# Statistical Modeling with Spline Functions
# Methodology and Theory

Mark H. Hansen
University of California at Los Angeles

Jianhua Z. Huang
University of Pennsylvania

Charles Kooperberg
Fred Hutchinson Cancer Research Center

Charles J. Stone
University of California at Berkeley

Young K. Truong
University of North Carolina at Chapel Hill

January 5, 2006

# 1
# Introduction

The topic of this book is function estimation using polynomial splines. The polynomial spline approach is one of several to estimate functions that are smooth, but not necessarily linear. Other approaches include kernel methods, local polynomial methods and smoothing splines. Some of the advantages of the polynomial spline approach are that the method can be generalized to many different function estimation approaches, and that, based on an ANOVA decomposition, the polynomial spline approach can be applied to high dimensional problems.

This book covers both the methodology and the theory of polynomial splines. While often the theoretical (convergence) results are obtained in more idealized situations than those encountered in real data, and the methodologies include many details that are impossible to include in the theory, we feel that methods and theory go hand in hand. The fact that certain fast convergence rates are achieved in theory, should give us confidence that the methods actually work. Some of the assumptions and conditions needed to proof theorems actually give insight in how to design software, while success stories and failures of the methods suggest which theoretical results may or may not be true.

In this book we discuss linear regression, generalized linear regression, density estimation, survival analysis, and several other models. For most of these models we discuss simple univariate problems, as well as complicated real-life high-dimensional problems. The book is both intended to explain the polynomial spline approach and to be a guide for those people who want to apply polynomial spline methods.

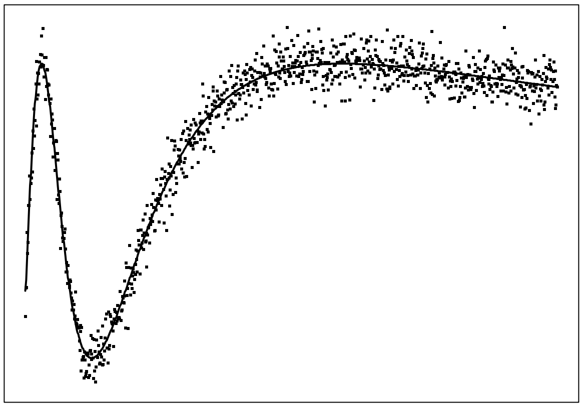

FIGURE 1.1. A simulated dataset with the true mean function $f$.

We start with giving a very brief introduction of polynomial splines in the simplest setting: univariate regression. After that we will give an overview of this book, and provide some background by comparing the development of polynomial splines to other function estimation (smoothing) methods, and by providing a historical perspective. We end this chapter by discussing software that is available to apply the methods discussed in this book.

## 1.1 Overview

In Figure 1.1 we show a simulated dataset. The $X_i$ are 1000 equidistant points between 0.1 and 1, and $Y_i = \phi(X_i) + Z_i$, where $\phi(x) = \sin(1/x)$ and the $Z_i \sim N(0, 0.1^2)$. The goal is to estimate the underlying mean function $\phi$. We can do this using polynomial splines.

Polynomial splines are piecewise polynomials, satisfying some continuity conditions at the places where the polynomials end. In a univariate setting these breakpoints are usually called knots. For the commonly used cubic splines a polynomial spline is a cubic polynomial on each interval between two consecutive knots, and it has two continuous derivatives at each knot; a linear spline is linear between each two consecutive knots, and it is continuous at the knots. Other than selecting the exact class of splines to use, the difficult problem is usually where to position the knots.

In Figure 1.2 we show four cubic spline fits to the data. The one in the left top is optimal, in that this was the model selected using backwards elimination and GCV, two notions which we will discuss in more detail in Chapter 3. The selection of the knots involves both selection of the number of knots and selection of the location of the knots. Both are crucial, as is obvious from the other panels of Figure 1.2. In the right top panel we use the optimal locations for a too small number of knots. We see that we miss important details of $\phi$. In the left bottom panel we use the optimal

