apts.ac.uk
Academy for PhD Training in Statistics

Adrian Bowman, Ludger Evers
School of Mathematics & Statistics
The University of Glasgow

Nonparametric Smoothing

Lecture Notes

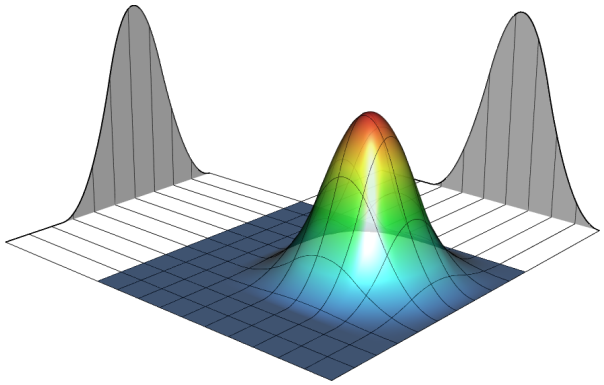


Table of Contents

1

Introduction

1.1 What this course is about

This APTS course will cover a variety of methods which enable data to be modelled in a flexible manner. It will use and extend a variety of topics covered in earlier APTS courses, including

– linear models, including the Bayesian version;
– generalised linear models;
– R programming;
– Taylor series expansions and standard asymptotic methods.

The main emphasis will be on regression settings, because of the widespread use and application of this kind of data structure. However, the material of the first chapter will include the simple case of density estimation, also covered in the preliminary material, to introduce some of the main ideas of nonparametric smoothing and to highlight some of the main issues involved.

As with any statistical topic, a rounded treatment involves a variety of approaches, including

– clear understanding of the underlying concepts;
– technical understanding of methods, with an exploration of their properties;
– appreciation of the practical computational issues;
– some knowledge of the tools available to fit relevant models in R;
– understanding of how these models can bring insight into datasets and applications.

The aim is to reflect all of these aspects in the course, but to varying degrees in different sections. There will not be time to cover all the material in the notes and some of the material is intended to provide pointers to topics which it might be of interest to explore in the context of your own research.

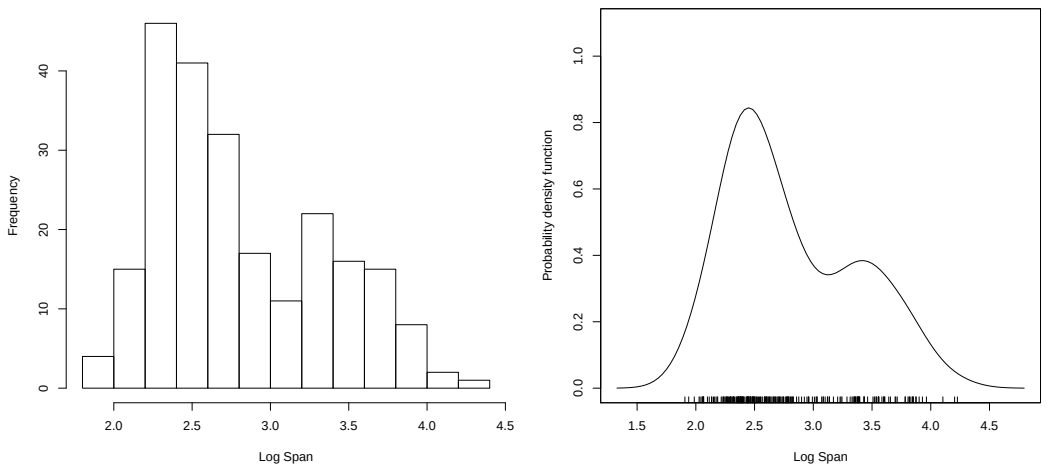
1.2 Brief review of density estimation

A probability density function is a key concept through which variability can be expressed precisely. In statistical modelling its role is often to capture variation sufficiently well, within a model where the main interest lies in structural terms such as regression coefficients. However, there are some situations where the shape of the density function itself is the focus of attention. The example below illustrates this.

Example 1.1 (Aircraft data). These data record six characteristics of aircraft designs which appeared during the twentieth century. The variables are:

Yr: year of first manufacture
Period: a code to indicate one of three broad time periods
Power: total engine power (kW)
Span: wing span (m)
Length: length (m)
Weight: maximum take-off weight (kg)
Speed: maximum speed (km/h)
Range: range (km)

A brief look at the data suggests that the six measurements on each aircraft should be expressed on the log scale to reduce skewness. Span is displayed on a log scale below, for Period 3 which corresponds to the years after the second World War. ◁



The pattern of variability shown in both the histogram and the density estimate exhibits some skewness. There is perhaps even a suggestion of a subsidiary mode at high values of log span, although this is difficult to evaluate.

The histogram is a very familiar object. It can be written as

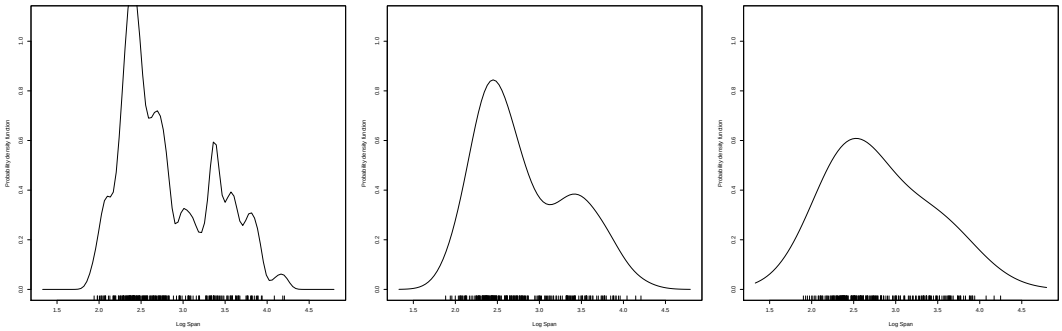
$$\tilde{f}(y) = \sum_{i=1}^{n} I(y - \tilde{y}_i; h),$$

where $\{y_1, \ldots, y_n\}$ denote the observed data, $\tilde{y}_i$ denotes the centre of the interval in which y_i falls and $I(z; h)$ is the indicator function of the interval $[-h, h]$. The form of the construction of $\tilde{f}$ highlights some feature which are open to criticism if we view the histogram as an estimator of the underlying density function. Firstly the histogram is not smooth, when we expect that the underlying density usually will be. Secondly, some information is lost when we replace each observation y_i by the bin mid-point $\tilde{y}_i$. Both of the issues can be addressed by using a density estimator in the form

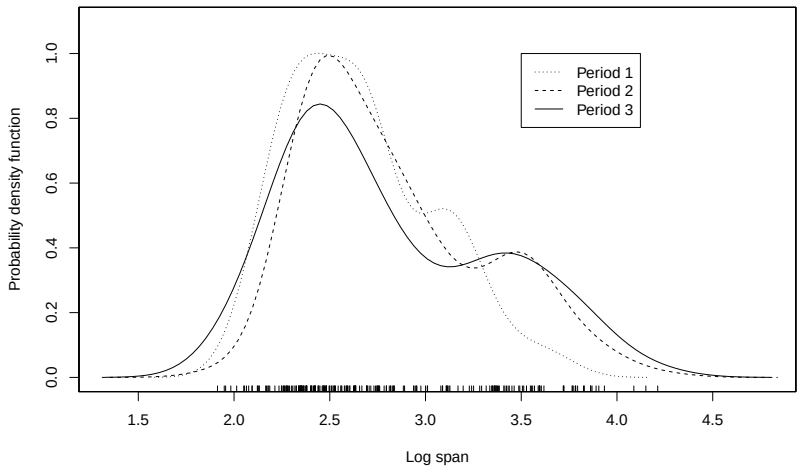
$$\hat{f}(y) = \frac{1}{n} \sum_{i=1}^{n} w(y - y_i; h),$$

where w is a probability density, called here a *kernel function*, whose variance is controlled by the *smoothing parameter* h.

Large changes in the value of the smoothing parameter have large effects on the smoothness of the resulting estimates, as the plots below illustrate.



One advantage of density estimates is that it is a simple matter to superimpose these to allow different groups to be compared. Here the groups for the three different time periods are compared. It is interesting that the 'shoulder' appears in all three time periods.



1.2.1 Simple asymptotic properties

Without any real restriction, we can assume that the kernel function can be written in the simple form $w(y - y_i; h) = \frac{1}{h} w\left(\frac{y-y_i}{h}\right)$. The preliminary material showed that the mean and variance of a density estimator can then be expressed as

$$\begin{aligned} \mathbb{E}\Big\{\hat{f}(y)\Big\} &= f(y) + \frac{h^2}{2}\sigma_w^2 f''(y) + o(h^2), \\ \mathsf{var}\Big\{\hat{f}(y)\Big\} &= \frac{1}{nh} f(y)\,\alpha(w) + o\left(\frac{1}{nh}\right), \end{aligned}$$

where we assume that the kernel function is symmetric so that $\int uw(u)du = 0$, and where σ_w^2 denotes the variance of the kernel, namely $\int u^2 w(u)du$, and $\alpha(w) = \int w^2(u)du$.

These expressions capture the essential features of smoothing. In particular, bias is incurred and we can see that this is controlled by f'', which means that where the density has peaks and valleys the density estimate will underestimate and overestimate respectively. This makes intuitive sense.

If we need it, a useful global measure of performance is the *mean integrated squared error* (MISE) which balances squared bias and variance.

$$\mathrm{MISE}(\hat{f}) = \frac{1}{4} h^4 \sigma_w^4 \int f''(y)^2 dy + \frac{1}{nh}\,\alpha(w) + o\left(h^4 + \frac{1}{nh}\right).$$

1.2.2 Extension to other sample spaces

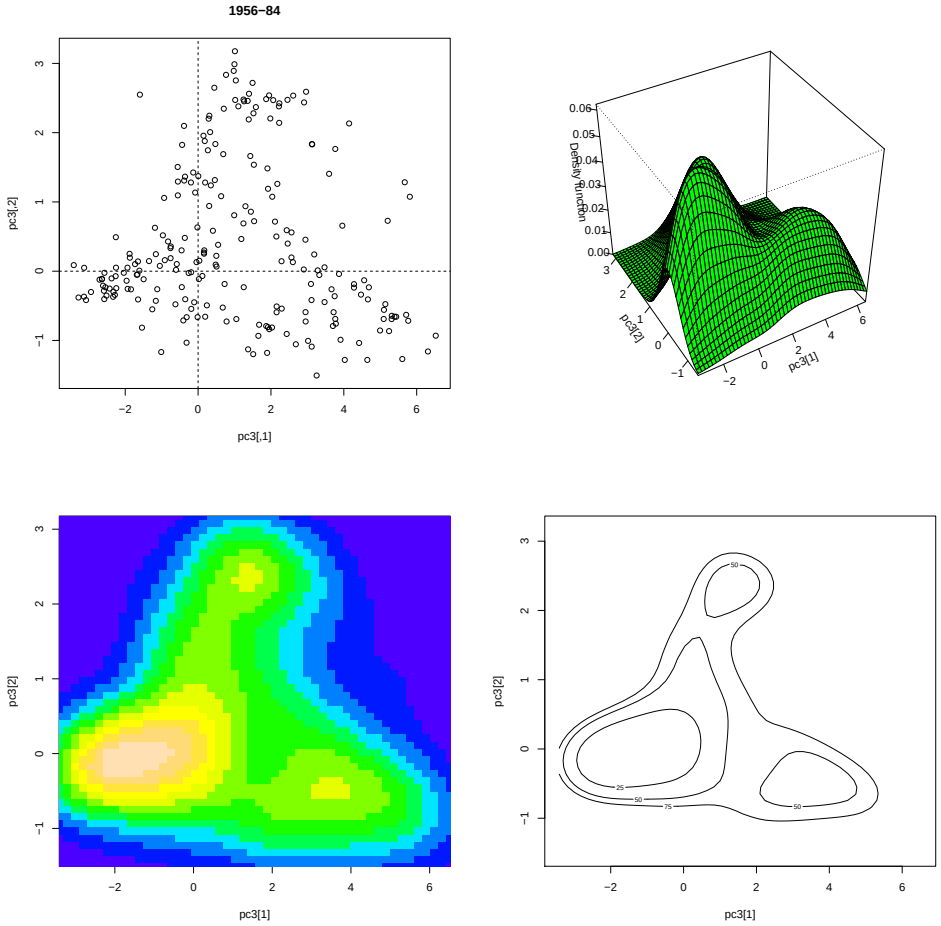
The simple idea of density estimation is to place a kernel function, which in fact is itself a density function, on top of each observation and average these functions. This extends very naturally to a wide variety of other types of data and sample spaces.

For example, a two-dimensional density estimate can be constructed from bivariate data $\{(y_{1i}, y_{2i}) : i = 1, \ldots, n\}$ by employing a two-dimensional kernel function in the form

$$\hat{f}(y_1, y_2) = \frac{1}{n} \sum_{i=1}^{n} w(y_1 - y_{1i}; h_1)\, w(y_1 - y_{2i}; h_2).$$

Notice that there are now two smoothing parameters, (h_1, h_2). A more general two-dimensional kernel function could be used, but the simple product form is very convenient and usually very effective.

Here is an example which uses the scores from the first two principal components of the aircraft data, again focussing on the third time period. The left hand scatterplot shows the individual scores while the right hand plot shows a density estimate, from which suggests three separate modes. This feature is not so easily seen from the raw scatterplot.



The lower two plots show alternative ways of presenting a two-dimensional estimate, using a coloured image on the left and contour lines on the right. Notice that the contours on the right have been chosen carefully to contain the quarters of the data with successively higher density, in a manner which has some similarities with a box plot.

This principle extends to all kinds of other data structures and sample spaces by suitable choice of an appropriate kernel function.

1.2.3 Deciding how much to smooth

It is not too hard to show that the value of h which minimizes MISE in an asymptotic sense is

$$h_{\text{opt}} = \left\{ \frac{\gamma(w)}{\beta(f)n} \right\}^{1/5},$$

where $\gamma(w) = \alpha(w)/\sigma_w^4$, and $\beta(f) = \int f''(y)^2 dy$. Of course, this is of rather limited use because it is a function of the unknown density. However, there are two practical approaches which can be taken to deciding on a suitable smoothing parameter to use. One is to construct an estimate of MISE and minimise this. Another is to estimate the optimal smoothing parameter. These two approaches are outlined below.

Cross-validation

The integrated squared error (ISE) of a density estimate is

$$\int \{\hat{f}(y) - f(y)\}^2 dy = \int \hat{f}(y)^2 dy - 2\int f(y)\hat{f}(y)dy + \int f(y)^2 dy.$$

Only the first two of these terms involve h and these terms can be estimated by

$$\frac{1}{n}\sum_{i=1}^{n} \int \hat{f}_{-i}^2(y)dy - \frac{2}{n}\sum_{i=1}^{n} \hat{f}_{-i}(y_i),$$

where $\hat{f}_{-i}(y)$ denotes the estimator constructed from the data without the observation y_i. The value of h which minimises this expression is known as the *cross-validatory* smoothing parameter.

Plug-in methods

By inserting suitable estimates of the unknown quantities in the formula for the optimal smoothing parameter, a *plug-in* choice can be constructed. The difficult part is the estimation of $\beta(f)$ as this involves the second derivative of the density function. Sheather & Jones (JRSSB 53, 683–90) came up with a good, stable way of doing this. The Sheather-Jones method remains one of the most effective strategies for choosing the smoothing parameter.

A very simple plug-in approach is to use the normal density function in the expression for the optimal smoothing parameter. This yields the simple formula

$$h = \left(\frac{4}{3n}\right)^{1/5} \sigma,$$

where σ denotes the standard deviation of the distribution. This is a surprisingly effective means of smoothing data, in large part because it is very stable.

1.2.4 Some simple inferential tools

Once an estimate has been constructed, a natural next step is to find its standard error. The earlier result on the variance of $\hat{f}$ is a natural starting point, but this expression involves the unknown density. A helpful route is to consider a 'variance stabilising' transformation. For any transformation $t(\cdot)$, a Taylor series argument shows that

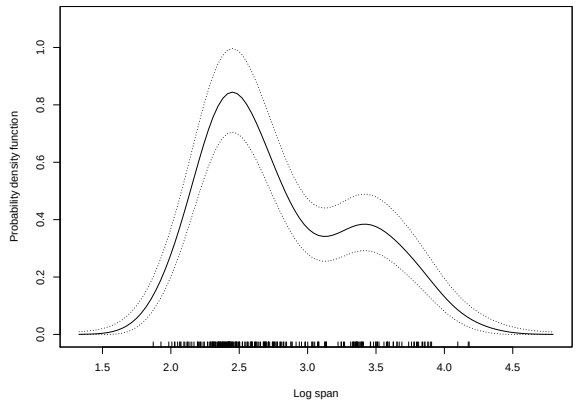
$$\mathsf{var}\left\{t(\hat{f}(y))\right\} \approx \mathsf{var}\left\{\hat{f}(y)\right\} \left[t'\left(\mathbb{E}\left\{\hat{f}(y)\right\}\right)\right]^2.$$

When $t(\cdot)$ is the square root transformation, the principal term of this expression becomes

$$\mathsf{var}\left\{\sqrt{\hat{f}(y)}\right\} \approx \frac{1}{4}\frac{1}{nh}\alpha(w),$$

which does not depend on the unknown density f. This forms the basis of a useful *variability band*. We cannot easily produce proper confidence intervals because of the

bias present in the estimate. However, if the standard error is constructed and the intervals corresponding to two s.e.'s on the square root scale are transformed back to the origin scale, then a very useful indication of the variability of the density estimate can be produced. This is shown below for the aircraft span data from period 3.



