrix P; we do not need it in this book, so we skip the details. The parameter λ sets the balance between the deviations from the data and the roughness of the fit. Increasing λ gives the penalty more influence, resulting in a smoother result.

The only innovation of P-splines is a small modification of the penalty: it is based directly on (higher-order) differences of the coefficient vector α, avoiding integrals of squared derivatives. This has several advantages. It is almost trivial

to compute the penalty term in (1.1), as P is replaced by $D'D$, where D is a matrix such that $D\alpha$ forms dth-order differences of α. The matrix D can be obtained with one line of code in `R` or Matlab, for any value of d. One is free to choose any order of the differences, without complications. In principle, O'Sullivan's approach allows higher-order derivatives, but the degree of the B-splines must be high enough, or the derivatives disappear (and with them the penalty). In P-splines, the degree of the B-splines and the order of the penalty can both be chosen freely and independent of each other. In some applications it makes perfect sense to combine, say, piecewise-constant B-splines with third-order differences in the penalty.

We do not pay a price for these attractive simplifications. A look at the Contents should convince you that there are many and diverse applications of P-splines. You will also learn that they are easy to implement. Along the way, you will discover some surprising results and the amazing power of penalties.

The Road Ahead. We lay the groundwork in Chapter 2, presenting (B-spline) regression bases and penalties while illustrating our first applications. In this chapter we also explore responses with non-normal distributions, like counts and binary data, adapting the setting of generalized linear models. We explore interpolation and extrapolation and limits of heavy smoothing. We also present the Whittaker smoother, which can be viewed as a bare-bones form of P-splines. It is suitable for data on a uniform grid when we only are interested in smoothed values on that grid. The B-spline basis matrix is now simplified to the identity matrix. The Whittaker smoother is ideal for studying essential properties of penalties, avoiding details of a B-spline basis. We will employ it several times.

We introduce an important theoretical and practical tool: the effective (model) dimension. It tells us how complex a model is; it will appear frequently in many chapters, especially when we estimate variances.

Next, in Chapter 3, we investigate how to obtain a reasonable value for the penalty tuning parameter λ in (1.1); one option is to minimize data-driven criteria, e.g., cross-validation or AIC. Density estimation by smoothing of histograms gets much attention here. It is a problem of great practical relevance and a good vehicle for showcasing various approaches to optimal smoothing. We also introduce automatic tuning of P-splines with mixed models, showing how penalty parameters can be interpreted as ratios of variances. The same is true for Bayesian P-splines.

We have the tools for automatic selection of smoothing parameter, but we should not use them blindly. Autocorrelation, overdispersion, and digit preference can do serious harm, and we discuss how to handle them properly.

In Chapter 4, we take our first steps into the field of multidimensional smoothing, starting with the generalized additive model. As the name suggests, it neglects interactions and models a response (or its linear predictor) as a sum of smooth functions. B-spline bases are very attractive, as they can simply be chained into one large design matrix, while the penalties can be neatly placed into an appropriate block-diagonal matrix. With this construction, the strong properties of P-splines for simple models automatically become available for additive models. Next, we develop varying coefficient models, which allow standard regression coefficients to vary over another variable, e.g., time, depth, or age.

To fully allow for nonadditive features, like interactions between the explanatory variables, we use tensor products of B-splines in conjunction with proper multidimensional difference penalties.

In the first four chapters, when smoothing a cloud of (x, y) pairs, our interest is only in the expected value of y, conditional on x. Chapter 5 sets a more ambitious goal: estimating sets of curves that characterize the conditional distribution of y. An example are quantile curves, corresponding to a chosen set of probabilities. Although less familiar, expectile curves look better, are easier to compute, and are more efficient than quantile curves. We also pay attention to the GAMLSS framework (Rigby and Stasinopoulos, 2005). It fits models where the distribution of the response variable can be non-normal, and the parameters for variance and skewness of that distribution are modeled as smooth functions of explanatory variables.

Chapter 6 is devoted to the penalized composite link model (PCLM) for counts. This model has received little attention in the literature, but it is a natural candidate when observations are generated by distortions of underlying smooth distributions. Examples include grouping in coarse intervals, digit preference, and overdispersed discrete distributions.

In Chapter 7, P-splines are used for regression on signals, such as spectra and time series. The number of coefficients is large, but they are ordered. This allows the use of a roughness penalty on the coefficients for regularization. Modeling these coefficients by B-splines gives a strong reduction of the system of equations to be solved. Multidimensional signal regression, single-index models, and other extensions are also presented.

The final Chapter 8 presents special applications, like circular B-spline bases, and the separation of signals into components. We also discuss specialized penalties for shape constraints, piece-wise constant smoothing, and adaptive smoothing. Survival and mortality smoothing are also given proper attention here.