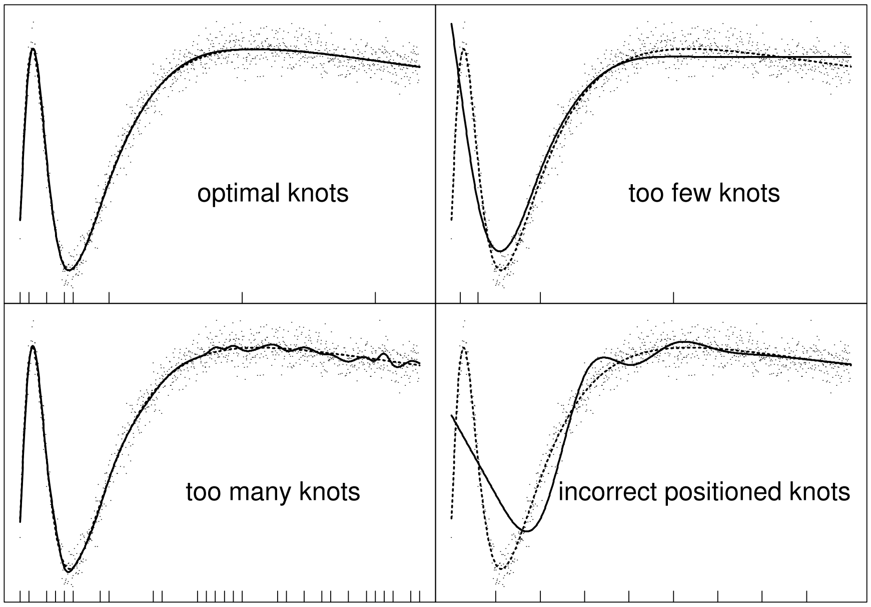

FIGURE 1.2. A simulated dataset with the true mean function $f$ (dashed) and four cubic spline fits (solid).

locations for too many knots: this results in wiggles for estimate $\widehat{\phi}$ for large values of $x$. When we use the same number of knots as in the optimal solution, but we put the knots equally spaced, we miss details and we get some undesirable wiggles. Figure 1.3 shows similar fits for piecewise linear splines: except that the fitted curves are obviously less smooth, we note the same issues as for cubic splines.

From a theoretical perspective the interesting question is how good are we able to approximate $\phi$ using a polynomial spline. This question often breaks up in two problems: how good an approximation based on splines to $\phi$ is possible, and how close can we expect an estimate based on sample data to get to the best possible approximation.

Clearly, when the sample size increases we want to increase the number of knots. This is both true for the methods and the theory. Here the theory will provide us insight as to how fast we should let the number of knots increase in the methodology.

### *1.1.1 Why do we end up using splines?*

Assume that we have a sample $(X_i, Y_i)$, $i = 1, \ldots, n$ from the model

$$E(Y|X) = \phi(X). \tag{1.1.1}$$

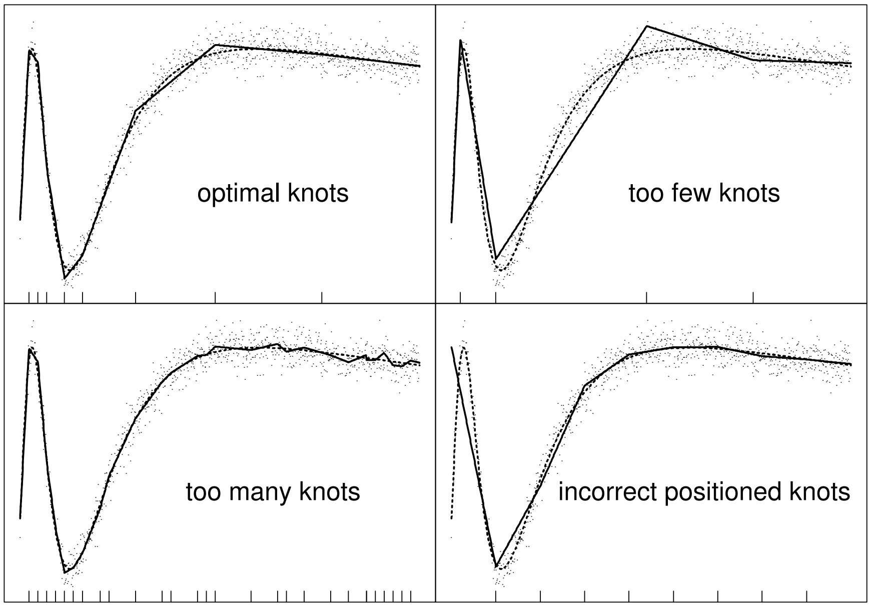

FIGURE 1.3. A simulated dataset with the true mean function $f$ (dashed) and four linear spline fits (solid).