A useful variation on this arises when the true density function is assumed to be normal with mean μ and variance σ^2, and the kernel function w is also normal. If the standard normal density function is denoted by ϕ, then the mean and variance of the density estimate at the point y are then

$$\begin{aligned} \mathbb{E}\left\{\hat{f}(y)\right\} &= \phi\left(y-\mu; \sqrt{h^2+\sigma^2}\right) \\ \mathsf{var}\left\{\hat{f}(y)\right\} &= \frac{1}{n}\phi\left(0; \sqrt{2}\,h\right)\ \phi\left(y-\mu; \sqrt{\sigma^2+\frac{1}{2}h^2}\right) \\ &\quad -\frac{1}{n}\phi\left(y-\mu; \sqrt{\sigma^2+h^2}\right)^2 \end{aligned}$$

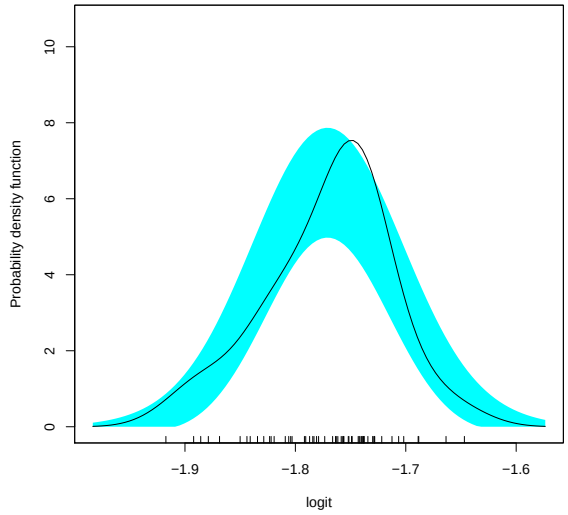
These expressions allow the likely range of values of the density estimate to be calculated, under the assumption that the data are normally distributed. This can be expressed graphically through a *reference band*.

Example 1.2 (Icelandic tephra layer). Data on the percentages of aluminium oxide found in samples from a tephra layer resulting from a volcanic eruption in Iceland around 3500 years ago are available in the `tephra` dataset in the `sm` package. To deal with the percentage scale, apply the logit transformation

```
logit <- log(tephra$Al2O3/(100-tephra$Al2O3))
```

.

Can the variation in the tephra data be adequately modelled by a normal distribution on this scale? ◁



The density estimate does not fit comfortably within the reference band at all points and this effect persists across a wide range of smoothing parameters. A global test of normality could be developed but the graphical device of a reference band offers very useful informal insight.

The preliminary material also discussed the role of the bootstrap in capturing the variability, but not immediately the bias, of a density estimate.

1.3 Broad concepts and issues of smoothing

The simple case of density estimation highlights features and issues which are common to a wide range of problems involving the estimation of functions, relationships or patterns which are *nonparametric* but *smooth*. The term nonparametric is used in this context to mean that the relationships or patterns of interest cannot be expressed in specific formulae which involved a fixed number of unknown parameters. This means that the parameter space is the space of functions, whose dimensionality is infinite. This takes us outside of the standard framework for parametric models and the main theme of the course will be to discuss how we can do this while producing tools which are highly effective for modelling and analysing data from a wide variety of contexts and exhibiting a wide variety of structures.

On a side note, the term nonparametric is sometimes used in the narrower setting of simple statistical methods based on the ranks of the data, rather than the original measurements. This is not the sense in which it will be used here.

Further details on density estimation are given in Silverman (1986), Scott (1992), Wand and Jones (1995) & Simonoff (1996).

The issues raised by our brief discussion of density estimation include

– how to construct estimators which match the type of data we are dealing with;
– how to find a suitable balance between being faithful to the observed data and incorporating the underlying regularity or smoothness which we believe to be present;
– how to construct and make use of suitable inferential tools which will allow the models to weigh the evidence for effects of interest, in a setting which takes us outside of standard parametric methods.

These broad issues will be explored in a variety of contexts in the remainder of the course.

1.4 Nonparametric regression

Regression is one of the most widely used model paradigms and this will be the main focus in the remainder of the course. Here is an example which will be used to illustrate the initial discussion.

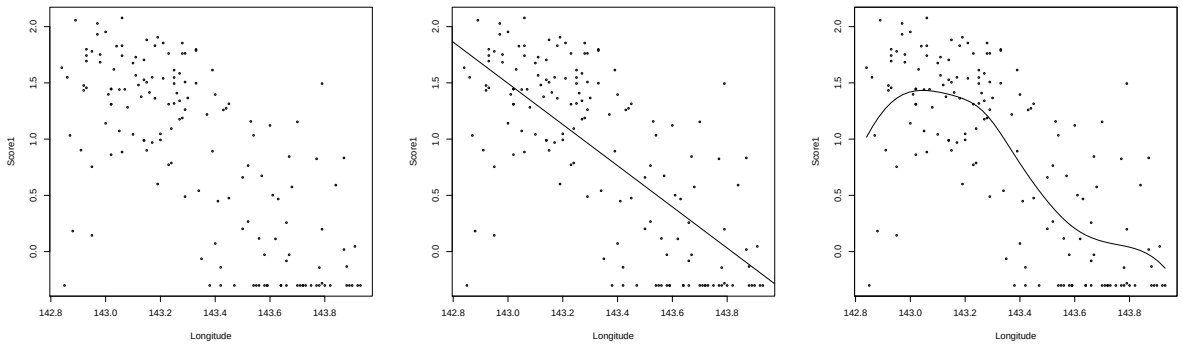
Example 1.3 (Great Barrier Reef data). A survey of the fauna on the sea bed lying between the coast of northern Queensland and the Great Barrier Reef was carried out. The sampling region covered a zone which was closed to commercial fishing, as well as neighbouring zones where fishing was permitted. The variables are:

Zone	an indicator for the closed (1) and open (0) zones
Year	an indicator of 1992 (0) or 1993 (1)
Latitude	latitude of the sampling position
Longitude	longitude of the sampling position
Depth	bottom depth
Score1	catch score 1
Score2	catch score 2

The details of the survey and an analysis of the data are provided by Poiner et al. (1997), *The effects of prawn trawling in the far northern section of the Great Barrier Reef*, CSIRO Division of Marine Research, Queensland Dept. of Primary Industries. ◁

The relationship between catch score (Score1) and longitude is of particular interest because, at this geographical location, the coast runs roughly north-south and so longitude is a proxy for distance offshore. We might therefore reasonably expect the abundance of marine life to change with longitude. The first of the three panels below shows that there is indeed a strong underlying negative relationship, with considerable variability also present. The middle panel summarises this in a simple linear regression

which captures much of this relationship. However, if we allow our regression model to be more flexible then a more complex relationship is suggested in the right hand panel, with a broadly similar mean level for some distance offshore followed by a marked decline, possibly followed by some levelling off thereafter. This gives valuable informal and graphical insight into the data but how can flexible regression models can be constructed and how can we use them to evaluate whether there is really evidence of non-linear behaviour in the data?



1.4.1 A local fitting approach

A simple nonparametric model has the form

$$y_i = m(x_i) + \varepsilon_i,$$

where the data (x_i, y_i) are described by a smooth curve m plus independent errors ε_i. One approach to fitting this is to take a model we know and fit it locally. For example, we can construct a *local linear regression*. This involves solving the least squares problem

$$\min_{\alpha,\beta} \sum_{i=1}^{n} \{y_i - \alpha - \beta(x_i - x)\}^2 \, w(x_i - x \, ; h)$$

and taking as the estimate at x the value of $\hat{\alpha}$, as this defines the position of the local regression line at the point x. This has an appealing simplicity and it can be generalised quite easily to other situations. This was the approach used to produce the nonparametric regression of the Reef data in the plot above.

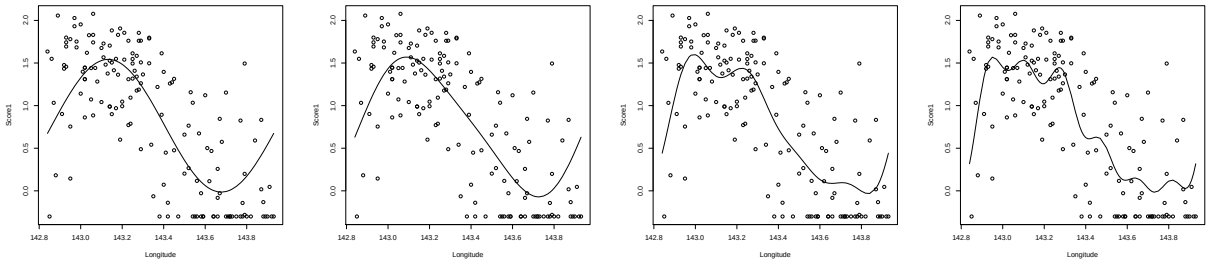
There is a variety of other ways in which smooth curve estimates can be produced and a further approach is outlined in the next section. It can sometimes reasonably be argued that the precise mechanism usually isn't too important and can be chosen for convenience.

1.4.2 Basis function approaches

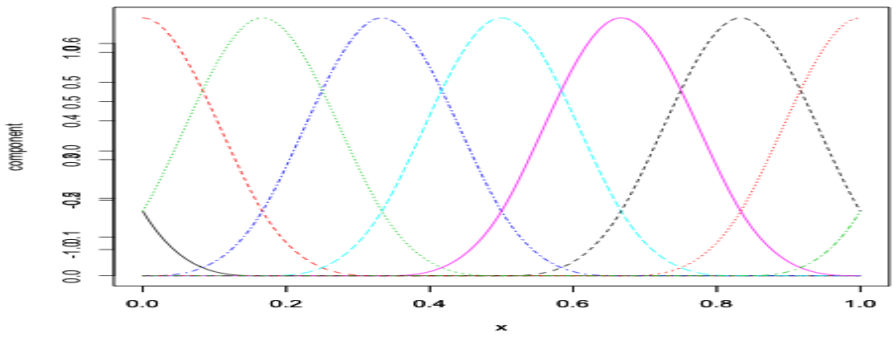
Basis approaches to function approximation have a very long history. One of the first was Fourier series, which is based on the expansion

$$m(x) \approx \frac{a_0}{2} + \sum_{j=1}^{p} a_j cos\left(\frac{2\pi jx}{P}\right) + b_j sin\left(\frac{2\pi jx}{P}\right)$$

where P is the range of the covariate. The plots below show the effects of fitting a Fourier series with $p = 1, 2, 4, 6$ frequencies.



Each basis function in a Fourier expansion has effects across the entire range of the data, with the result that the effect of fitting data well in one part of the sample space can create artefacts elsewhere. An alternative approach is to use a set of basis functions which are more local in their effects. The figure below shows a set of *b-spline* functions, $b_1(x), \ldots, b_p(x)$ which can be used for this.



A curve estimate can then be produced simply by fitting the regression

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_p b_2(x_i) + \ldots + \beta_p b_p(x_i) + \varepsilon_i.$$

A popular approach, known as *p-splines*, uses a large number of basis functions but includes a penalty to control the smoothness of the resulting estimator. Specifically, the coefficients β_i are chosen to minimise

$$\sum_{i=1}^{n} \{y_i - \beta_0 - \beta_1 b_1(x_i) - \beta_p b_2(x_i) - \ldots - \beta_p b_p(x_i)\}^2 + \lambda P(\beta),$$

where the penalty function P might, for example, have the form

$$P(\beta) = \sum_{j=2}^{p} (\beta_j - \beta_{j-1})^2$$

or some other measure of roughness along the sequence of coefficients. Here the parameter λ controls the degree of smoothness of the resulting estimator. This idea of using a

penalty to find a good balance between fit to the data and smoothness of the estimator is a general theme which will reappear at various points during the course.

A useful connection can be made here with wavelets, where local basis functions are also used to construct sophisticated function estimators.

1.5 Further illustrations of smoothing

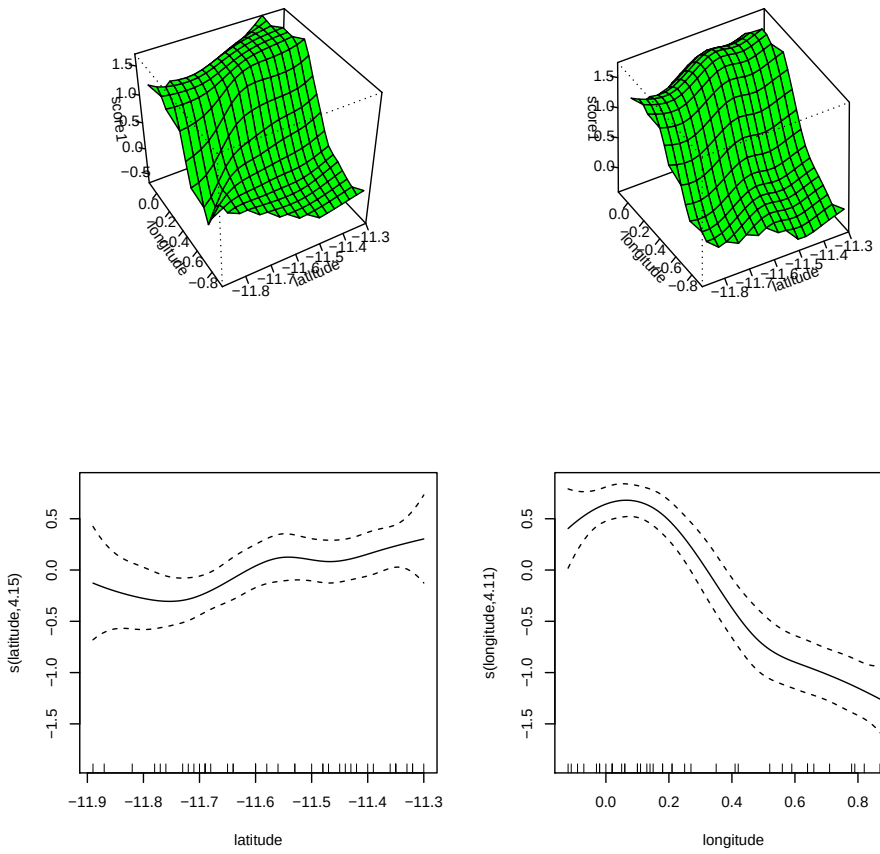
This section gives a few brief illustrations of how the ideas outlined in simple settings can be extended to much more complex situations and data structures. Some of these will be revisited in greater detail later in the course.

1.5.1 Regression with more than one covariate

It is rare to have problems which involve only a single covariate. For the Reef data a natural extension is to look at the relationship between the catch score and both latitude (x_1) and longitude (x_2), in a model

$$y_i = m(x_{1i}, x_{2i}) + \varepsilon_i.$$

the top left hand panel of the plot below shows the effect of this. The effect of longitude dominates, as we see from the earlier nonparametric regression. However, a small effect of latitude is also suggested. The methods by which surfaces of this type can be constructed will be discussed in the next two chapters.



It would be unrealistic to generalise this much further, by modelling additional covariates through functions of ever-increasing dimension. A very powerful approach is to construct *additive models* of the form

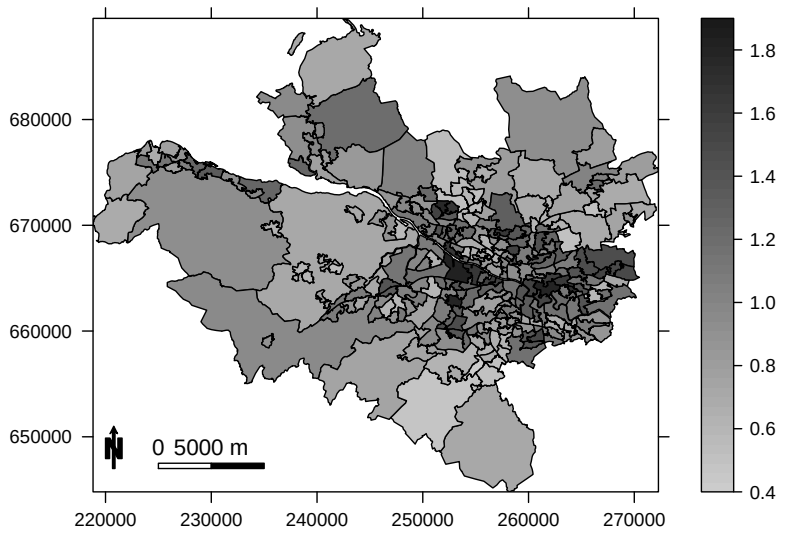
$$y_i = m_1(x_{1i}) + m_2(x_{2i}) + \varepsilon_i,$$

where the component functions m_1 and m_2 describe the separate and additive effects of the two covariates. Estimated additive model components, and their combined effects as an additive surface, are displayed in the other panels of the figure above. Methods for fitting models of this type will also be discussed later.

1.5.2 Areal data

Data quantifying population-level summaries of disease prevalence for n non-overlapping areal units are available from both the English and Scottish Neighbourhood Statistics databases. They are used in many different applications, including quantifying the effect of an exposure on health, and identifying clusters of areal units that exhibit elevated risks of disease. The health data are denoted by $\mathbf{Y} = (Y_1, \ldots, Y_n)$ and $\mathbf{E} = (E_1, \ldots, E_n)$, which are the observed and expected numbers of disease cases in each areal unit. The covariates are denoted by an $n \times p$ matrix $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, and could include environmental exposures or measures of socio-economic deprivation.

One example concerns the prevalence of respiratory disease in Glasgow in 2010, where interest focusses on the hospitalisation risk $\text{SIR}_k = Y_k/E_k$. The data are displayed in the plot below.



A suitable statistical model is

$$\begin{aligned} Y_k &\sim \text{Poisson}(E_k R_k), \\ \log(R_k) &= \mathbf{x}_k^{\mathrm{T}}\boldsymbol{\beta} + \phi_k, \\ \phi_k|\boldsymbol{\phi}_{-k}, \tau^2, W &\sim \text{N}\left(\frac{\sum_{i=1}^n w_{ki}\phi_i}{\sum_{i=1}^n w_{ki}}, \frac{\tau^2}{\sum_{i=1}^n w_{ki}}\right), \end{aligned}$$

where

– R_k quantifies disease risk in area k.

– $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_n)$ are random effects that model residual spatial autocorrelation and improve estimation of disease risk by smoothing the risks across neighbouring areas.