The appendices fill in a variety of details, starting with *P-splines for the impatient*, followed by a detailed comparison of P-splines to other popular smoothers (Appendices A and B, respectively). Computational details for the construction of B-splines and efficient computation of (sparse) B-spline bases and stable penalized regression are presented in Appendix C. For large data sets, a straightforward implementation of tensor product P-splines puts very high demands on memory use and computation time. Appendix D presents a very efficient alternative, the so-called array algorithm. P-splines can be written as a mixed model, and variance estimation can be used for optimal smoothing. Appendix E discusses the mixed model equations in detail and shows how they can be simplified. A short Appendix F also discusses standard errors.

The final Appendix G outlines our website. Here we provide documentation – with descriptions and scripts – for every figure in the book. In this way, readers can reproduce our graphs and investigate details of the calculations. We hope that they will explore variations by modifying our code, and applying it in their own work. Our `R` package (`JOPS`), with support functions and help files, is available on the website – hopefully, it will find its way into the CRAN repository. It provides core functions for working with P-splines and also contains all data sets that occur in the book. As a teaching tool, a special section of the website provides programs for playing with P-splines interactively.

We have a remark about terminology. When we coined the name P-splines in our 1996 paper (Eilers and Marx, 1996), we thought it was clear what it stood for: many evenly spaced B-splines, combined with a difference penalty. We did not own a trademark, but we hoped to establish a clear name for a clear product. Surprisingly, many workers in the field called various types of penalized splines "P-splines." For example, Yu and Ruppert (2002), Ruppert et al. (2003), and Jarrow et al. (2004) referred to truncated power functions (TPF) with unequally spaced knots as P-splines. This is unfortunate, as we have exposed the disadvantages of penalized TPF in several places in our publications. We hope this book will set the record straight. Historical overviews of P-splines can be found in Eilers and Marx (2010) and Eilers et al. (2015), while our very first writings on the subject go back to Eilers and Marx (1992).

A final remark, about notation. We have not tried to achieve strict uniformity between chapters. That would demand too many symbols with too many decorations like tildes, subscripts, and superscripts. We take a more relaxed approach: notation is consistent within, but not always across, chapters.

2

Bases, Penalties, and Likelihoods

P-splines combine two simple ideas: regression on (many) B-splines and a difference penalty on their coefficients. The B-splines are local functions, each of them covering only a small part of the x-axis. They can give a very flexible fit to data. To keep a fitted curve from getting too flexible, the penalty comes in. It lets adjacent coefficients "hold hands," encouraging a smooth fit. Once the number of B-splines has been set, a single parameter, λ, tunes smoothness.

P-splines have many interesting and useful properties. Interpolation and extrapolation of the fitted curve are automatic. Standard errors and derivatives of the fitted curve are easy to compute.

With a heavy penalty a polynomial curve fit is obtained, creating a bridge between semi-parametric and classic parametric models. But in essence, P-splines are parametric models. The coefficients have a very clear interpretation and can be presented graphically as the skeleton of the fitted curve.

P-splines are grounded in linear regression. Extensions to generalized linear regression are straightforward through penalized likelihood. Counts and binomial data can be handled in an elegant way.

The penalty makes the number of B-splines irrelevant, as long as it is large enough. With many of them, say 10 to 50, the number of coefficients in the model is moderately large. Yet, as will be shown, the effective dimension of the model will be (much) smaller than this number, depending on the amount of smoothing.

2.1 Linear and Polynomial Regression

Consider a scatterplot of data pairs (x_i, y_i), $i = 1 : m$. Figure 2.1 displays $m = 111$ daily readings of wind speed (mph) (x-axis) and the maximum of daily ozone (ppm) (y-axis) in New York City (data set `airquality` in `R`).

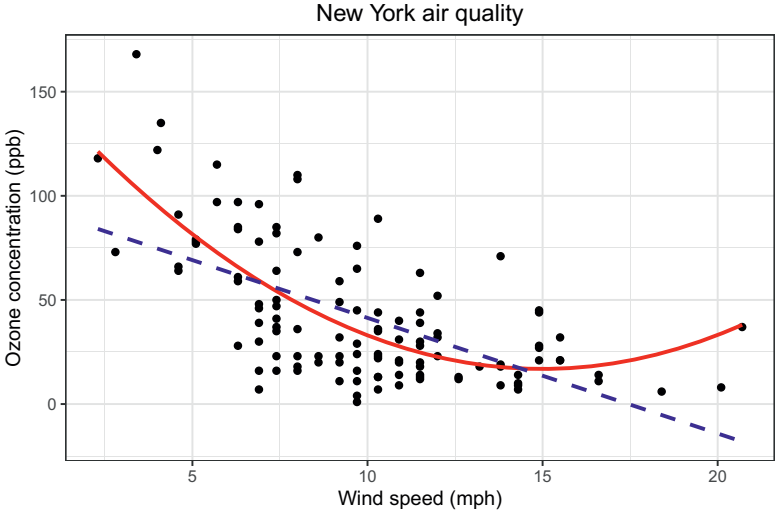

Figure 2.1 Air pollution in New York: scatterplot of daily maximum ozone concentrations and wind speed. Least squares linear (blue broken line) and quadratic (solid red curve) fits. R code in `f-air-wind.R`

The straight blue broken line shows the linear least squares fit, while the solid red line shows the quadratic fit.