A traditional linear regression model for such data is

$$\widehat{\phi}(x) = \beta_0 + \beta_1 x, \tag{1.1.2}$$

and the parameters $\beta_0$ and $\beta_1$ in such a model are estimated using the method of least squares. Such a model is only appropriate if the true $\phi$ is (approximately) linear. If that is not the case, we have to consider other models. A traditional extension of (1.1.2) is to consider polynomials:

$$\widehat{\phi}(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots \beta_q x^q.$$

In Figure 1.4 we show a linear, quadratic, cubic and quartic fit to the part of the data for which $x < 0.25$ from Figure 1.1. There are a number of problems with high order polynomial fits: (i) the design matrix in the regression problem can get very ill-conditioned, and (ii) the effects of the polynomial are nonlocal. This second effect can be observed from the last two panels of Figure 1.4, where the fitted curves become much smaller or larger only just outside the data. Keeping this in mind, it is not surprising that the variance of polynomial fits gets very large near the boundary of the range of the data, something that can be easily verified theoretically.

An alternative way to estimate $\phi$ at a particular value of $x$ is to take all data points that are within a bandwidth $h$ of $x$, and to take the average of these observations (see Figure 1.5). Such an estimator is known as a kernel estimator or a local average estimator (Fan and Gijbels 1996). From the

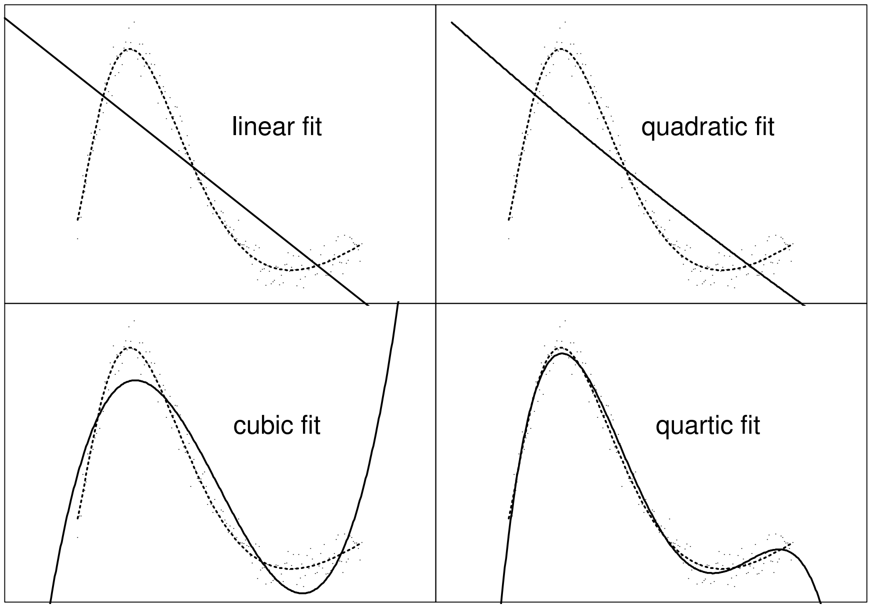

FIGURE 1.4. A simulated dataset with the true mean function $f$ (dashed) and four polynomial fits (solid).

left top panel of Figure 1.6 it is evident that this estimator has problems with boundary effects, as well as with estimating the true height of peaks. These effects can be diminished by reducing $h$, but at the price of increased variability. These boundary effects can also be reduced by using linear regression to fit a line to all points that are within $h$ of $x$, and to use as $\widehat{\phi}(x)$ the regression estimate of that line at $x$ (Figure 1.5 right-hand side). This is known as a local linear estimator (Cleveland 1979; Stone 1980; Fan and Gijbels 1996; Loader 1999). From the right top panel of Figure 1.6 we observe that the boundary effect has disappeared, but that the underestimation at the peak is still present. Higher order local polynomial fits would eliminate this effect as well.

Rather than fitting a straight line at each point, we could also decide to fit a smaller number of linear regressions to disjoint sets of the data. In the left bottom panel of Figure 1.6 we show the fits of five separate regression lines: for data with $x \in [0.1, 0.13)$, for data with $x \in [0.13, 0.16)$, for data with $x \in [0.16, 0.19)$, for data with $x \in [0.19, 0.22)$, and for data with $x \in [0.22, 0.25]$. This piecewise linear curve is not continuous. From here it is a small step to fitting a piecewise linear curve that is continuous. This is exactly the type of linear spline that we plotted in Figure 1.3. The right bottom panel in Figure 1.6 shows a linear spline with four equidistant knots. As we already discussed above, selection of the location and number of knots is critical for using polynomial splines.