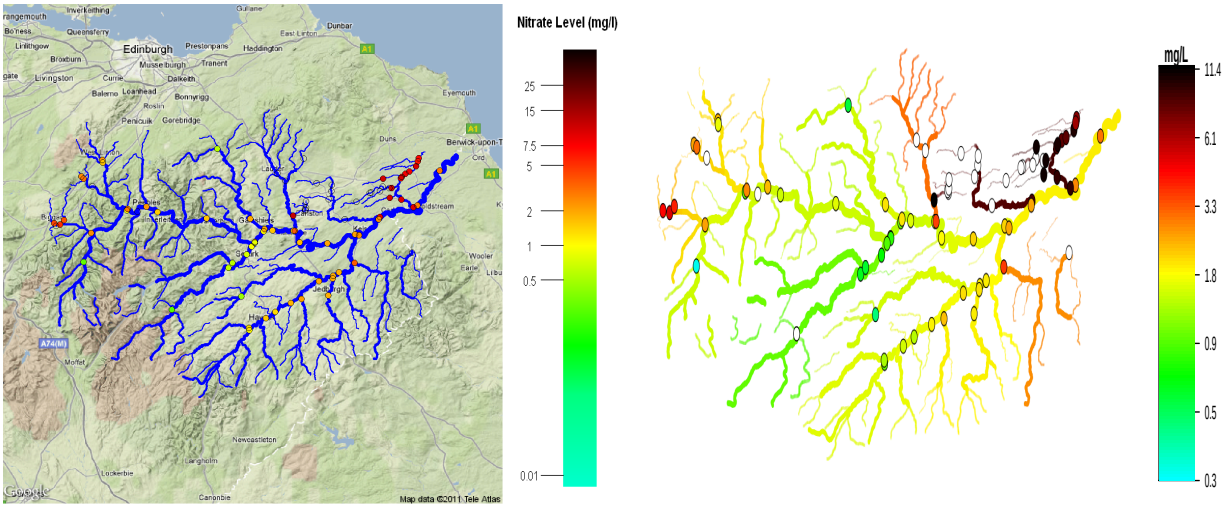
– Commonly, conditional autoregressive (CAR) models are used for this smoothing, where $W = (w_{ki})$ is a binary $n \times n$ neighbourhood matrix.

Material kindly provided by Duncan Lee, University of Glasgow.

1.5.3 Network data

Models for spatial data often involve some form of smoothing, either implicitly or explicitly. An interesting forms of spatial data arises from river networks, where models for the variation in measurements should respect both the spatial pattern of the interconnecting water channels as well as the mixing effect of confluence points where channels meet. The data below refer to nitrate pollution in the River Tweed. The point measurements on the left can be used to construct a spatial model for the whole network, whose

predictions are shown in the panel on the right.



This example arises from joint work with Alastair Rushworth, David O'Donnell, Marian Scott, Mark Hallard (SEPA).

2

Fitting models locally

2.1 Local mean and local linear estimators

In chapter 1, the idea of fitting a linear model locally was introduced. In fact, there is an even simpler approach by fitting a local mean. Specifically, at any point of interest x, we choose our estimator of the curve there as the value of μ which minimises

$$\sum_{i=1}^{n}\{y_i-\mu\}^2 w(x_i-x;h)$$

and this is easily shown to produce the 'running mean'

$$\hat{m}(x)=\frac{\sum_{i=1}^{n} w(x_i-x;h)\,y_i}{\sum_{i=1}^{n} w(x_i-x;h)}.$$

If we do the algebra to minimise the sum-of-squares given in Chapter 1 to define the local linear approach, then an explicit formula for the local estimator can be derived as

$$\hat{m}(x)=\frac{1}{n}\sum_{i=1}^{n}\frac{\{s_2(x;h)-s_1(x;h)(x_i-x)\}w(x_i-x;h)y_i}{s_2(x;h)s_0(x;h)-s_1(x;h)^2},$$

where $s_r(x;h)=\{\sum(x_i-x)^r w(x_i-x;h)\}/n$.

In both the local mean and the local linear cases, the estimator is seen to be of the form $\sum_i \kappa_i y_i$, where the weights κ_i sum to 1. There is a broad sense then in which even the local linear method is 'locally averaging' the data. In fact, many other forms of nonparametric regression can also be formulated in a similar way.

2.2 Some simple properties

One question which immediately arises is whether it matters very much which form of nonparametric smoothing is used. Sometimes computational and other issues may

constrain what choices are practical. However, if we take the simple local mean and local linear examples, what principles can we use to guide our choice? Deriving expression which capture simple properties such as bias and variance is an obvious place to start.

We will start with the local mean estimator. The exploration will be a little informal, without the full technicality of formal proofs (although these could be added if time permitted). The aim is to identify the properties of the estimator in conceptual form. If the numerator and denominator of the local mean estimator are both scaled by $1/n$, then the denominator has a familiar form, namely a kernel density estimator. As we saw earlier, this has expectation

$$\mathbb{E}\left\{\frac{1}{n}\sum_i w(x_i - x; h)\right\} = f(x) + \frac{h^2}{2}f''(x) + o(h^2),$$

where, as before, we assume for convenience that the kernel function can be rewritten as $\frac{1}{h}w((x_i - x)/h)$ and w is a symmetric probability density function around 0 with variance 1. Turning now to the numerator, we have

$$\begin{aligned}
\mathbb{E}\left\{\frac{1}{n}\sum_i w(x_i - x; h)\, y_i\right\} &= \frac{1}{n}\sum_i \frac{1}{h}w\left(\frac{x_i - x}{h}\right) m(x_i) \\
&\approx \int \frac{1}{h}w\left(\frac{z - x}{h}\right) m(z)f(z) \quad \text{[integral approximation]} \\
&= \int w(u)m(x + hu)f(x + hu)du \quad \text{[change of variable]}
\end{aligned}$$

Now apply a Taylor series expansion to the terms involving $x + hu$, to give

$$\begin{aligned}
m(x + hu) &= m(x) + hu\, m'(x) + \frac{(hu)^2}{2}m''(x) + o(h^2) \\
f(x + hu) &= f(x) + hu\, f'(x) + \frac{(hu)^2}{2}f''(x) + o(h^2)
\end{aligned}$$

Substituting these in and integrating over u gives

$$\mathbb{E}\left\{\frac{1}{n}\sum_i w(x_i - x; h)\, y_i\right\} \approx m(x)f(x) + h^2\left\{\frac{1}{2}f(x)m''(x) + m'(x)f'(x) + \frac{1}{2}f''(x)m(x)\right\} + o(h^2).$$

Dividing both numerator and demoninator by $f(x)$ gives

$$\begin{aligned}
\text{numerator:} \quad & m(x) + h^2\left\{\frac{1}{2}m''(x) + m'(x)\frac{f'(x)}{f(x)} + \frac{1}{2}\frac{f''(x)}{f(x)}m(x)\right\} + o(h^2) \\
\text{denominator:} \quad & 1 + \frac{h^2}{2}\frac{f''(x)}{f(x)} + o(h^2)
\end{aligned}$$

The dominant term in the mean of the ratio of numerator and denominator is the ratio of the means. Applying the series expansion for $(1 + x)^{-1}$ allows the reciprocal of the denominator to be written as

$$1 - \frac{h^2}{2}\frac{f''(x)}{f(x)} + o(h^2)$$

Multiplying the different terms out, we have

$$\begin{aligned}\mathbb{E}\{\hat{m}(x)\} \approx & \left\{m(x) + h^2\left\{\frac{1}{2}m''(x) + m'(x)\frac{f'(x)}{f(x)} + \frac{1}{2}\frac{f''(x)}{f(x)}m(x)\right\} + o(h^2)\right\} \\ & \left\{1 - \frac{h^2}{2}\frac{f''(x)}{f(x)} + o(h^2)\right\} \\ = & \; m(x) + h^2\left\{\frac{1}{2}m''(x) + \frac{m'(x)f'(x)}{f(x)}\right\} + o(h^2)\end{aligned}$$

Phew!

A similar sequence of manipulations (which you might like to try on your own) gives an asymptotic expression for the variance as

$$\mathsf{var}\{\hat{m}(x)\} \approx \frac{1}{nh}\left\{\int w(u)^2 du\right\}\sigma^2\frac{1}{f(x)},$$

where σ^2 denotes the variance of the error terms ε_i.

In the local linear case, the estimator can be written as $\sum_i a_i y_i / \sum_i a_i$, where $a_i = \frac{1}{n}\frac{1}{h}w(\frac{x_i - x}{h})\{s_2 - (x_i - x)s_1\}$. Consider first s_1, which can be written as

$$\begin{aligned}s_1 &= \frac{1}{n}\sum_j \frac{1}{h}w\left(\frac{x_i - x}{h}\right)(x_j - x) \\ &\approx \int \frac{1}{h}w\left(\frac{x - x}{h}\right)f(z)(z - x)dz \\ &= \int w(u)hu\{f(x) + huf'(x) + o(h)\}du \\ &= h^2 f'(x) + o(h^2)\end{aligned}$$

By a similar argument,

$$s_2 \approx h^2 f(x) + o(h^2).$$

The weights a_i can then be approximated by

$$a_i \approx \frac{1}{n}\frac{1}{h}w\left(\frac{x_i - x}{h}\right)h^2\{f(x) - (x_i - x)f'(x)\}.$$

The mean of the estimator is $\mathbb{E}\{\hat{m}(x)\} = \sum_i a_i m(x_i) / \sum_i a_i$. Ignoring the term h^2 which cancels in the ratio, the numerator can be expressed as

$$\left\{f(x)^2 + \frac{h^2}{2}f(x)f'(x) - h^2 f'(x)^2\right\}m(x) + \frac{h^2}{2}f(x)^2 m''(x)^2,$$

after an integral approximation, a change of variable and a Taylor series expansion. By a similar argument, the denominator of $\mathbb{E}\{\hat{m}(x)\}$ can be approximated by

$$f(x)^2 + \frac{h^2}{2}f(x)f''(x) - h^2 f'(x)^2.$$

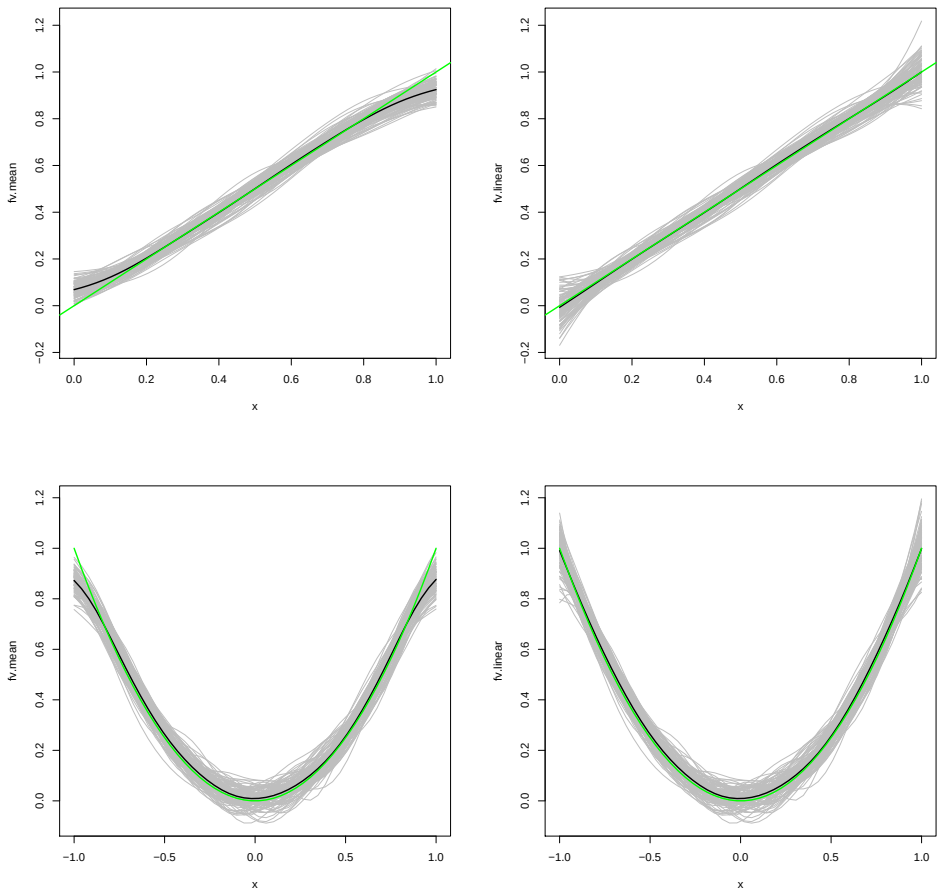
The principal term of the ratio then gives

$$\mathbb{E}\{\hat{m}(x)\} \approx m(x) + \frac{h^2}{2} m''(x).$$

So, after considerable work, a very simple expression has been achieved. Similar manipulations for the variance produces an expression which is exactly the same as that for the variance of the local mean estimator.

A comparison of the expressions for the local mean and local linear estimators is interesting. For example, the principal terms in the expression for the mean of the local linear estimator is not only simpler but also does not involve $f(x)$, both of which are attractive properties. This is one of the reasons that the local linear estimator is generally preferred over the local mean.

However, another issue concerns edge effects. These require more careful analysis to identify so, instead, we will use a simple illustration based on simulation. The figures below show the results of repeatedly simulating 50 data points, equally spaced over $[0, 1]$, from the model $y = x + \varepsilon$, where the standard deviation of the error terms is 0.1. For each set of simulated data, a nonparametric regression curve is plotted, using local mean (left) and local linear (right) estimators. Notice that at the ends of the sample space the local mean has strong bias, because there is data only on one side of the estimation point of interest. In contrast, the local linear method is unaffected. The same pattern is displayed in the lower plots, using the model $y = x^2 + \varepsilon$ over the range $[-1, 1]$.



With a little more theoretical work, a central limit theorem can be constructing to show that

$$\frac{\hat{m}(x) - m(x) - b(x)}{\sqrt{v(x)}} \to N(0,1),$$

where $b(x)$ and $v(x)$ denote the bias band variance of $\hat{m}(x)$.

Following on from the discussions in density estimation, the performance of a nonparametric estimator can be summarised in the *mean integrated squared error*, defined as

$$MISE = \int \mathbb{E}\{\hat{m}(x) - m(x)\}^2 f(x)dx$$

and an optimal smoothing parameter can be defined as the value of h which minimises the asymptotic approximation of MISE, namely

$$h_{\text{opt}} = \left\{ \frac{\gamma(w)\sigma^2}{\int [m''(x)]^2 f(x)dx} \right\}^{1/5} n^{-1/5}$$

If we use this optimal smoothing parameter, then both the bias and the square root of the variance, which determines the rate of convergence, are of order $n^{-2/5}$. Notice that this rate of convergence is slower than the $n^{-1/2}$ which applies for parametric models.

Fan and Gijbels (1996) give further details on the theoretical aspects of local polynomial smoothing.

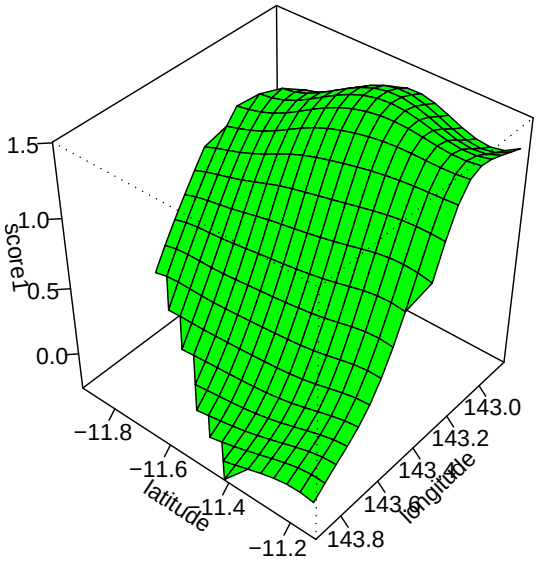
2.3 Smoothing in two dimensions

It can often be of interest to smooth over two covariates simultaneously, for example when dealing with a response variable defined over geographical co-ordinates. The local linear approach is particularly easy to extend to this setting. If the observed data are denoted by $\{x_{1i}, x_{2i}, y_i; i = 1, \ldots, n\}$, then for estimation at the point (x_1, x_2) the weighted least squares formulation is

$$\min_{\alpha,\beta,\gamma} \sum_{i=1}^{n} \{y_i - \alpha - \beta(x_{1i} - x_1) - \gamma(x_{2i} - x_2)\}^2 w(x_{1i} - x_1; h_1)\, w(x_{2i} - x_2; h_2).$$

The value of the fitted surface at (x_1, x_2) is simply $\hat{\alpha}$. With careful thought, the computation can be performed efficiently.

This is illustrated below with one year of Reef data from the closed zone.



2.4 Degrees of freedom and standard errors

It is helpful to express the fitted values of the nonparametric regression as

$$\hat{m} = Sy,$$

where $\hat{m}$ denotes the vector of fitted values, S denotes a *smoothing matrix* whose rows consist of the weights appropriate to estimation at each evaluation point, and y denotes the observed responses in vector form. This linear structure is very helpful.