The formula for a quadratic curve is $\mu_i = \alpha_0 + \alpha_1 x_i + \alpha_2 x_i^2$. The vector $\alpha = [\alpha_0\ \alpha_1\ \alpha_2]'$ that gives the "best" fit to the data is found by minimizing the least squares objective

$$S = \sum_{i=1}^{m} (y_i - \mu_i)^2 = \sum_{i=1}^{m} (y_i - \alpha_0 - \alpha_1 x_i - \alpha_2 x_i^2)^2.$$

It is easier to work in matrix notation. For the above quadratic curve, we express the m by 3 regressor matrix B, the 3 by 1 unknown parameter vector α, and the m by 1 mean vector μ as

$$B = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_m & x_m^2 \end{bmatrix} \quad \alpha = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{bmatrix} \quad \mu = B\alpha, \tag{2.1}$$

respectively. Now the least squares solution minimizes the objective

$$S = ||y - B\alpha||^2, \tag{2.2}$$

defined as the squared-norm $(y - B\alpha)'(y - B\alpha)$. This leads to the normal equations $B'B\alpha = B'y$ or $\hat{\alpha} = (B'B)^{-1}B'y$.

A more general setting introduces a vector of weights, w. They can reflect known precisions of the data, or w can contain zeros and ones, where a zero indicates a missing y. Using such weights, missingness can be introduced deliberately, e.g., to (temporarily) exclude selected observations. It is more convenient than excluding rows of B and y. With $W = \text{diag}(w)$, we get

$$S = (y - B\alpha)'W(y - B\alpha), \tag{2.3}$$

and the normal equations $B'WB\hat{\alpha} = B'Wy$, with solution $\hat{\alpha} = (B'WB)^{-1}B'Wy$.

The regression scheme can be extended to higher powers of x by adding columns in B with third, fourth, or higher powers. In theory, the computation of $\hat{\alpha}$ does not change, but in practice one has to center and scale x to avoid numerical instabilities. Modern regression software overcomes this issue by using specialized algorithms, like the QR decomposition (Wood, 2017). We will not get into the details of the QR decomposition here. After showing that high-degree polynomial curve fits have serious and fundamental problems, we will discard them as a general smoothing tool.

More generally, the nth degree polynomial model is

$$\mu_i = \alpha_0 + \alpha_1 x_i + \alpha_2 x_i^2 + \alpha_3 x_i^3 + \cdots + \alpha_n x_i^n,$$

resulting in $n + 1$ columns in the matrix B, augmenting (2.1) to powers of n. This again gives $\mu = B\alpha$ in matrix notation. We call B a basis matrix and the powers of x the basis functions.

Figure 2.2 shows data from a simulated motorcycle crash, with a complicated trend: it is a time series of the acceleration of a helmet (Härdle, 1992). These data have become a workbench data set and a rite of passage for many smoothing techniques. A polynomial of low degree has no chance to fit these data well, so we try degree 9 (an arbitrary choice). Two fits are provided: one where all data were used (solid blue curve) and another where all data less than 5 ms were dropped (broken red curve). This small change has rather large consequences. The two curves differ strongly at the left (near 5 ms), which is expected, as we have changed the data there. But we also find large differences at the very right end (near 50 ms), which is unsettling.

Polynomial basis functions are global: they have a nonzero value for almost every x. The net effect is that any change in one of the coefficients in α results in a change in the curve over the entire domain of x. Worse, the higher the degree of the polynomial, the stronger this effect becomes.