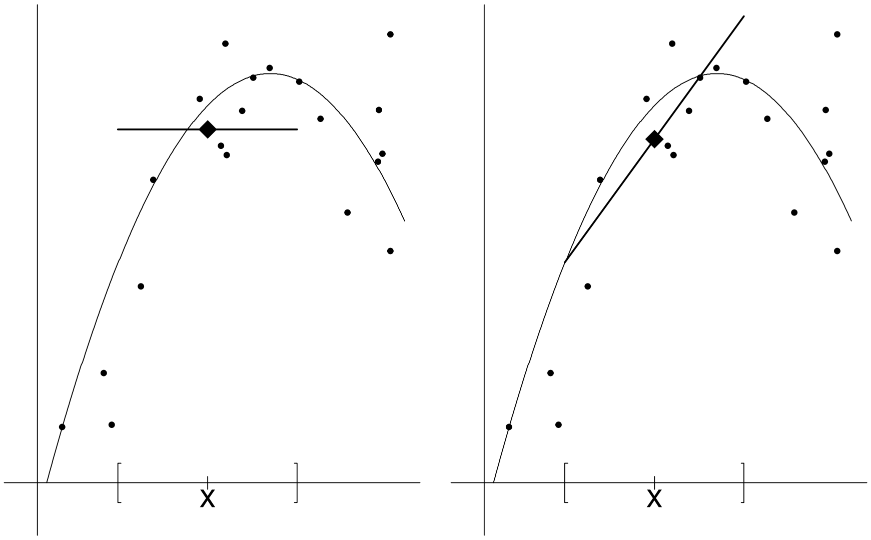

FIGURE 1.5. A local average (left) and local linear (right) estimate of $\phi(x)$.

### *1.1.2 Broad outline of methods used*

Univariate, polynomial splines are piecewise polynomials of some degree $d$. The breakpoints marking a transition from one polynomial to the next are referred to as *knots*. Typically, a spline will also satisfy *smoothness constraints* describing how the different pieces are to be joined. These restrictions are specified in terms of the number of continuous derivatives, $s$, exhibited by the piecewise polynomials. Consider, for example, piecewise linear curves. Without any constraints, these functions can have discontinuities at the knots. By adding the condition that the functions be globally continuous, we force the separate linear pieces to meet at each knot. If we demand even greater smoothness (say, continuous first derivatives), we loose flexibility at the knots and the curves become simple linear functions. In the literature on approximation theory, the term "linear spline" is applied to a continuous, piecewise linear function. Similarly, the term "cubic spline" is reserved for piecewise cubic functions having two continuous derivatives, allowing jumps in the third derivatives at the knots.

Given a degree $d$ and a knot vector $\boldsymbol{t} = \{t_1, \ldots, t_K\}^t$, the collection of polynomial splines having $s$ continuous derivatives forms a linear space. For example, the collection of linear splines with knot sequence $\boldsymbol{t}$ is spanned by the functions

$$1, x, (x - t_1)_+, \ldots, (x - t_K)_+ \tag{1.1.3}$$

where $(\cdot)_+ = \max(\cdot, 0)$. We refer to this set as the *truncated power basis* of the space. Cubic splines have $d = 3$ and $s = 2$ and its basis is spanned by

$$1, x, x^2, x^3, (x - t_1)^3_+, \ldots, (x - t_k)^3_+ \,. \tag{1.1.4}$$

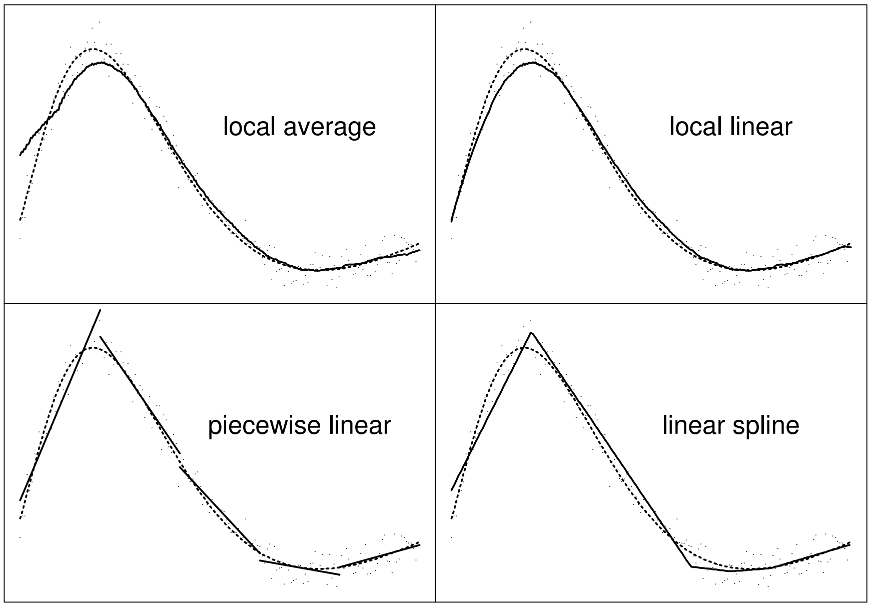

FIGURE 1.6. A simulated dataset with the true mean function $f$ (dashed) and four linear spline fits (solid).