For example, it gives us a route to defining *degrees of freedom* by analogy with what happens with the usual linear model, where the number of parameters is the trace of the projection matrix. An approximate version of these can be constructed for nonparametric models as

$$\mathrm{df} = \mathrm{tr}\left\{S\right\}.$$

Similarly, we can construct an estimate of the error variance σ^2 through the residual sum-of-squares, which in a nonparametric setting is simply

$$\mathrm{RSS} = \sum \{y_i - \hat{m}(x_i)\}^2.$$

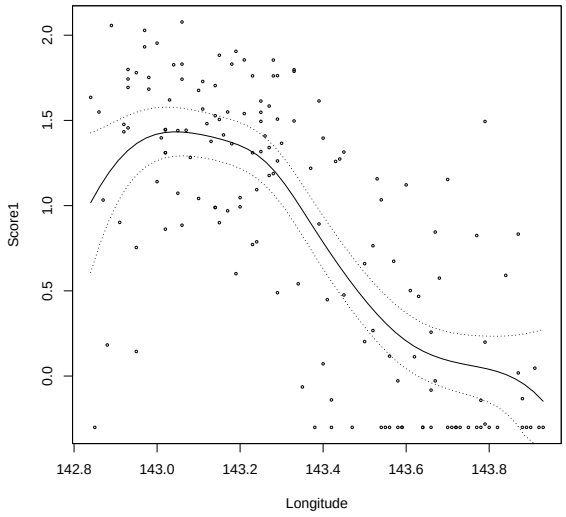
This leads to the estimator of the error variance

$$\hat{\sigma}^2 = \mathrm{RSS}/\mathrm{df}.$$

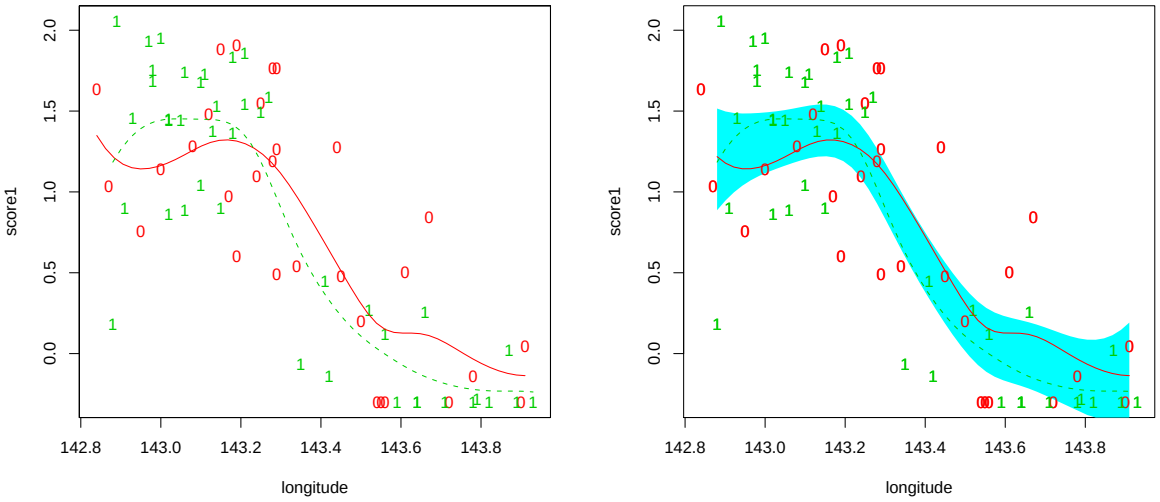
The linear structure of the fitted values also makes it very easy to produce standard errors which quantify the variability of the estimate at any value of x. Unfortunately, we can't easily produce confidence intervals for the curve because of the bias mentioned above. However, by adding and subtracting two standard errors at each point on the curve we can produce *variability bands* which express the variation in the curve estimate. In fact, we don't need to rely on the asymptotic formula for variance. If $\hat{m}$ denotes the estimated values of m at a set of evaluation points then

$$\mathsf{var}\{\hat{m}\} = \mathsf{var}\{Sy\} = SS^T\sigma^2$$

and so, by plugging in $\hat{\sigma}^2$ and taking the square root of the diagonal elements, the standard errors at each evaluation point are easily constructed. The plot below illustrates this on the Reef data.



Sometime we want to compare curves, at least informally, and we can use the standard errors from each curve to do that. At any point x, the standard error of the difference between the curves is $se_d(x) = \sqrt{se_1(x)^2 + se_2(x)^2}$, where $se_1(x)$ and $se_2(x)$ denote the standard errors of each curve at that point. A neat trick is to plot a band whose width is $2se_d(x)$. By centring this band at the average of the two curves we can see where they are more than two standard errors apart. We can see this with the Reef data separated into two groups corresponding to open and closed fishing zones.



2.5 How much to smooth

One of the key questions with nonparametric models is how much smoothing to apply to the data. For exploratory work, it can often be helpful simply to experiment with different degrees of smoothing. One appealing way to do that is to specify how many *degrees of freedom* (see discussion above) you would like to have. This puts things on a natural scale.

However, in more complicated situations that can be difficult and it is helpful to have an automatic way of producing a suitable level of smoothing. There are several ways to do this, some of which are carefully tailored to particular models. Here we will outline a method called *cross-validation* which, although it has some difficulties, has the advantage that the generality of its definition allows it to be applied to quite a wide variety of settings. In the present setting, the idea is to choose h to minimise

$$\text{CV:} \quad \sum_{i=1}^{n} \{y_i - \hat{m}_{-i}(x_i)\}^2.$$

The subscript $-i$ denotes that the estimate of the smooth curve at x_i is constructed from the remainder of the data, excluding x_i. The aim then is to evaluate the level of smoothing through the extent to which each observation is predicted from the smooth curve produced by the rest of the data. The value of h which minimises the expression above should provide a suitable level of smoothing.

It is often convenient to use an approximation known as *generalised cross-validation* (GCV) which has the efficient computational form

$$\text{GCV:} \quad n\text{RSS}/\{\text{tr}\,\{I - S\}^2\}$$

.

The degree of smoothing can also be selected automatically by minimising a quantity based on *Akaike's information criterion*, namely

$$\text{AIC:} \quad \frac{\text{RSS}}{n} + 1 + \frac{2(\nu+1)}{(n-\nu-2)},$$

Other interesting approaches will be outlined in the next lecture.

3

Splines

3.1 Introduction

This chapter covers splines, which are one of the most popular tools for flexible modelling. This section discusses a number of more philosophical concepts, some of which we have already touched upon in chapter 2. Each of these issues will come up again when we look at the details of spline-based flexible regression later on in this chapter.

In parametric modelling (*e.g.* estimating the rate of a Poisson distribution, linear regression) we assume we know the data generating process up to a finite number of parameters. In flexible modelling we want to fit a function to data, without making such a strict parametric assumption. All we are willing to assume is typically that the function of interest is sufficiently smooth. More formally speaking, this corresponds to working with an infinite-dimensional parameter space. This flexible approach has a number of advantages, most importantly it is less likely that the model is mis-specified. However there is a price to pay. Estimation becomes more difficult.

Example 3.1. Figure 3.1 shows two smooth functions describing the relationship between the response Y_i and the covariate x_i. In this example both functions yield the same fitted values $\hat{y}_i = \hat{m}(x_i)$. This also implies that the least-squares loss $\sum_{i=1}^{n}(y_i - \hat{m}(x_i))^2$ is the same for both functions, i.e. the data alone does not tell us which function does a better job. There is no global answer to this question.

Which of the two functions appears better suited to us depends on the context and also to some extent our subjective choice. In most circumstances we would prefer the function in the left-hand panel as it is the "simpler" function. However, if we expect the signal to have a periodic component (say we are expecting a day-of-the-week effect) then we might prefer the function shown in the right-hand panel. ◁

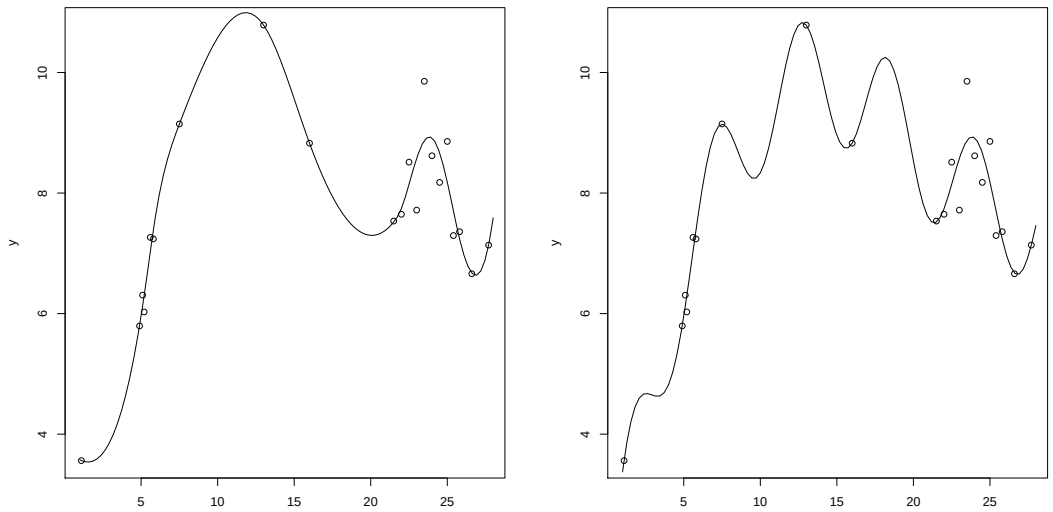


Figure 3.1. Two possible smooth functions modelling the relationship between the response Y_i and the covariate x_i. Note that both functions yield the same fitted values $\hat{y}_i = \hat{m}(x_i)$ and thus the same least-squares loss $\sum_{i=1}^{n}(y_i - \hat{m}(x_i))^2$.

What we have seen in the example is simply that the family of smooth functions is so large that observing a finite sample alone will not tell us enough to learn the function of interest $m(\cdot)$.

We need to provide additional information, which can be of different types:

– We can assume that the function of interest $m(\cdot)$ comes from a more restricted family of functions. We might even assume a rich class of parametric models. We will use this idea when we are looking at splines based on truncated power series and B-splines in section 3.2.4.
– We express a preference for some functions over others (without looking at the data) and use this in the model fitting procedure. Typically we prefer a smooth function to a more wiggly function. In a frequentist setting, this leads to penalty-based approach, or can be viewed as a Bayesian prior over the space of functions. We will discuss this in sections 3.2.3 and 3.3.

3.2 Univariate splines

3.2.1 Polynomial regression

We will start by revising polynomial regression. To fix notation, we quickly state the simple linear regression model

$$\mathbb{E}(Y_i) = \beta_0 + \beta_1 x_i \qquad \text{for } i = 1, \ldots, n,$$

or equivalently, in matrix-vector notation,

$$\mathbb{E}(\mathbf{y}) = \mathbf{B}\boldsymbol{\beta} \qquad \text{with } \mathbf{y} = (Y_1, \ldots, Y_n)^\top \text{ and } \mathbf{B} = \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}.$$

The simple linear regression model can be extended into a polynomial regression model by including powers of the covariates x_i in the design matrix. The polynomial regression model

$$\mathbb{E}(Y_i) = \beta_0 + \beta_1 x_i + \ldots + \beta_r x_i^r \qquad \text{for } i = 1, \ldots, n,$$

just corresponds to linear regression using the expanded design matrix

$$\mathbf{B} = \begin{pmatrix} 1 & x_1 & \ldots & x_1^r \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \ldots & x_n^r \end{pmatrix}.$$

We can then estimate $\boldsymbol{\beta}$ using the same techniques as used in multiple linear regression, i.e. the least-squares estimator is

$$\hat{\boldsymbol{\beta}} = (\mathbf{B}^\top \mathbf{B})^{-1} \mathbf{B}^\top \mathbf{y}$$

Polynomial regression is a very simple example of a basis expansion technique. We have simply replaced the design matrix of simple linear regression by an augmented design matrix. In the case of polynomial regression we have simply added powers of the x_i's. Many of the techniques covered in this chapter will be based in this idea of basis expansions.

Polynomial regression can be a useful tool if a polynomial of very low order yields a sufficient fit to the data.

Example 3.2 (Glucose levels in potatoes). Figure 3.2 shows a quadratic regression model fitted to a simple data set from an experiment in which the glucose level in potatoes was measured over the course of several weeks. Given the small number of observations there is little need to go beyond a simple quadratic regression model. ◁

However, polynomial regression is not very well suited for modelling more complex relationships, as the following example shows.

Example 3.3. Consider the data set simulated using the model

$$y_i = 1 - x_i^3 - 2\exp(-100x_i^2) + \varepsilon_i$$

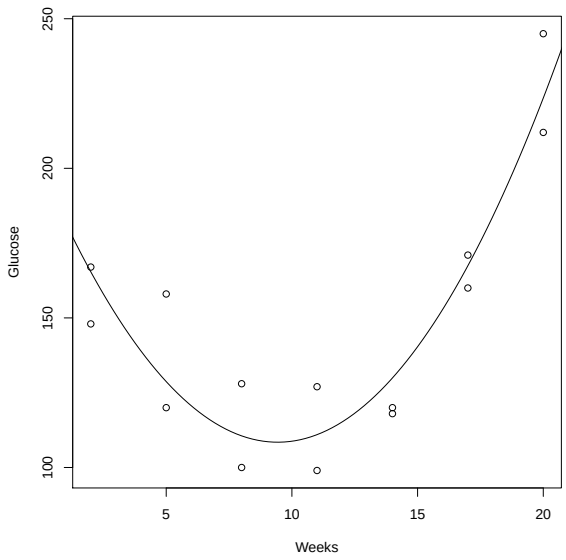


Figure 3.2. Glucose level in potatoes. The solid line is the fitted regression function obtained from quadratic regression.

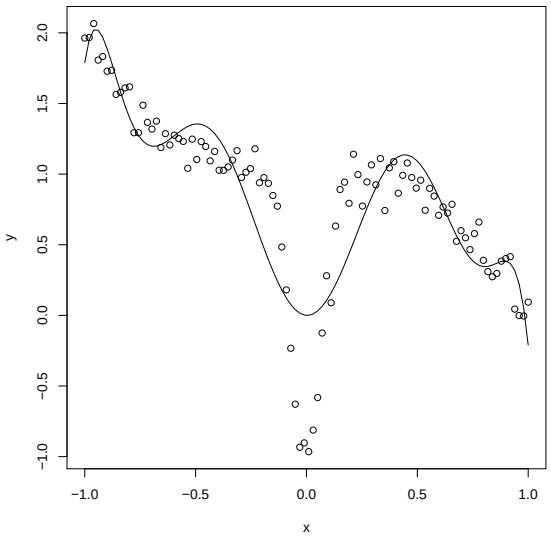
with $\mathbf{x} = (-1, -0.98, \ldots, 0.98, 1)$ and $\varepsilon_i \sim \mathsf{N}(0, 0.1^2)$. Figure 3.3(a) shows the data together with the fitted function obtained for a polynomial regression model of degree 10. The polynomial model of degree 10 is not flexible enough to capture the sharp dip around 0. If we increase the degree to 17 we can capture the dip better (panel (b)). However, the polynomial fit of degree 17 shows strong oscillations which are not supported by the data. Panel (c) shows the fitted regression function using a spline based model, which we will discuss later on in this chapter. The spline-based approach can capture the sharp dip much better and without yielding any oscillations.

Figure 3.4 allows some insight into why the polynomial model struggles. It shows `image` plots of the hat matrix $\mathbf{S} = \mathbf{B}(\mathbf{B}^\top\mathbf{B})^{-1}\mathbf{B}^\top$ for the three models under consideration. The hat matrix maps the observed response to the fitted response, i.e.

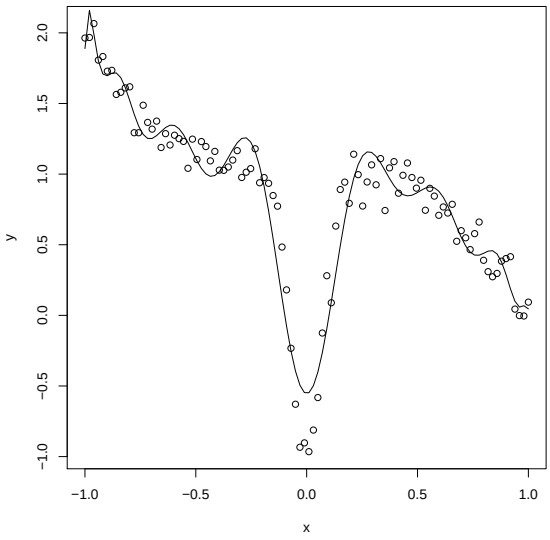
$$\hat{\mathbf{y}} = \mathbf{B}\hat{\boldsymbol{\beta}} = \mathbf{B}(\mathbf{B}^\top\mathbf{B})^{-1}\mathbf{B}^\top\mathbf{y} = \mathbf{S}\mathbf{y}$$

When performing flexible regression we would expect the prediction at x_i to almost only depend on observations close to x_i, i.e. we would expect the hat matrix $\mathbf{S}$ to be largely band-diagonal with a rather narrow band width. However, polynomials are not "local". As one can see from a Taylor series expansion, the coefficients of the polynomial can be learnt from higher order derivatives observed at a single point. The problem is that sharp dip provides more information than the data on either side of it, yielding to a poor fit on both sides. This is known as Runge's phenomenon in Numerical Analysis.

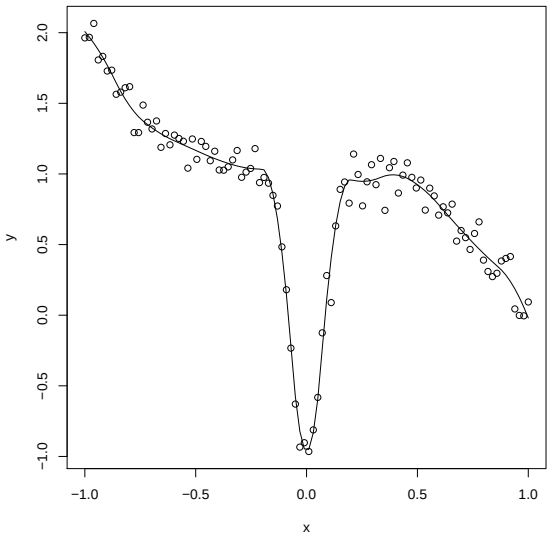
Figure 3.3(a) and figure 3.3(b) shows another drawback of polynomial regression. As $x \to \pm\infty$ the polynomial must go to $\pm\infty$ as well. This often leads to very high curvature at both ends of the range, which is typically not supported by the data.