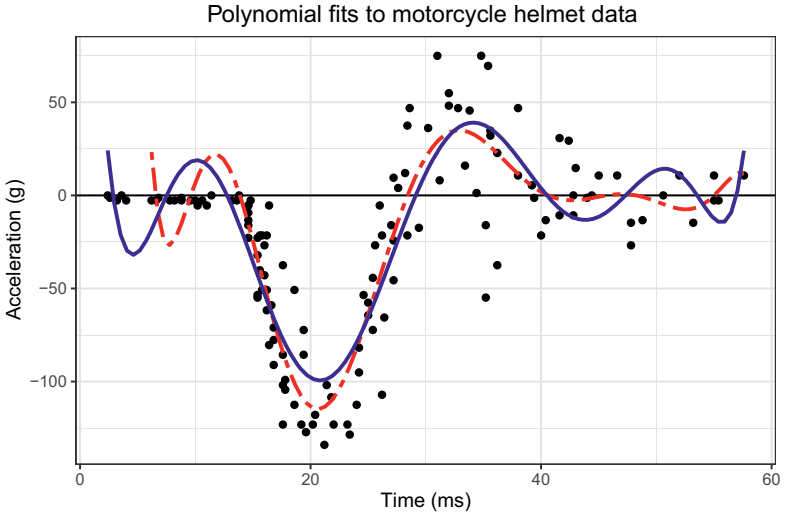

Figure 2.2 The acceleration of a motorcycle helmet in a simulated crash. Two polynomial (degree 9) fits are displayed. Blue line: based on all data; red broken line: after discarding the observations at less than 5 ms. `R` code in `f-motpol1.R`

2.2 B-splines

We first visualize B-splines before using them. The left panel of Figure 2.3 shows seven B-splines, shifted vertically to separate them. The right panel provides a more standard presentation. For either panel, the middle curve shows one complete B-spline, which strongly resembles a normal density. The other curves are shifted copies of this middle curve, but truncated at the left or right boundary.

These are so-called cubic, or degree 3, B-splines. Each B-spline consists of four polynomial segments, each of degree 3, that begin and end at specific values of x called knots. In Figure 2.3, the knots are located at the integer numbers 0 to 4. At the inner knots (1 to 3) two polynomial segments of the same B-spline meet; their values and those of the first and second derivative are equal on both sides of each knot. Together these (degree $+1$) polynomial segments form one B-spline basis function (resembling a normal density).

The knots divide the domain of x into four sections of equal length. The number of B-splines is seven because they have degree 3. In both panels a vertical broken line visualizes the evaluation of the B-splines for one value of x.

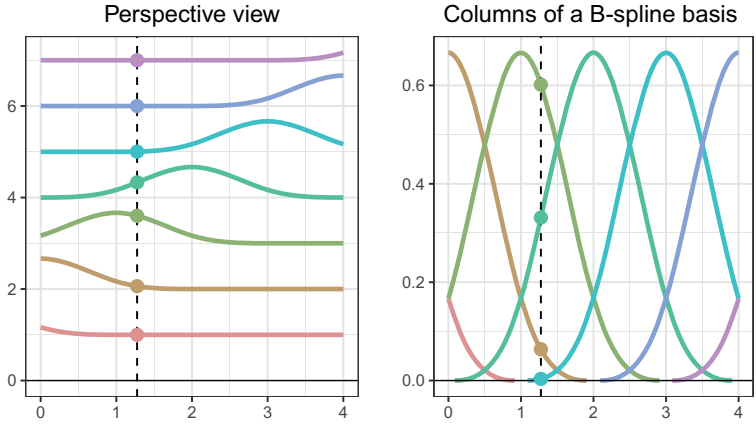

Figure 2.3 B-splines in perspective. In the left panel, the splines are offset vertically, in the right panel they are plotted on top of each other. R code in `f-persp.R`

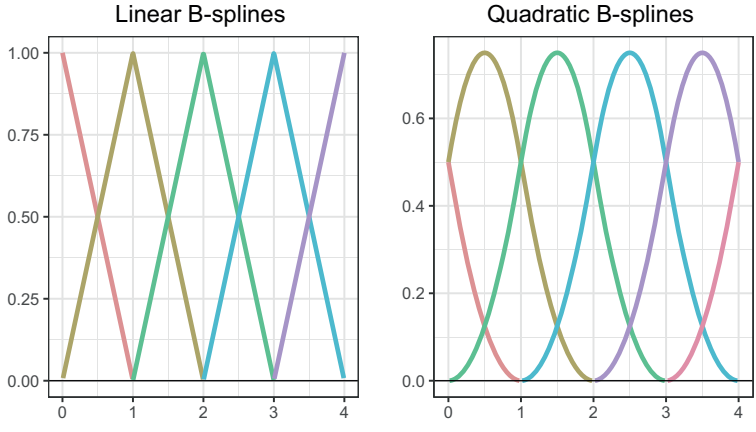

Figure 2.4 Linear (left) and quadratic (right) B-spline bases illustrated. R code in `f-B-lin-quad.R`

Only four of the evaluations have a nonzero value; which four is determined by the value of x. It is easy to check this by imagining a vertical line anywhere in the two panels. The number four is determined only by the degree of the B-splines and does not depend on their number. Said differently, even in a large basis with many B-splines, only four of them are nonzero for any x. Figure 2.4 shows linear and quadratic B-splines for the same choice of knots.