From a modeling standpoint, the truncated power basis is convenient because the individual functions are tied to knot locations. In the expressions (1.1.3) and (1.1.4), there is exactly one function associated with each knot, and eliminating that function effectively removes the knot.

The truncated power functions (1.1.3) and (1.1.4) are known to have rather poor numerical properties. In linear regression problems, for example, the condition of the design matrix deteriorates rapidly as the number of knots increases. An important alternative representation is the so-called *B-spline basis* de Boor (1978). These functions are constructed to have support only on a few neighboring intervals defined by the knots. A detailed description of this basis is given in Chapter 2.

For the moment, assume we can find a basis $B_1(x;\boldsymbol{t}),\ldots,B_J(x;\boldsymbol{t})$ for the space of splines of degree $d$ with smoothness $s$ and knot sequence $\boldsymbol{t}$ so that any function in the space can be written as

$$g(x;\boldsymbol{\beta},\boldsymbol{t}) = \beta_1 B_1(x;\boldsymbol{t}) + \cdots \beta_J B_J(x;\boldsymbol{t})\,, \tag{1.1.5}$$

for some coefficient vector $\boldsymbol{\beta} = (\beta_1,\ldots,\beta_J)^t$. We can use functions of the form (1.1.5) to approximate an unknown regression function. Assume again that we have a sample $(X_i, Y_i)$, $i = 1,\ldots,n$ from model (1.1.1). For any fixed knot vector $\boldsymbol{t}$ we can estimate $\boldsymbol{\beta}$ using the method of least squares

$$\hat{\beta} = \arg\min_{\boldsymbol{\beta}} \mathrm{RSS}(\boldsymbol{\beta},\boldsymbol{t}), \tag{1.1.6}$$

where

$$\begin{aligned} \text{RSS}(\boldsymbol{\beta}, \boldsymbol{t}) &= \sum_i (Y_i - g(x_i; \boldsymbol{\beta}, \boldsymbol{t}))^2 \\ &= \sum_i (Y_i - \beta_1 B_1(X_i; \boldsymbol{t}) - \cdots - \beta_J B_J(X_i; \boldsymbol{t}))^2 \,. \end{aligned}$$

Then $g(\cdot; \hat{\beta}, \boldsymbol{t})$ is an estimate for $\phi(\cdot)$.

As was shown in Figures 1.2 and 1.3 selecting both the number of knots $K$ and the location $\boldsymbol{t}$ is of critical importance. For many of the methodologies discussed later in this book we use a stepwise algorithm. That is, we start with a small initial set of knots $\boldsymbol{t}^0$. After estimating $\hat{\beta}^{\mathbf{0}}$ based on these knots, we have a (heuristic) search for that knot, whose addition to the set of knots will reduce the RSS as much as possible. We continue this process of adding knots until a maximum number is reached, after which we start stepwise deletion of knots. Here at each stage we remove the knot that cause the smallest increase in the RSS. This way we obtain a sequence of models. Among these models the largest model has the smallest RSS. However, this model may over-fit the data. Therefore we usually select the model that optimizes a version of the RSS that is penalized for the model size, such as GCV or AIC. Many of the details that are essential in developing a polynomial spline linear regression methodology are discussed in Chapter 3. In Chapter 10 we discuss alternatives to the stepwise algorithms to select knots.

For models other than linear regression people typically use the method of maximum likelihood to estimate the parameters. Similarly, in polynomial spline routines, we would replace, equation (1.1.6) by

$$\hat{\beta} = \arg\max_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta}, \boldsymbol{t}),$$

where $\ell$ is the log-likelihood function. Details of such procedures are discussed in later chapters.