(a) Polynomial regression of degree 10

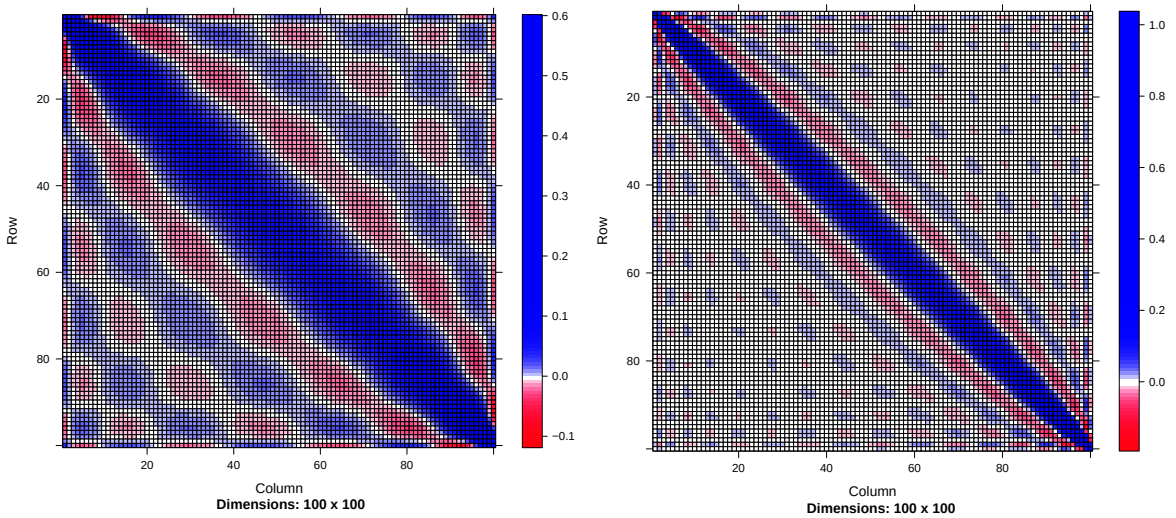


(b) Polynomial regression of degree 17



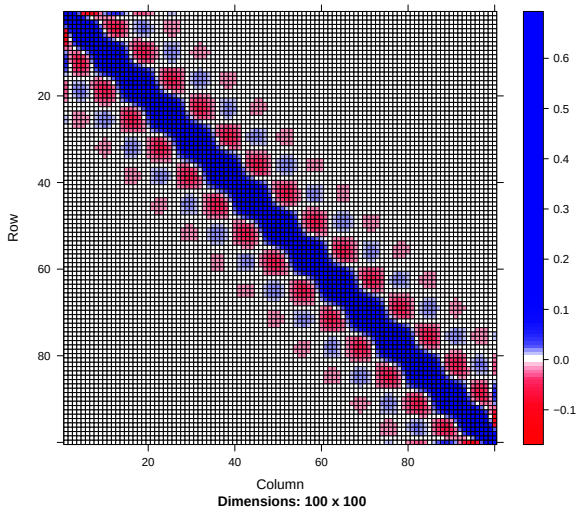
(c) Quadratic-spline-based regression

Figure 3.3. Data and fitted function for the simulated data from example 3.3 for polynomial regression of degrees 10 and 17 as well as for a spline-based model.



(a) Polynomial regression of degree 10

(b) Polynomial regression of degree 17

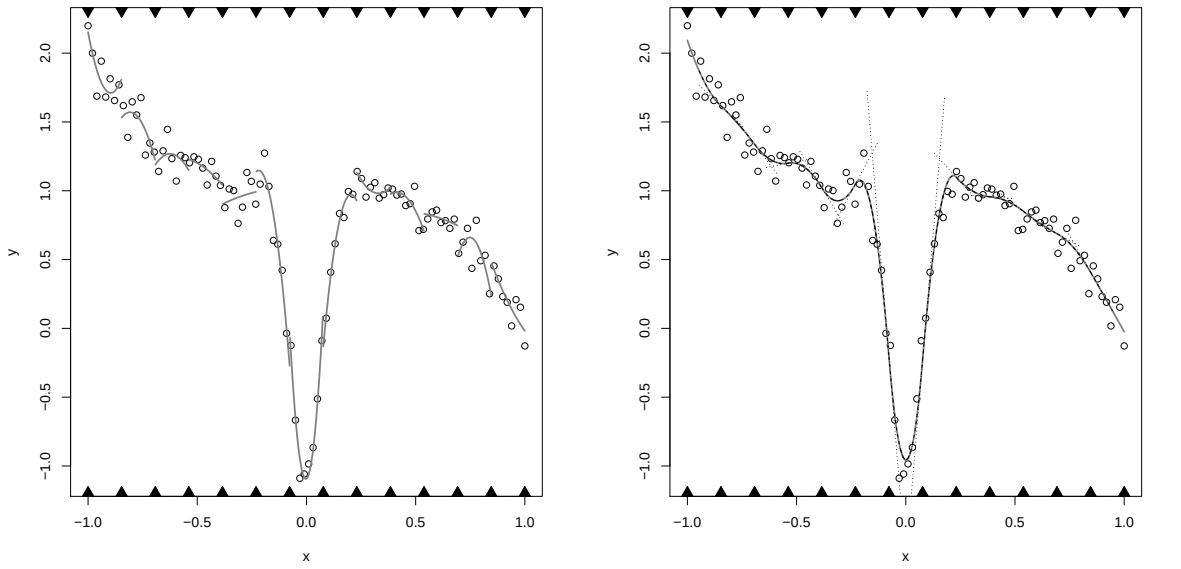


(c) Quadratic-spline-based regression

Figure 3.4. Hat matrix $\mathbf{S} = \mathbf{B}(\mathbf{B}^\top\mathbf{B})^{-1}\mathbf{B}^\top$ for polynomial regression of degrees 10 and 17 as well as for splines applied to the simulated data from example 3.3.

Yet another reason for avoiding polynomial regression is that it is highly likely to be numerically unstable. Due to the large correlations between the powers of the x_i, which make up the columns of the design matrix, the design matrix $\mathbf{B}$ and the matrix of cross-products $\mathbf{B}^\top\mathbf{B}$ is very likely to be ill-conditioned. The condition number[1] of $\mathbf{B}^\top\mathbf{B}$ for the polynomial regression model of degree 17 is 1.56×10^{12}, i.e. $\mathbf{B}^\top\mathbf{B}$ is barely invertible. For comparison, the corresponding condition number for the spline-based model is 32.49. ◁

As we have seen in the example above, polynomial regression is, unless modelling very simple relationships, not a suitable tool for flexible regression. In the next section we will consider piecewise polynomial models, which are better suited for flexible regression. These are based on the idea of splitting the input domain and fitting low-order polynomials in each interval. As we can see from figure 3.5(a) fitting polynomials independently of each other does not yield satisfactory results. We will thus introduce additional constraints which make the function continuous and (potentially) differentiable (cf. panel (b)).



(a) Discontinuous piecewise polynomials

(b) Piecewise polynomials which form a continuously differentiable function (derivatives at knots shown as dashed lines)

Figure 3.5. Piece-wise polynomials fitted to the data from example 3.3 with an without smoothness constraints. The back triangles show the positions of the knots.

[1] The condition number of a matrix is defined as the ratio of the largest singular value divided by the smallest singular value. For a symmetric positive-definite matrix this is the same as the ratio of the largest over the smallest eigenvalue. The condition number is of measure of how numerically unstable matrix operations like taking the inverse will be.

3.2.2 Polynomial splines

In this section we will introduce polynomial splines which are piecewise polynomials, which "glued together" at the knots so that the resulting function is r-times continuously differentiable.

Definition 3.1 (Polynomial spline). *Given a set of knots $a = \kappa_1 < \kappa_2 < \ldots < \kappa_l = b$, a function $m : [a, b] \to \mathbb{R}$ is called a (polynomial) spline of degree r if*

– $m(\cdot)$ is a polynomial of degree r on each interval (κ_j, κ_{j+1}) $(j = 1, \ldots, l-1)$.
– $m(\cdot)$ is $r-1$ times continuously differentiable.[2]

Historically, a spline was an elastic ruler used to draw technical designs, notably in shipbuilding and the early days of aircraft engineering. Figure 3.6 shows such a spline.[3]

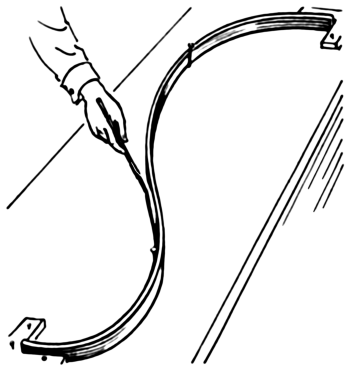


Figure 3.6. A spline.

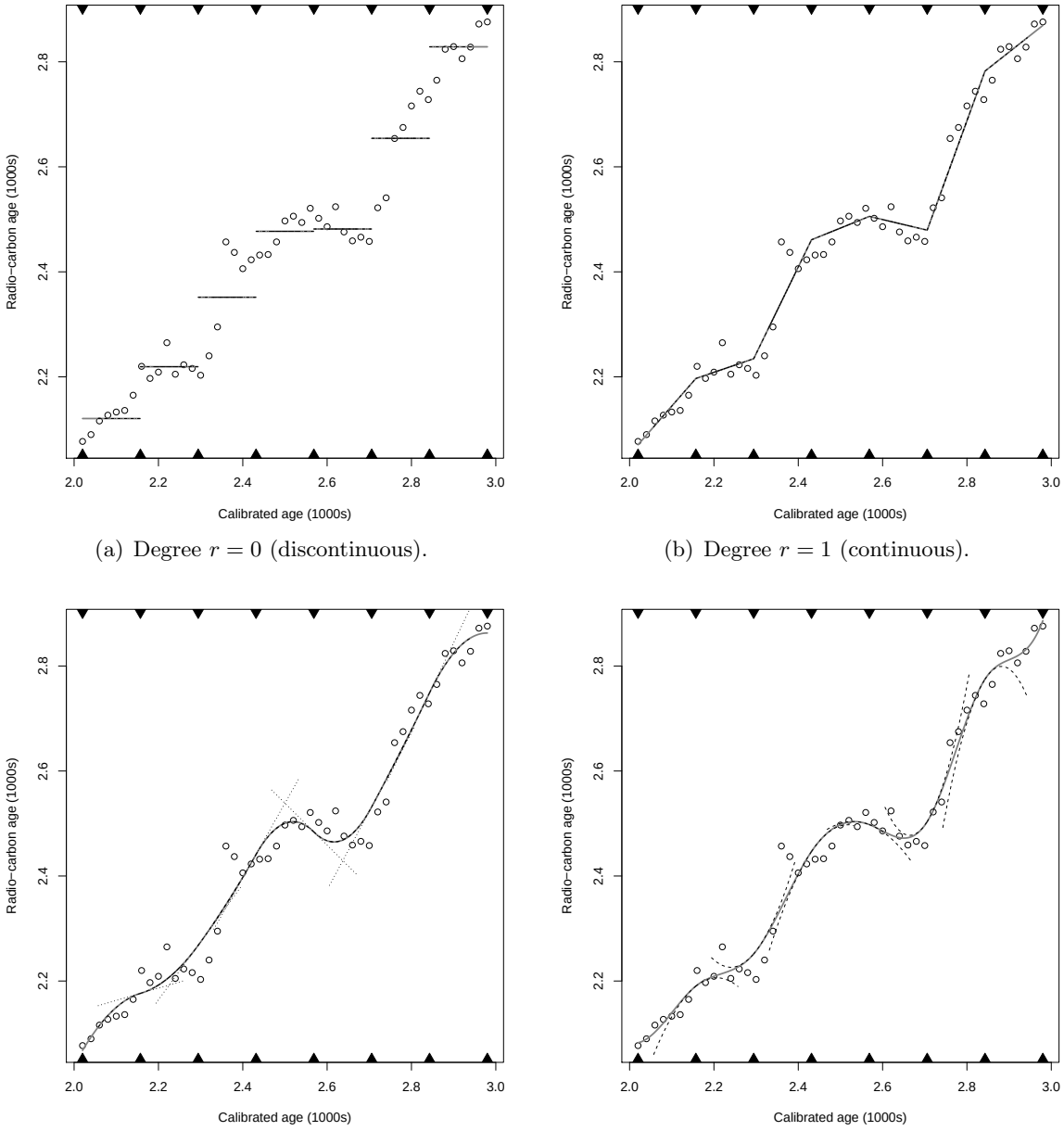
Choice of degree r. The degree r of the spline controls the smoothness in the sense of controlling its differentiability. For $r = 0$ the spline is a discontinuous step function. For $r = 1$ the spline is a polygonal line. For larger values of r the spline is increasingly smooth, but also behaves more and more like one global polynomial. It is worth noting that assuming too smooth a function can have significant detrimental effects on the fitted regression function (*e.g.* oscillations, ballooning). In practice it is rarely necessary to go beyond $r = 3$.

Example 3.4 (Radiocarbon dating). In a scientific experiment high-precision measurements of radiocarbon were performed on Irish oak. To construct a calibration curve

[2] For a spline of degree 0 the function $m(\cdot)$ does not need to be continuous. For a spline of degree 1 the function $m(\cdot)$ needs to be continuous, but does not need to be differentiable.

[3] See `http://pages.cs.wisc.edu/~deboor/draftspline.html` for a picture (probably from the 1960's) of a Boeing engineer using a spline.

we need to learn the relationship between the radiocarbon age and the calendar age. Figure 3.7 shows spline fits to the data using splines of different degrees. ◁



(a) Degree $r = 0$ (discontinuous).

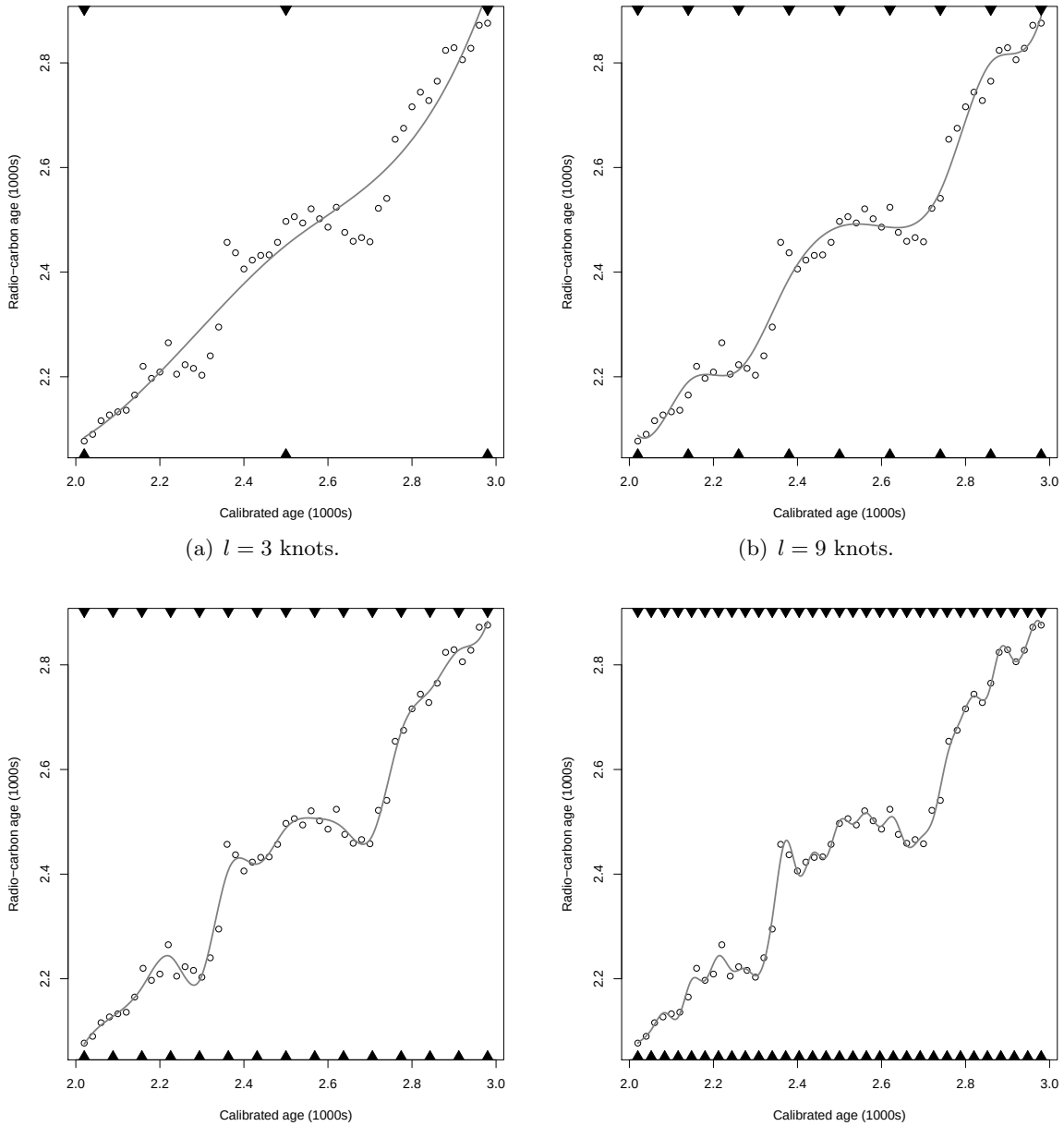
(b) Degree $r = 1$ (continuous).

(c) Degree $r = 2$ (continuous first derivative).

(d) Degree $r = 3$ (continuous second derivative).

Figure 3.7. Splines of degree $r \in \{0, 1, 2, 3\}$ fitted to the radiocarbon data.

Choice of the number of knots l. In an (unpenalised) spline the number of knots acts as a smoothing parameter. The more knots are used, the more flexible the regression function can become. As we have seen in chapter 2 a more flexible regression function has a lower bias, but a higher variance.



(a) $l = 3$ knots.

(b) $l = 9$ knots.

(c) $l = 15$ knots.

(d) $l = 31$ knots.

Figure 3.8. Cubic spline with different number of knots $l \in \{3, 9, 15, 31\}$ fitted to the radiocarbon data.

Example 3.5 (Radiocarbon dating (continued)). Figure 3.8 shows a cubic spline fitted to the radiocarbon data using an increasing number of knots. Too few knots lead to an underfit to the data: the fitted function does not fully represent the relationship between radiocarbon age and calendar age. Too many knots on the other hand lead to an overfit: the spline does not only pick up the signal, but also adapts to artefacts in the noise. ◁

Especially when the number of knots is small, the positioning of the knots can be important. The simplest strategy consist of using a set of equally spaced knots; this is computationally the simplest. Alternatively, we can place the knots according to the quantiles of the covariate. This makes the spline more flexible in regions with more data (and thus potentially more information) and less flexible in areas with less data (and potentially less information). A third strategy consists of trying to find an optimal placement of the knots. This usually is computationally very demanding.