For multivariate problems, that is problems in which for each case $i$ there is a vector of $q$ predictors $\boldsymbol{X}_i = \{X_{i1}, \ldots, X_{ip}\}^t$, and the true model is

$$E(Y|\boldsymbol{X}) = \phi(\boldsymbol{x}). \tag{1.1.7}$$

The usual parametric (linear) model to estimate $\phi$ is

$$E(Y|\boldsymbol{X}) = \beta_0 + \beta_1 X_1 + \cdots \beta_q X_q.$$

The simplest generalization of the model (1.1.1) is the additive model

$$E(Y|\boldsymbol{X}) = \phi_1(X_1) + \cdots \phi_q(X_q).$$

In such a model each component can be modeled using a polynomial spline approach. Alternatively, we can use a more general form using a functional

ANOVA decomposition. Suppose that $q = 3$ than we can write

$$\begin{aligned} E(Y|\boldsymbol{X}) \;=\; \phi(\boldsymbol{X}) \;=\; & \phi_0 + \phi_1(X_1) + \phi_2(X_2) + \phi_3(X_3) + \phi_{12}(X_{12}) \\ & + \phi_{13}(X_{13}) + \phi_{23}(X_{23}) + \phi_{123}(X_{123}). \qquad (1.1.8) \end{aligned}$$

We need some orthogonality constraints on the $\phi$ to guarantee uniqueness. In a polynomial spline methodology, we model each of the components of (1.1.8) separately using polynomial splines. In practice, especially when $p$ is large, we may not want to model each of the components of $\phi$ separately in (1.1.8), but rather we may want to select a smaller number of (lower dimensional) components and use only those. In the next section we will see that selecting only lower dimensional components has a good theoretical foundation. In our methodologies we have found that estimating only the one- and two-dimensional components works well in practice.

### *1.1.3 Broad outline of theory*

Both the theory and the methodology of polynomial splines makes extensively use of the notion of extended linear models. These models are discussed extensively in Chapters 3, 4, and 11. Briefly, extended linear models compromises a large class of commonly used statistical models, including linear regression, generalized regression, density estimation, and survival analysis, that have a concave log-likelihood like function. This log-likelihood function depends on a regression function $\phi$, as in the model (1.1.7). When we try to estimate $\phi$ we will assume that this function falls in a particular function class, for example, functions that satisfy some smoothness condition. As the true function $\phi$ may not satisfy those, the best that we will be able to do is to find an estimate $\phi^*$, which is the closest function in the function class that we consider to $\phi$. This all will be formalized in Chapter 11. Stone (1980) established that the best possible convergence rates for the distance between an estimate $\widehat{\phi}$ and $\phi^*$, In particular, he established that

$$\|\widehat{\phi} - \phi^*\| = O_P(n^{-p/(2p+q)}). \qquad (1.1.9)$$

for a $q$-dimensional function $\phi$, that has a smoothness $p$ ($p = 2$ implies that the regression function has bounded second derivatives). Here one can interpret $\|\widehat{\phi} - \phi^*\|$, the distance between $\widehat{\phi}$ and $\phi^*$, as the error in the estimation. This implies that if we want the error to be small, we need $\log n$ to be of the order $(2p+q)/p$. Stone (1980) established that these rates are attained by local polynomial methods. Similarly, these rates can be attained by polynomial splines, provided the number of knots increases slowly with the sample size. These results, unfortunately, imply that if $q$ gets large the convergence can be very slow, and we need a real large sample to obtain a good estimate of $\phi^*$. This is known as the curse of dimensionality.

To circumvent the curse of dimensionality, we make use of the same type of ANOVA decompositions as was done for the methodology above. As an example, let $q = 3$. We first decompose the true regression function $\phi$ as

$$\begin{aligned}\phi(\boldsymbol{X}) &= \phi_0 + \phi_1(X_1) + \phi_2(X_2) + \phi_3(X_3)\\ &\quad +\phi_{12}(X_{12}) + \phi_{13}(X_{13}) + \phi_{23}(X_{23}) + \phi_{123}(X_{123}),\end{aligned}$$

We can now also decompose the best possible approximation

$$\begin{aligned}\phi^*(\boldsymbol{X}) &= \phi_0^* + \phi_1^*(X_1) + \phi_2^*(X_2) + \phi_3^*(X_3)\\ &\quad +\phi_{12}^*(X_{12}) + \phi_{13}^*(X_{13}) + \phi_{23}^*(X_{23}) + \phi_{123}^*(X_{123}),\end{aligned}$$

If we only consider additive functions to approximate $\phi$, then

$$\phi^*(\boldsymbol{X}) = \phi_0^* + \phi_1^*(X_1) + \phi_2^*(X_2) + \phi_3^*(X_3).$$

Stone (1985) established that for additive regression the cobvergence rate result (1.1.9) gets replaced by

$$\|\widehat{\phi} - \phi^*\| = O_P(n^{-p/(2p+1)}), \tag{1.1.10}$$

which is a much faster rate when $q$ is large. In particular, this gives a theoretical justification for using generalized additive models (Hastie and Tibshirani 1990). Obviously there is a price, if $\phi$ is not additive, the distance between $\phi$ and $\phi^*$ now becomes much larger, so the estimate $\widehat{\phi}$ still may not be good.