Yet another approach consists of using "too many" knots — one knot per observation in the most extreme case — and use a penalty term to control for the smoothness. This avoid the need to select the number of knots altogether. We will study two such approaches in sections 3.2.3 and 3.3.

Splines as a vector space. For a given set of l knots and given degree r, the space of polynomial splines is a vector space, i.e. the sum of two splines as well as a scalar multiples of each spline are again splines. To find the dimension of the vector space have to find the number of "free parameters".

– Each polynomial has $r+1$ parameters and there are $l-1$ polynomials. Thus the spline model has $(r+1)\cdot(l-1)$ parameters. However we cannot choose all these parameters freely, as the resulting function needs to be $r-1$ times continuously differentiable.
– At the $l-2$ inside knots we have to guarantee that $m(\cdot)$ is $r-1$ times continuously differentiable. This corresponds to r constraints ($r-1$ constraints for each derivative and one for $m(\cdot)$ to be continuous). Thus there are $r\cdot(l-2)$ constraints (which are all linearly independent).

Thus there are $(r+1)\cdot(l-1)-r\cdot(l-2)=r+l-1$ free parameters. Thus the vector space of polynomial splines of degree r with l knots is $r+l-1$.

In section 3.2.4 we will explore different ways of constructing a basis for this space. The dimension will come in handy when proving that a given set of basis functions is indeed a basis of this space, as we only need to show that the basis functions are independent and that we use the correct number of basis functions.

Natural cubic splines. Finally, we will introduce the concept of a natural cubic spline. It is based on the idea that it is "safer" (or more "natural") to assume that the curvature

of the spline at the first and last knot is zero. If we were to extrapolate, we would then extrapolate linearly.

Definition 3.2 (Natural cubic spline). *A polynomial spline $m : [a, b] \to \mathbb{R}$ of degree 3 is called a natural cubic spline if $m''(a) = m''(b) = 0$.*

Given a set of l knots the vector space of all cubic splines has dimension $l+2$. Natural cubic splines introduce two additional constraints, thus they form a vector space of dimension l. This makes natural cubic splines perfectly suited for interpolation.

Proposition 3.3. *A set of l points (x_i, y_i) can be exactly interpolated using a natural cubic spline with the $x_1 < \ldots < x_l$ as knots. The interpolating natural cubic spline is unique.*

Proof. The space of natural cubic splines with knots at $x_1, \ldots, x_l$ is vector space of dimension l. Introducing l additional constraints ($y_i = m(x_i)$ for $i = 1, \ldots, l$) yields a system of l equations and l free parameters, which yields a unique solution.[4] □

In the next section we will show that natural cubic spline have an important optimality property.

3.2.3 Optimality of splines

This section provides a theoretical justification for the choice of splines for flexible regression.

In this section we will ask a rather general question. Given a data set (x_i, y_i) with $a \leq x_i \leq b$ we try to find, amongst all twice continuously differentiable functions, the function which "best" models the relationship between response y_i and covariate x_i.

First of all, we need to specify what we mean by "best". We could look for the function $m(\cdot)$ which yields the smallest least-squares criterion

$$\sum_{i=1}^{n} (y_i - m(x_i))^2$$

This is however not a good idea. Any function which interpolates all the observations (x_i, y_i) would be optimal in this sense, yet such a function would typically not describe

[4] Strictly speaking, we would need to show that the system of equations cannot be rank-deficient, which could cause the solution to be either non-unique or non-existing.

the relationship between x_i and y_i but rather model the artefacts of the random noise. Thus we will consider a so-called *penalised* (or *regularised*) criterion which tries to balance out two aspects which are important to us:

Fit to the data. We want $m(\cdot)$ to follow the data closely.

Simplicity/smoothness. We want the function $m(\cdot)$ not to be too complicated so that it generalises well to future unseen data.

We will thus the following penalised fitting criterion

$$\underbrace{\sum_{i=1}^{n}(y_i - m(x_i))^2}_{\text{Fit to the data}} + \lambda \underbrace{\int_a^b m''(x)^2 \, dx}_{\text{Roughness penalty}}, \tag{3.1}$$

where $\lambda > 0$ is a tuning parameter which controls the trade off between following the data and preventing $m(\cdot)$ from being too rough.

We will now show that the minimiser of (3.2) over all twice continuously differentiable functions has to be a natural cubic spline, i.e. natural cubic splines with knots at each of the unique x_i are in this sense the optimal class functions.

We will start by showing that natural cubic splines are optimal interpolators, in the sense that they minimise the roughness penalty $\int_a^b m''(x)^2 \, dx$.

Lemma 3.4. *Amongst all functions on $[a, b]$ which are twice continuously differentiable and which interpolate the set of points (x_i, y_i), a natural cubic spline with knots at the x_i yields the smallest roughness penalty*

$$\int_a^b m''(x)^2 \, dx.$$

Proof. Let $m(\cdot)$ be the natural cubic spline with knots at the x_i, interpolating the data. Suppose there is another function $g(\cdot)$, which is twice continuously differentiable and which also interpolates the data. Denote by $h(x) = g(x) - m(x)$ the difference between the two functions.

1. We will first of all show that we can decompose

$$\int_a^b g''(x)^2 \, dx = \int_a^b m''(x)^2 \, dx + \int_a^b h''(x)^2 \, dx$$

   i. As both $m(\cdot)$ and $g(\cdot)$ interpolate the (x_i, y_i) we have that $m(x_i) = g(x_i) = y_i$, thus $h(x_i) = g(x_i) - m(x_i) = 0$.

   ii. Using integration by parts we obtain that

$$\begin{aligned}\int_a^b m''(x)h''(x)\,dx &= \underbrace{[m''(x)h'(x)]_{x=a}^b}_{=0 \text{ (as } m''(a)=m''(b)=0)} - \int_a^b m'''(x)h'(x)\,dx \\ &= -\sum_{i=1}^{n-1}\int_{x_i}^{x_{i+1}} m'''(x)h'(x)\,dx \\ &= -\sum_{i=1}^{n-1} m'''\left(\frac{x_i+x_{i+1}}{2}\right)\cdot \underbrace{\int_{x_i}^{x_{i+1}} h'(x)\,dx}_{=\underbrace{h(x_{i+1})}_{=0}-\underbrace{h(x_i)}_{=0}=0} \\ &= 0\end{aligned}$$

In the second line we have used that the natural cubic spline is piece-wise cubic polynomial, i.e. between two knots x_i and x_{i+1} the third derivative $m'''(x)$ is constant.

iii. Thus

$$\begin{aligned}\int_a^b g''(x)^2\,dx &= \int_a^b (g''(x)-m''(x)+m''(x))^2\,dx = \int_a^b (h''(x)+m''(x))^2\,dx \\ &= \int_a^b h''(x)^2\,dx + 2\underbrace{\int_a^b h''(x)m''(x)\,dx}_{=0} + \int_a^b m''(x)^2\,dx \\ &= \int_a^b h''(x)^2\,dx + \int_a^b m''(x)^2\,dx\end{aligned}$$

2. Because of $\int_a^b h''(x)^2\,dx \geq 0$ we have that

$$\int_a^b g''(x)^2\,dx = \int_a^b m''(x)^2\,dx + \underbrace{\int_a^b h''(x)^2\,dx}_{\geq 0} \geq \int_a^b m''(x)^2,$$

i.e. the natural cubic spline cannot have a larger roughness penalty.

3. In the above inequality equality holds if and only if $\int_a^b h''(x)^2\,dx = 0$, which, given that $h(x_i)=0$, can only be the case if $g(x)=m(x)$ for all $x\in[a,b]$. □

Spline-based interpolation is implemented in the R functions `spline` and `splinefun`. We will now generalise the result about interpolation to the case of smoothing.

$$\underbrace{\sum_{i=1}^n (y_i - m(x_i))^2}_{\text{Model fit}} + \lambda \underbrace{\int_a^b m''(x)^2\,dx}_{\text{Roughness penalty}}, \tag{3.2}$$

where λ is a tuning parameter which controls the trade off between following the data and preventing $m(\cdot)$ from being too rough.

Theorem 3.5. *The minimiser of*

$$\sum_{i=1}^{n}(y_i - m(x_i))^2 + \lambda \cdot \int_a^b m''(x)^2 \, dx$$

amongst all twice continuously differentiable functions on $[a, b]$ *is given by a a natural cubic spline with knots in the unique* x_i*.*

This is an extremely powerful theorem. Even though we consider the entire infinite-dimensional vector space of all twice continuously differentiable functions, we only need to consider the finite-dimensional vector space of natural cubic splines. We have thus reduced the complexity of the optimisation problem to the comparatively simple problem of finding the optimal coefficients of the natural cubic spline. This can be done using least-squares.

Proof. Let $g(\cdot)$ be a twice continuously differentiable function. We will now create a competitor to $g(\cdot)$, which is a natural cubic spline with knots in the x_i. We will now show that, unless $g(\cdot)$ is already a natural cubic spline, $m(\cdot)$ leads to a smaller value of the objective function. We choose the natural cubic spline $m(\cdot)$ such that it interpolates the fitted values $g(\cdot)$ generates, i.e. $m(x_i) = g(x_i)$. Thus $\sum_{i=1}^{n}(y_i - m(x_i))^2 = \sum_{i=1}^{n}(y_i - g(x_i))^2$, i.e. both functions model the data equally well, however as we have shown in Lemma 3.4 the natural cubic spline $m(\cdot)$ has the smaller roughness penalty. □

Note that the proof did not make use of the fact that we have used the least-squares loss function. In fact, the theorem holds for any pointwise loss function.

The technique of *smoothing splines* is based on this theoretical result and finds the natural cubic spline minimising (3.2), and, due to the theorem, the optimal function amongst all twice continuously differentiable functions. This approach is implemented in the Rfunction `smooth.spline`. We will study this method in more detail in section 3.3.

3.2.4 Constructing splines

In this section we will studies two ways of constructing a basis for the vector space of polynomial splines: the truncated power basis and the B-spline basis. We will only cover the case of generic polynomial splines. However one can modify these bases to only span the space of natural cubic splines.

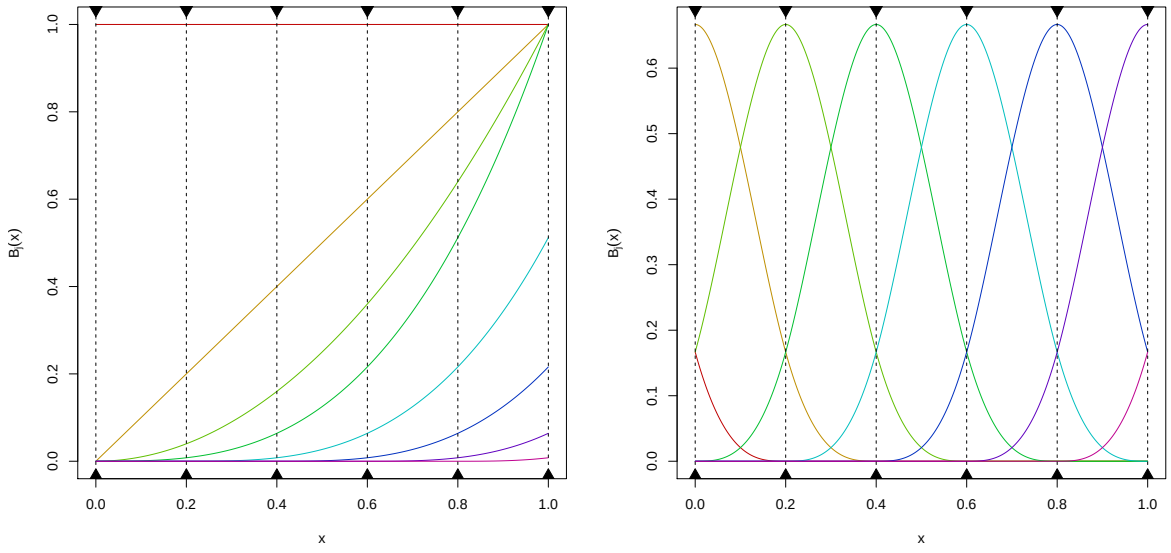
Truncated power basis. The simplest basis for polynomial splines is the truncated power basis.

Definition 3.6 (Truncated power basis). *Given a set of knots* $a = \kappa_1 < \ldots < \kappa_l = b$ *the truncated power basis of degree* r *is given by*

$$\left(1, x, \ldots, x^{r-1}, (x-\kappa_1)_+^r, (x-\kappa_2)_+^r, \ldots, (x-\kappa_{l-1})_+^r\right),$$

where $(z)_+^r = \begin{cases} z^r & \text{for } z > 0 \\ 0 & \text{otherwise.} \end{cases}$

The truncated power basis has $r + l - 1$ basis functions. It is easy to see that they are linearly independent. Thus the truncated power basis is indeed a basis of the vector space of polynomial splines. Figure 3.9 shows the truncated power series basis of degree 3 for six equally spaced knots.



(a) Truncated power series (b) B-splines

Figure 3.9. Basis functions $B_j(x)$ of the cubic truncated power series basis (left panel) and B-splines (right panel). The vertical lines indicate the location of the knots.

To fit a polynomial spline to data we can exploit the fact the truncated power basis is a basis of the vector space of polynomial splines of the given degree and with the given set of knots. Thus we can write any spline $m(\cdot)$ as a linear combination of the basis functions, i.e.

$$m(x) = \beta_0 + \beta_1 x + \ldots + \beta_{r-1} x^{r-1} + \beta_r (x-\kappa_1)_+^r + \ldots + \beta_{r+l-2} (x-\kappa_{l-1})_+^r$$

We can thus find the optimal spline $m(\cdot)$ by just finding the optimal set of coefficients β_j, which is nothing other than a linear regression problem with design matrix

$$\mathbf{B} = \begin{pmatrix} 1 & x_1 & \dots & x_1^{r-1} & (x_1-\kappa_1)_+^r & \dots & (x_1-\kappa_{l-1})_+^r \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \dots & x_n^{r-1} & (x_n-\kappa_1)_+^r & \dots & (x_n-\kappa_{l-1})_+^r \end{pmatrix}$$

We can use the design matrix $\mathbf{B}$ in exactly the same way as we would use the design matrix of a classical linear model.

We can interpret the truncated power series as a regression model in which the leading coefficient changes at each knot. At each knot, the remaining coefficients change as well. However they are fully constrained by the condition that the spline has to be $r-1$ times continuously differentiable at each knot.

Example 3.6 (Radiocarbon data (continued)). Figure 3.10 illustrates the use of a truncated power series basis for fitting a spline-based flexible regression model for the radiocarbon data.

As one can see from the middle panel of figure 3.10 and from figure 3.11, some of the estimated coefficients are very large: some of the basis functions are scaled up by a factor of more than 1000, with "neighbouring" basis functions having opposite signs. The reason for this is the high correlation between the columns of the design matrix of the truncated power series. The largest correlation between columns is 0.99921, which is very close to 1.

Figure 3.12 shows a scree plot of the singular values of the design matrix $\mathbf{B}$. The condition number of the matrix $\mathbf{B}$ is 225333.0, with the condition number of $\mathbf{B}^\top\mathbf{B}$ being $5,857,413,839$, i.e. $\mathbf{B}^\top\mathbf{B}$ is close to being numerically singular. This suggests that finding the least-squares estimate of the coefficients is close to being numerically unstable. ◁

As we have seen in the above example the truncated power basis can easily lead to numerical instability. Thus we will turn to an alternative basis, the so-called B-spline basis.

B-splines. B-splines form a numerically more stable basis. They also make the definition of meaningful penalty matrices easier, which we will exploit in section 3.3.

The key idea of B-splines is to use basis functions which are local, i.e. only non-zero for a "small" proportion of the range of the covariate and which are bounded above. We can think of B-splines as a sequence of "bumps". Figure 3.13 shows a B-spline basis function for degrees $r \in \{0,1,2,3\}$. We will define B splines recursively.

Definition 3.7 (B-spline basis). *(a) Given a set of l knots the B-spline basis of degree 0 is given by the functions $(B_1^0(x),\dots,B_{l-1}^0)$ with*

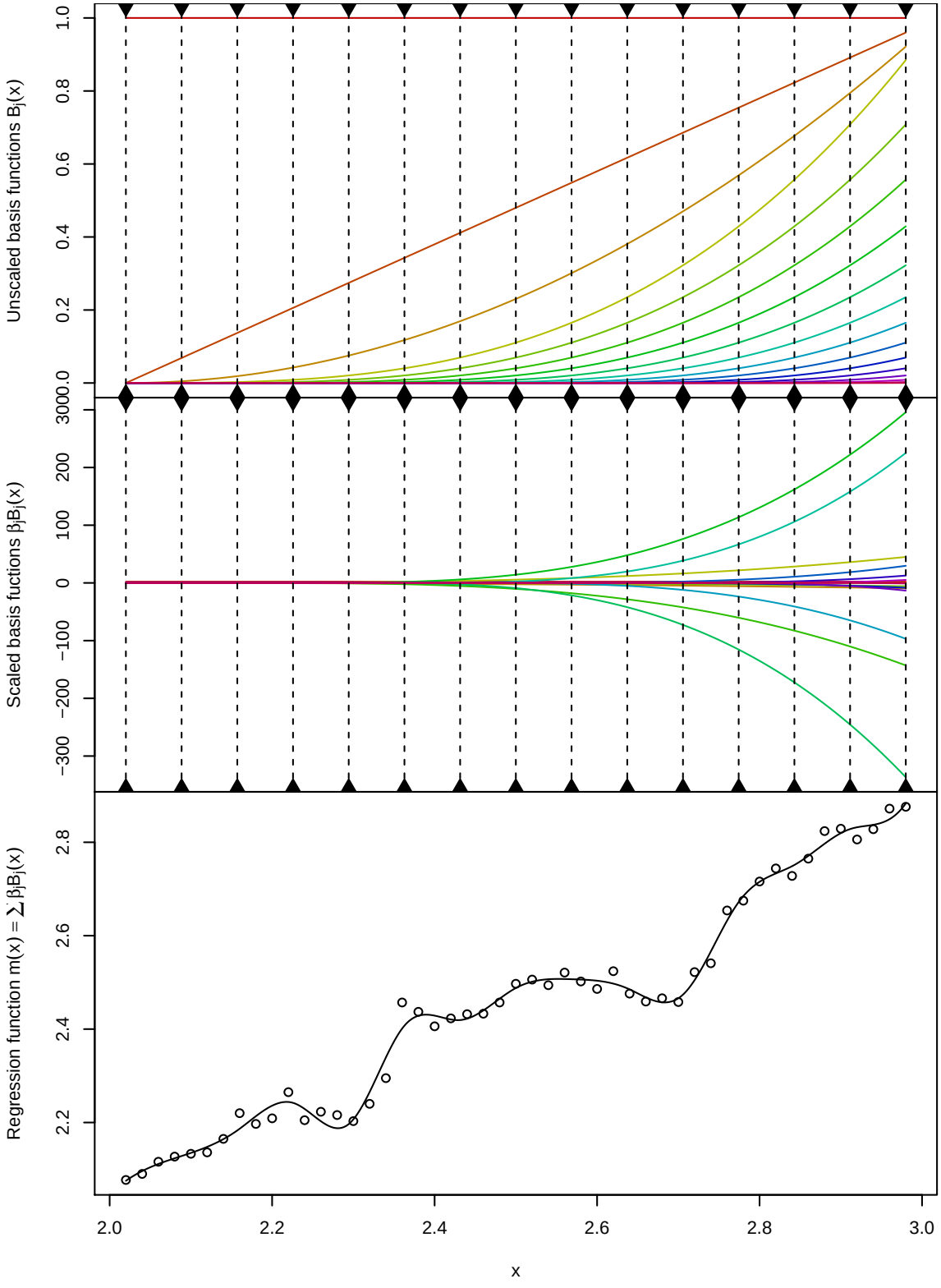


Figure 3.10. Illustration of flexible regression using the truncated power series basis of degree 3 applied to the radiocarbon data. The top panel shows the unscaled basis functions $B_j(x)$. The middle panel shows the scaled basis functions $\hat{\beta}_j B_j(x)$. The bottom panel shows a scatter plot of the data together with the fitted function $\hat{m}(x) = \sum_j \hat{\beta}_j B_j(x)$.