If we do not believe that $\phi$ is additive, we could, however, decide to model $\phi$ using second order interactions. In this case,

$$\begin{aligned}\phi^*(\boldsymbol{X}) &= \phi_0^* + \phi_1^*(X_1) + \phi_2^*(X_2) + \phi_3^*(X_3)\\ &\quad +\phi_{12}^*(X_{12}) + \phi_{13}^*(X_{13}) + \phi_{23}^*(X_{23}),\end{aligned}$$

and also

$$\begin{aligned}\widehat{\phi}(\boldsymbol{X}) &= \widehat{\phi}_0 + \widehat{\phi}_1(X_1) + \widehat{\phi}_2(X_2) + \widehat{\phi}_3(X_3)\\ &\quad +\widehat{\phi}_{12}(X_{12}) + \widehat{\phi}_{13}(X_{13}) + \widehat{\phi}_{23}(X_{23}).\end{aligned}$$

Stone (1985) established that in this situation

$$\|\widehat{\phi} - \phi^*\| = O_P(n^{-p/(2p+2)}),$$

or, in general, if the highest order interactions are of order $d$, then

$$\|\widehat{\phi} - \phi^*\| = O_P(n^{-p/(2p+d)}).$$

Similar rates were later obtained for generalized regression (Stone 1986; Stone 1994), desnity estimation (Stone 1990; Stone 1994), conditional density estimation (Stone 1991; Stone 1994; Hansen 1994), hazard regression (Kooperberg, Stone, and Truong 1995b), spectral density stimation (Kooperberg, Stone, and Truong 1995d), event history analysis (Huang and Stone 1998), and proportional hazards models (Huang, Kooperberg, Stone, and Truong 2000). Hansen (1994) and Huang (2001) established a theoretical synthesis that holds for a much larger class of extended linear models.

### *1.1.4 Chapter by chapter overview*

Chapter 2, Preliminaries, discusses some properties of splines, as well as some basic elements in numerical analysis that are useful for extended linear modeling. Depending on ones background some users may want to use this chapter as a reference rather than reading it in detail. The following Chapters 3 through 10 discuss methodology, and the last two Chapters 11 and 12 discuss theory.

Chapters 3 through 9 discuss the use of polynomial splines and extended linear modeling to various methodological settings. All of these chapters contain discussion of methodological issues, as well as substantial applications of the methodology. In particular, Chapter 3 discusses both univariate and multivariate regression, including the MARS (Friedman 1991) algorithm. Chapter 4 extends the approach from the previous chapter to generalized regression, including logistic and Poisson regression models. These first two methods chapters are fairly general, and the later methods chapters all use key ideas first discussed in these chapters. The next five methods chapters discuss five different methods problems, and are independent of each other. In particular, Chapter 5 discusses polychotomous regression and classification, including the Polyclass and PolyMARS (Kooperberg, Bose, and Stone 1997) algorithms. Chapter 6 discusses Logspline density estimation (Kooperberg and Stone 1991; Kooperberg and Stone 1992; Kooperberg and Stone 2001b). This chapter also contains a brief discussion about the merits of free-knot splines, and their use in inference. Chapter 7 discusses survival analysis, and in particular the Hare (Kooperberg, Stone, and Truong 1995a; Kooperberg and Clarkson 1997), Heft (Kooperberg, Stone, and Truong 1995a), and PHare (Huang, Kooperberg, Stone, and Truong 2000) methodologies. This chapter includes some discussion on how simulations and the bootstrap can be used to make inferences in polynomial spline models. Chapter 8 discusses the Lspec approach to estimation of a possibly mixed spectral density (Kooperberg, Stone, and Truong 1995c) Chapter 9 discusses the Triogram (Hansen, Kooperberg, and Sardy 1998) methodology using truly bivariate splines. While the two examples in this chapter are regression and density estimation, reading of the density estimation chapter is not required before reading Chapter 9. Other than a brief excursion to free-knot splines in Chapter 6 model selection in Chapters 3–9 is carried out primarily using stepwise methods. In Chapter 10, however, we discuss alternatives to the stepwise approach using a Bayesian approach, Markov chain Monte Carlo, and simulated annealing. As the two main examples in this chapter are Logspline density estimation and Triogram regression, it is recommended to read Chapters 6 (Density Estimation) and 9 (Multivariate Splines) before reading Chapter 10.

The theory Chapters 11 and 12 can be read independently of the methods Chapters 3–10. A better appreciation of the theory may be obtained if also Chapters 3 (Linear Models) and 4 (Generalized Linear Models) are read.

Chapter 11 establishes convergence rates results for extended linear models. In this Chapter the number of knots increases with the sample size, but the knot sequence is assumed to be determined externally, satisfying some mild regularity conditions. The results in Chapter 11 are quite general, and apply to most of the problems discussed in the methods chapters. In particular, the theory makes use of a similar ANOVA decomposition for higher dimensional problems as is being used in the methods chapters. In Chapter 12 the results of Chapter 11 are generalized to free knot splines, the situation in which the knot locations are also free parameters.

The main author of the preliminary Chapter 2 was Young Truong, the eight methodological Chapters 3–10 were primarily written by Mark Hansen and Charles Kooperberg, and the two theoretical Chapters 11 and 12 were primarily written by Jianhua Huang and Charles Stone.

## 1.2 Background

### *1.2.1 Other smoothing methods*

1. kernel and other averaging methods including nearest neighbor stuff
2. local polynomial fits including Cleveland's stuff
3. generalized additive models including Hastie-Tibshirani stuff
4. smoothing splines, P-splines
5. wavelets (or maybe not)

### *1.2.2 Some history*

## 1.3 Software

There are several sources of software to apply the methodologies that are discussed in Chapters 3–10. Here we list several approaches. The authors of this book are not responsible for the various programs listed here, and should not be taken as an endorsements of these programs. The responsibility for the code lies by the author of the software. Websites and email addresses change over time. The list here is correct as of August, 2000. We plan to keep an up-to-date list of software for polynomial splines on the webpage of Charles Kooperberg, which currently is `http://bear.fhcrc.org/`*sim*`clk`. Please report any other sources of software for polynomial splines to him. Do not report bugs in any of the software to him, but rather report those to the authors of the software.

- Insightful Inc., maker of S-Plus$^{\mathrm{tm}}$ plans to make code available for Logspline density estimation, hazard regression (Hare) and adaptive generalized linear models (aglm) in a future release of S-Plus. Their implementation of Logspline and Hare is virtually identical to the methods discussed in Chapters 6 and 7. Their aglm program applies a similar modeling approach to linear and generalized linear models, and, as such, includes algorithms similar to MARS$^{\mathrm{tm}}$, PolyMARS, Polyclass(for two classes only), and other methods discussed in Chapters 3 and 4. (`http://www.insightful.com`).
- Versions of Polyclass, PolyMARS, Logspline, Hare, Heft, and Lspec written in C, with a interface to S-Plus for Unix/Linux, are available from the homepage of Charles Kooperberg

  `http://bear.fhcrc.org/~clk/soft.html`

  This page also contains links to other webpages containing translations of some of this code for R and for S-Plus for Windows. These programs are also available from CRAN.
- Some of Kooperberg's code has been ported to Xplore (`http://www.xplore-stat.de`).
- Software for Triogram regression, discussed in Chapter 9 is available from Mark Hansen (`cocteau@research.bell-labs.com`).
- Salford Systems markets a commercial implementation of the MARS$^{\mathrm{tm}}$ algorithm of Friedman (1991) discussed in Chapter 3. (`http://www.salford-systems.com`).

We should also point out that it is less difficult than it may appear to design your own code for fitting adaptive polynomial spline models using stepwise algorithms.

1. To fit a model with fixed knots/basis functions you will need a B-spline or natural spline routine, such as `bs()` or `ns()` in S-Plus, to generate a design matrix, and a fitting routine, such as (`glm()`).
2. To create a stepwise *deletion* algorithm, you would in addition need to compute Wald statistics, using the formulas given in Chapter 4. Note that most fitting routines will already provide the Hessian and the coefficients, so the only difficult part is to get the matrix **A** with linear contrasts. If this turns out to be too complicated, you can always resort to actually fitting all candidate models that can be obtained by removing one basis function. As there usually is a very limited number, this will often be fairly efficient.
3. To add a stepwise addition component may be harder. You will need a routine that identifies candidate basis functions to be added to your

current set of basis functions, a routine to compute Rao statistics (in S-Plus the `step()` function is useful), and a routine to roughly optimize knot-locations.

4. Evaluating which model fits best using AIC or GCV is usually a straightforward combination of some of the output of the fitting algorithms.