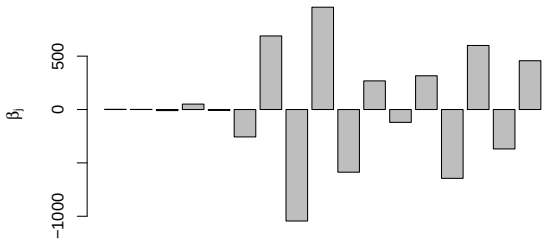


Figure 3.11. Bar plot of the coefficients $\hat{\boldsymbol{\beta}}$ estimated using the truncated power series regression model shown in figure 3.10.

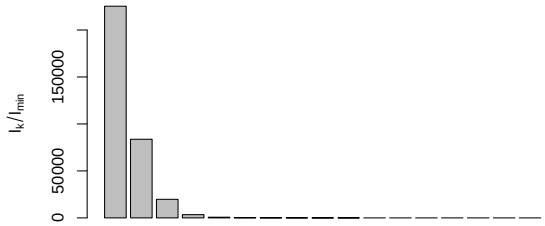
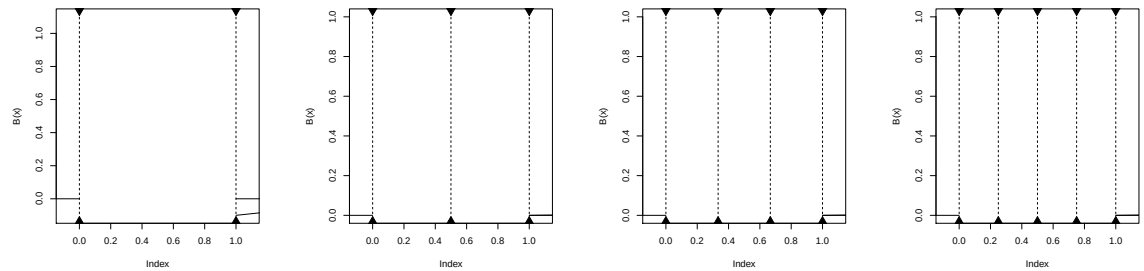


Figure 3.12. Scree plot of the singular values of the design matrix $\mathbf{B}$ (square root of the eigenvalues of the cross-product matrix $\mathbf{B}'\mathbf{B}$) for the truncated power series regression model shown in figure 3.10.



(a) One basis function of degree $r = 0$ (b) One basis function of degree $r = 1$ (c) One basis function of degree $r = 2$ (d) One basis function of degree $r = 3$

Figure 3.13. One basis function of a B-spline basis with degree $r \in \{0, 1, 2, 3\}$ using $r + 1$ knots.

$$B_j^0(x) = \begin{cases} 1 & \text{for } \kappa_j \leq x < \kappa_{j+1} \\ 0 & \text{otherwise.} \end{cases}$$

(b) Given a set of l knots the B-spline basis of degree $r > 0$ is given by the functions $(B_1^r(x), \ldots, B_{l+r-1}^r)$ with

$$B_j^r(x) = \frac{x - \kappa_{j-r}}{\kappa_j - \kappa_{j-r}} B_{j-1}^{r-1}(x) + \frac{\kappa_{j+1} - x}{\kappa_{j+1} - \kappa_{j+1-r}} B_j^{r-1}(x).$$

In order to be able to construct the splines recursively we have to introduce additional outside knots to the left of κ_1 and to the right of κ_l. In order to be able to construct a basis of degree r we need r additional outside knots on each side. Figure 3.15 illustrates this idea. These outside knots are just used to construct the basis.

From their recursive definition one can derive that B-splines have the following properties. These can also be seen in figure 3.15.

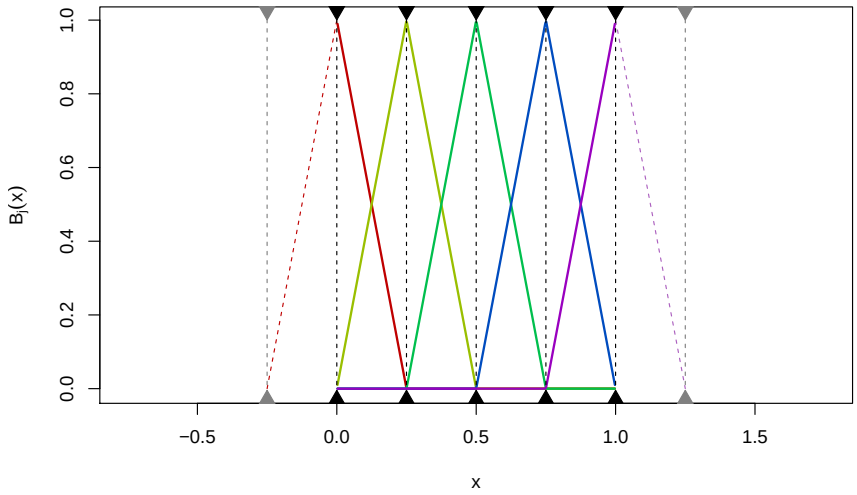
- A B-spline basis function of degree r is made up of $r + 1$ polynomials of degree r. Outside these $r+1$ intervals, the basis function is zero. This makes the basis functions local.
- At every $x \in (a, b)$ only $r + 1$ basis functions are non-zero.
- The basis functions sum to 1 for all $x \in [a, b]$. This implies that we do not need to include an intercept in the design matrix.
- One can show (homework exercise) that the derivative of a B-spline of degree r is a B-spline of degree $r - 1$.

We can fit a B-spline model to data by using the design matrix

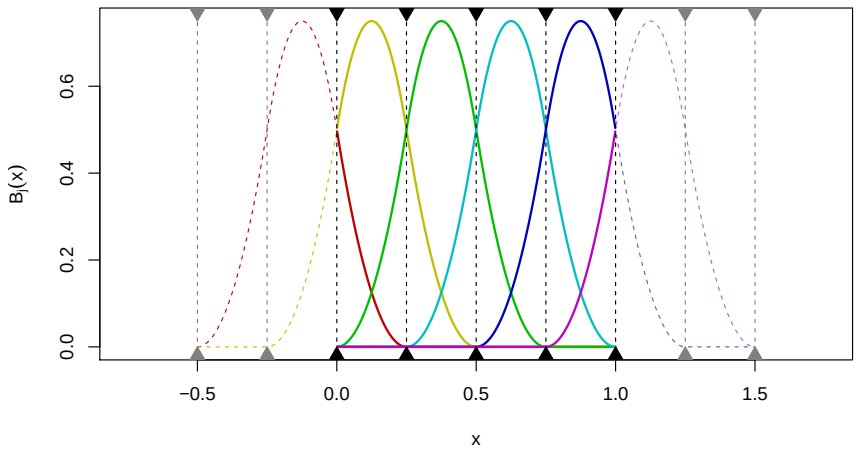
$$\mathbf{B} = \begin{pmatrix} B_1^r(x_1) & \ldots & B_{l+r-1}^r(x_1) \\ \vdots & \ddots & \vdots \\ B_1^r(x_n) & \ldots & B_{l+r-1}^r(x_n) \end{pmatrix}.$$

Example 3.7 (Radiocarbon data (continued)). Figure 3.15 illustrates the use of a B-spline basis for fitting a spline-based flexible regression model for the radiocarbon data.

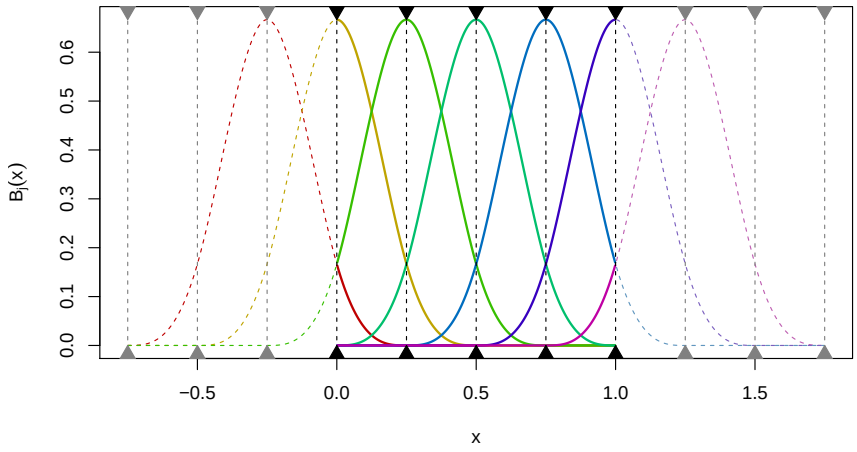
The B-spline basis is numerically much better behaved. The coefficient values (cf. figure 3.16) are not too large and the columns of the design matrix $\mathbf{B}$ are much less correlated than the columns of the truncated power basis; the maximum correlation is 0.8309. The condition number of $\mathbf{B}$ is 25.664 (cf. figure 3.17) and the condition number of $\mathbf{B}^\top\mathbf{B}$ is 358.263. ◁



(a) Degree $r = 1$



(b) Degree $r = 2$



(c) Degree $r = 3$

Figure 3.14. B spline bases for degrees $r \in \{1, 2, 3\}$.

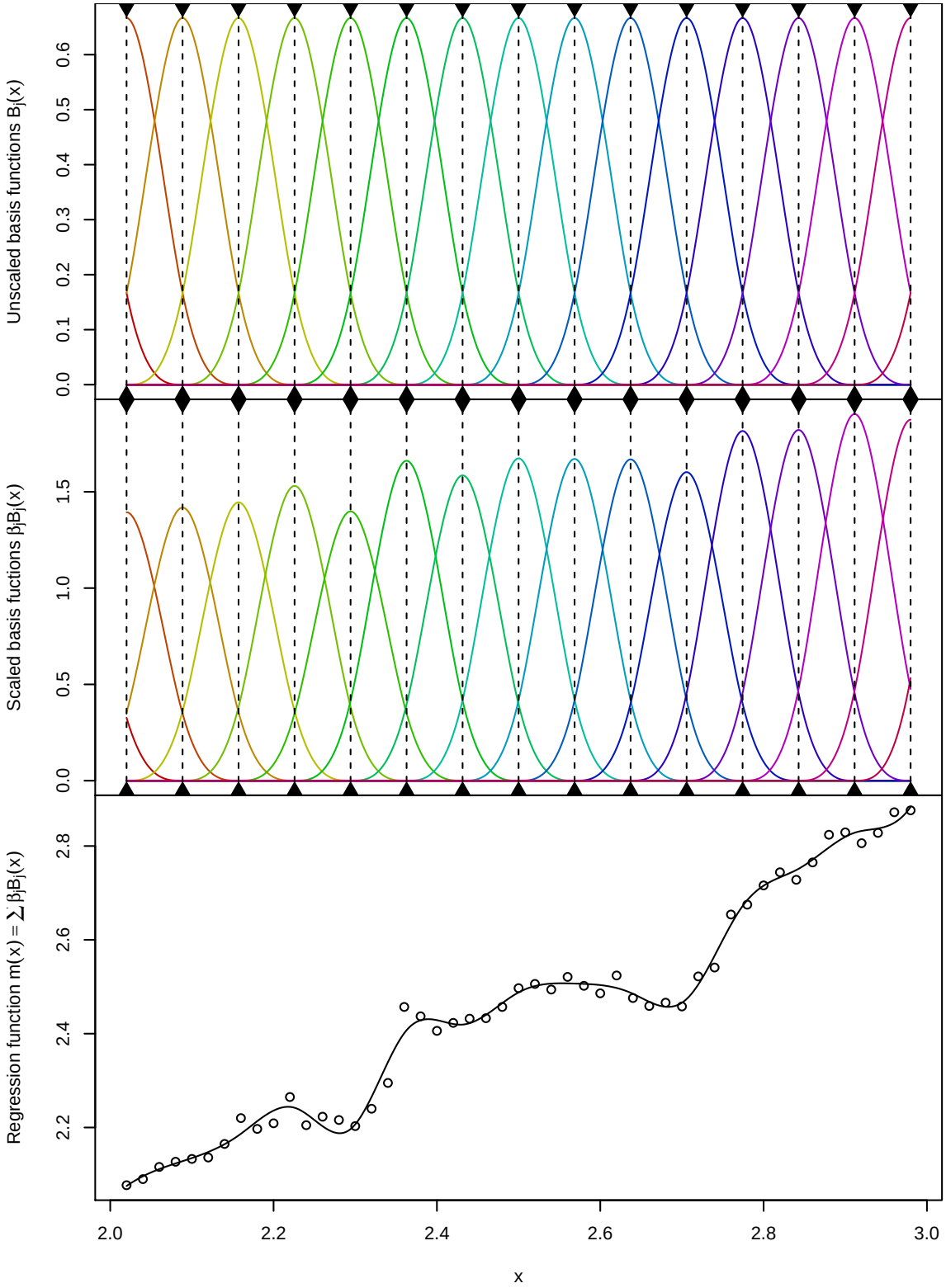


Figure 3.15. Illustration of flexible regression using the B-spline basis applied to the radiocarbon data. The top panel shows the unscaled basis functions $B_j(x)$. The middle panel shows the scaled basis functions $\hat{\beta}_j B_j(x)$. The bottom panel shows a scatter plot of the data together with the fitted function $\hat{f}(x) = \sum_j \hat{\beta}_j B_j(x)$.

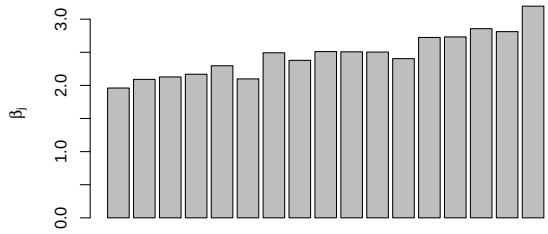


Figure 3.16. Bar plot of the coefficients $\hat{\boldsymbol{\beta}}$ estimated using the B-spline regression model shown in figure 3.15.

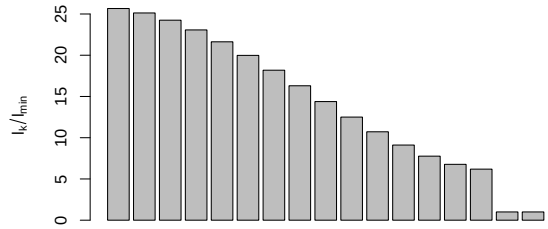


Figure 3.17. Scree plot of the singular values of the design matrix $\mathbf{B}$ (square root of the eigenvalues of the cross-product matrix $\mathbf{B}'\mathbf{B}$) for the B-spline regression model shown in figure 3.15. The condition number of $\mathbf{B}'\mathbf{B}$ is 395.661.

3.3 Penalised splines (P-splines)

A reminder of ridge regression

Ridge regression solves the penalised (or regularised) least-squares criterion

$$\|\mathbf{y} - \mathbf{B}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|^2,$$

where $\mathbf{B}$ is the matrix of covariates. The solution of this problem is given by

$$\hat{\boldsymbol{\beta}}_{\text{ridge}} = (\mathbf{B}^\top\mathbf{B} + \lambda\mathbf{I}_p)^{-1}\mathbf{B}^\top\mathbf{y}$$

To compute $\hat{\boldsymbol{\beta}}_{\text{ridge}}$ it is numerically more stable to use a QR decomposition to minimise the augmented system

$$\left\| \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix} - \begin{pmatrix} \mathbf{B} \\ \sqrt{\lambda}\mathbf{I} \end{pmatrix} \boldsymbol{\beta} \right\|^2$$

When using splines the positioning of the knots can have a large influence on the fitted function, especially if a comparatively small number of basis functions is used. One way of avoiding this problem is to use *penalised splines*. They are based on the idea of *not* using the number of basis functions to control the smoothness of the estimate, but

to use a roughness penalty to this end. This is similar in spirit to the approach discussed in section 3.2.3, though in most cases it is not necessary to use one basis function per observation. Around 20 to 30 basis functions should be sufficient. Without including a penalty in the fitting criterion this would most likely lead to an overfit to the data. Thus we need to consider a penalised criterion which, just like in section 3.2.3, contains a roughness penalty. In this section we will use $\|\mathbf{D}\boldsymbol{\beta}\|^2$ as roughness penalty, i.e. we choose the regression coefficients $\boldsymbol{\beta}$ by minimising

$$\sum_{i=1}^{n}(y_i - m(x_i))^2 + \lambda\|\mathbf{D}\boldsymbol{\beta}\|^2. \tag{3.3}$$

This objective function is, with the exception of the inclusion of the matrix $\mathbf{D}$, the objective function of ridge regression. As before, λ controls the trade-off between following the data (small λ) and obtaining a strongly regularised curve (large λ). In analogy with ridge regression one can show that the optimal $\boldsymbol{\beta}$ is given by

$$\boldsymbol{\beta} = (\mathbf{B}^\top\mathbf{B} + \lambda\mathbf{D}^\top\mathbf{D})^{-1}\mathbf{B}^\top\mathbf{y},$$

where $\mathbf{B}$ is the design matrix corresponding to the B-spline basis used for $m(\cdot)$. Numerically, it is more advantageous to represent the penalty term $\lambda\|\mathbf{D}\boldsymbol{\beta}\|^2$ by including it into an expanded design matrix, i.e. to solve

$$\left\| \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix} - \begin{pmatrix} \mathbf{B} \\ \sqrt{\lambda}\mathbf{D} \end{pmatrix} \boldsymbol{\beta} \right\|^2$$

using a QR decomposition.

There are (at least) two possible approaches for choosing $\mathbf{D}$. We can choose $\mathbf{D}$ to be a difference matrix, or we can choose $\mathbf{D}$ such that $\|\mathbf{D}\boldsymbol{\beta}\|^2 = \int_a^b m''(x)^2\,dx$. The former is both conceptually and computationally simpler; the latter is closer to what the theory suggests as optimal.

3.3.1 Difference penalties

The simplest choice of $\mathbf{D}$ is to use a difference penalty. Using the identity matrix for $\mathbf{D}$, as we would in ridge regression, is usually not appropriate: it shrinks all coefficients to zero, i.e. it shrinks the regression function $m(\cdot)$ to zero as well, which rarely desirable (cf. figure 3.18(a)). As we can see from the middle panel of figure 3.15, we obtain a smooth function when neighbouring β_j's are similar.

This can be achieved by using one of the following choices. We assume that we are using equally-spaced knots.

First-order differences. We can set

$$\mathbf{D}_1 = \begin{pmatrix} 1 & -1 & \dots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 1 & -1 \end{pmatrix}.$$

This calculates the roughness penalty as the sum of the squared first-order differences between the neighbouring β_j, i.e.

$$\|\mathbf{D}_1\boldsymbol{\beta}\|^2 = \sum_{j=1}^{l+r-2} (\beta_{j+1} - \beta_j)^2$$

This penalty shrinks the coefficients towards a common constant (cf. figure 3.18(b)) and thus shrinks the regression function $m(\cdot)$ towards a constant function. Adding a constant to $m(\cdot)$ does thus not change the penalty.

This penalty is the natural choice if B-splines of order 2 are used.

Second-order differences. We can set

$$\mathbf{D}_2 = \begin{pmatrix} 1 & -2 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 1 & -2 & 1 \end{pmatrix}.$$

This calculates the roughness penalty as the sum of the squared second-order differences between the neighbouring β_j, i.e.

$$\|\mathbf{D}_2\boldsymbol{\beta}\|^2 = \sum_{j=1}^{l+r-3} (\beta_{j+2} - 2\beta_{j+1} + \beta_j)^2$$

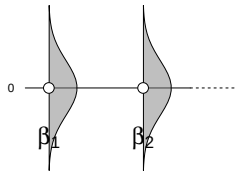
This penalty shrinks the coefficients towards a linear sequence (cf. figure 3.18(c)) and thus shrinks the regression function $m(\cdot)$ towards a linear function. Adding a linear function to $m(\cdot)$ does thus not change the penalty.

This penalty is the natural choice if B-splines of order 3 are used.

Higher-order differences. Higher-order difference matrices can be constructed using the recursive formula $\mathbf{D}_r = \mathbf{D}_1\mathbf{D}_{r-1}$ where $\mathbf{D}_r$ denotes the penalty matrix of order r.

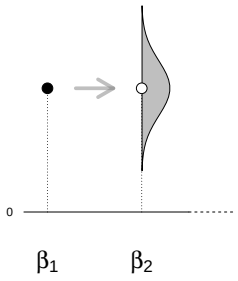
Example 3.8 (Radiocarbon dating (continued)). Figure 3.19 shows the model fit obtained when fitting a P-spline model with different values of the smoothing parameter λ. The smaller λ the closer the fitted function $\hat{m}(\cdot)$ is to the data, which leads for very small values of λ to an overfit to the data. ◁

Penalty interpretation: Only an all-zero coefficient vector incurs no penalty.
Bayesian interpretation: Independent zero-mean Gaussian prior.



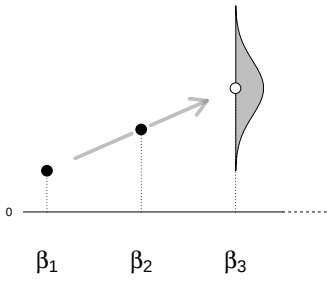
(a) Illustration of a 0-th order penalty (ridge regression).

Penalty interpretation: Only an all constant coefficient vector incurs no penalty.
Bayesian interpretation: Conditional distribution of β_2 given β_1 is Gaussian with mean β_1.
(First-order random walk)



(b) Illustration of a first-order penalty.

Penalty interpretation: Only a coefficient vector which forms a linear sequence incurs no penalty.
Bayesian interpretation: Conditional distribution of β_3 given β_1 and β_2 is Gaussian with mean $2 \cdot \beta_2 - \beta_1$.
(Second-order random walk)



(c) Illustration of a second-order penalty.

Figure 3.18. Illustration of difference penalties of order 0 to 2.

3.3.2 Other penalties

Difference penalties are not the only choice of penalty matrix. An alternative choice consists of choosing $\mathbf{D}$ such that $\|\mathbf{D}\boldsymbol{\beta}\|^2 = \int_a^b m''(x)^2\,dx$, which is the roughness penalty we have used in section 3.2.3.

Using that $m''(x) = \sum_{j=1}^{l+r-1} \beta_j B_j''(x)$ we have that

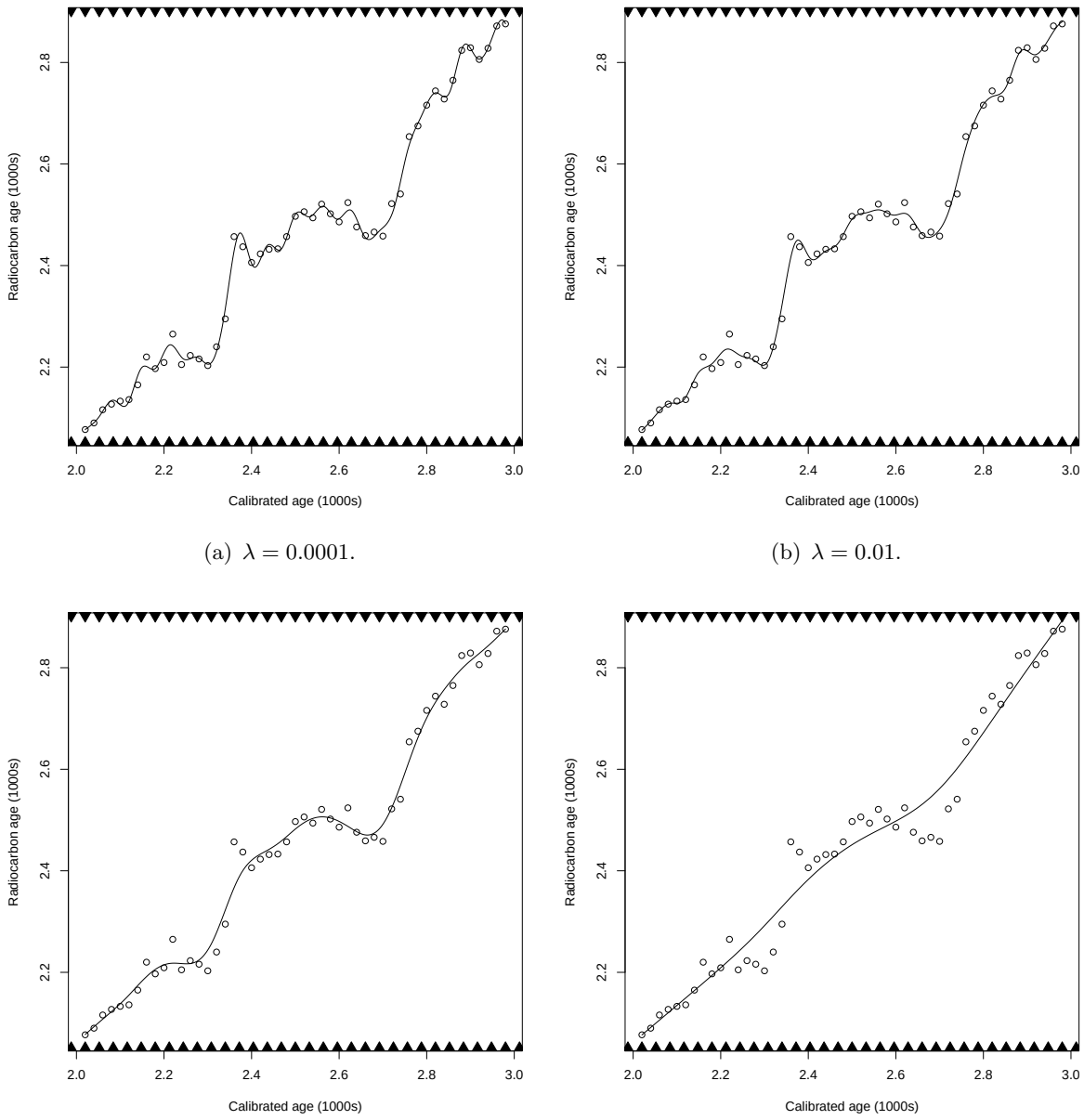
$$\begin{aligned}\int_a^b m''(x)^2\,dx &= \sum_{j=1}^{l+r-1}\sum_{k=1}^{l+r-1} \beta_j\beta_k \int_a^b B_j''(x)B_k''(x)\,dx \\ &= \boldsymbol{\beta}^\top \begin{pmatrix} \int_a^b B_1''(x)B_1''(x)\,dx & \dots & \int_a^b B_1''(x)B_{l+r-1}''(x)\,dx \\ \vdots & \ddots & \vdots \\ \int_a^b B_1''(x)B_{l+r-1}''(x)\,dx & \dots & \int_a^b B_{l+r-1}''(x)B_{l+r-1}''(x)\,dx \end{pmatrix} \boldsymbol{\beta}\end{aligned}$$

Thus we just need to choose $\mathbf{D}$ such that

$$\mathbf{D}^\top\mathbf{D} = \begin{pmatrix} \int_a^b B_1''(x)B_1''(x)\,dx & \dots & \int_a^b B_1''(x)B_{l+r-1}''(x)\,dx \\ \vdots & \ddots & \vdots \\ \int_a^b B_1''(x)B_{l+r-1}''(x)\,dx & \dots & \int_a^b B_{l+r-1}''(x)B_{l+r-1}''(x)\,dx \end{pmatrix}.$$

3.3.3 Effective degrees of freedom

Finally we introduce the notion of effective degrees of freedom, also sometimes called the effective number of parameters. In an un-penalised regression problem, the number of parameters provides us with information about the complexity of the model. More



(a) $\lambda = 0.0001$.

(b) $\lambda = 0.01$.

(c) $\lambda = 1$.

(d) $\lambda = 10$.

Figure 3.19. P-spline with different values of the smoothing parameter λ fitted to the radiocarbon data.

complex models have more parameters than simpler models. For penalised regression problems counting the parameters is however not meaningful. Due to the roughness penalty not all parameters are "free". Recall that in linear regression the hat matrix $\mathbf{S} = \mathbf{B}(\mathbf{B}^\top\mathbf{B})^{-1}\mathbf{B}^\top$ is a projection matrix and thus the trace $\mathrm{tr}(\mathbf{S})$ equals the number of parameters. We can generalise this to penalised models and define the effective degrees of freedom as

$$\mathrm{edf}(\lambda) = \mathrm{tr}(\mathbf{S}_\lambda),$$

where $\mathbf{S}_\lambda = \mathbf{B}(\mathbf{B}^\top\mathbf{B} + \lambda\mathbf{D}^\top\mathbf{D})^{-1}\mathbf{B}^\top$.

3.3.4 Random effects interpretation

Random effect models – Likelihood

In the random effects model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\alpha} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\varepsilon}$$

with error term $\boldsymbol{\varepsilon} \sim \mathsf{N}(\mathbf{0}, \sigma^2\mathbf{I})$ and random effect $\boldsymbol{\gamma} \sim \mathsf{N}(\mathbf{0}, \tau^2\mathbf{I})$ twice the loglikelihood is (ignoring the variance parameters) given by

$$-\frac{1}{\sigma^2}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^\top\boldsymbol{\alpha} - \mathbf{z}_i^\top\boldsymbol{\gamma})^2 - \frac{1}{\tau^2}\sum_{j=1}^{q}\gamma_j^2$$

Comparing the penalised least squares criterion (3.3) to the loglikelihood suggests that we can interpret the penalised regression model as a random effects model with no fixed effect and random effect $\boldsymbol{\beta}$. However the problem is that, at least for difference matrices, $\mathbf{D}^\top\mathbf{D}$ is not of full rank, thus we cannot take its inverse matrix square root. In order to obtain a proper random-effects representation we need to "split" $\boldsymbol{\beta}$ into an (unpenalised) fixed effect and a (penalised) random effect.

In the following we will only consider the case of a difference penalty of order 1 or 2. In the case of a first-order difference penalty we define $\mathbf{G} = (1, \ldots, 1)$. For a second-order difference penalty we define $\mathbf{G} = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ 1 & 2 & \ldots & l+r-1 \end{pmatrix}$. The rows in $\mathbf{G}$ are parameter sequences which do not incur a penalty, i.e. $\mathbf{GD} = \mathbf{0}$. We also define $\mathbf{H} = \mathbf{D}^\top(\mathbf{D}\mathbf{D}^\top)^{-1}$. We can now write

$$\boldsymbol{\beta} = \mathbf{G}\boldsymbol{\alpha} + \mathbf{H}\boldsymbol{\gamma}$$

Because $\mathbf{D}\mathbf{D}^\top$ is of full rank we have that $\mathbf{DH} = \mathbf{D}\mathbf{D}^\top(\mathbf{D}\mathbf{D}^\top)^{-1} = \mathbf{I}$. Plugging this into the objective function (3.3) gives

$$\|\mathbf{y} - \mathbf{BG}\boldsymbol{\alpha} - \mathbf{BH}\boldsymbol{\gamma}\|^2 + \lambda\Big(\underbrace{\boldsymbol{\alpha}\mathbf{G}^\top\mathbf{D}^\top\mathbf{DG}\boldsymbol{\alpha}}_{=0} + 2\underbrace{\boldsymbol{\alpha}\mathbf{G}^\top\mathbf{D}^\top\mathbf{DH}\boldsymbol{\gamma}}_{=0} + \boldsymbol{\gamma}^\top\underbrace{\mathbf{H}^\top\mathbf{D}^\top\mathbf{DH}}_{=\mathbf{I}}\boldsymbol{\gamma}\Big)$$
$$= \|\mathbf{y} - \mathbf{BG}\boldsymbol{\alpha} - \mathbf{BH}\boldsymbol{\gamma}\|^2 + \lambda\|\boldsymbol{\gamma}\|^2$$

Defining $\mathbf{X} = \mathbf{BG}$ and $\mathbf{Z} = \mathbf{BH}$ and denoting rows of $\mathbf{X}$ and $\mathbf{Z}$ by $\mathbf{x}_i$ and $\mathbf{z}_i$ respectively, this is equivalent to

$$\sum_{i=1}^{n}(y_i - \mathbf{x}_i^\top\boldsymbol{\alpha} - \mathbf{z}_i^\top\boldsymbol{\gamma})^2 + \lambda\sum_{j=1}^{q}\gamma_j^2,$$

which is $-\sigma^2$ times the loglikelihood of a random-effects model, which we have stated above. Hereby we have used $\lambda = \sigma^2/\tau^2$.

Thus the penalised regression model is nothing other than a random effects effect and we can use standard mixed model software to fit these models. Most importantly we can estimate the variances σ^2 and τ^2 is a mixed model (using (restricted) maximum likelihood, which gives us a way of estimating the otherwise rather elusive smoothing parameter $\hat{\lambda} = \hat{\sigma}^2/\hat{\tau}^2$.

3.3.5 Bayesian interpretation

Rather than interpreting the penalised fitting criterion as a random effects model we can treat the penalised regression model as a fully Bayesian model with the following prior and data model.

$$\mathbf{D}\boldsymbol{\beta}|\tau^2 \sim \mathsf{N}(\mathbf{0}, \tau^2\mathbf{I})$$
$$\mathbf{y}|\boldsymbol{\beta}, \sigma^2 \sim \mathsf{N}(\mathbf{B}\boldsymbol{\beta}, \sigma^2\mathbf{I})$$

The prior distribution of $\boldsymbol{\beta}$ is improper if $\mathbf{D}$ is not of full rank, which is the case for all difference penalties. However in the case of difference penalties the prior distribution of $\boldsymbol{\beta}$ can be expressed in terms of random walks (cf. figure 3.18).

First-order random walk The first-order penalty corresponds to an improper flat prior on β_1 and $\beta_j|\beta_{j-1} \sim \mathsf{N}(\beta_{j-1}|\tau^2)$ (for $j \geq 2$).

Second-order random walk The second-order penalty corresponds to an improper flat prior on β_1 and β_2 and $\beta_j|\beta_{j-1}, \beta_{j-2} \sim \mathsf{N}(2\beta_{j-1} - \beta_{j-2}|\tau^2)$ (for $j \geq 3$).

It seems natural to complement the model with priors for σ^2 and τ^2

$$\sigma^2 \sim \mathsf{IG}(a_{\sigma^2}, b_{\sigma^2})$$

$$\tau^2 \sim \mathsf{IG}(a_{\tau^2}, b_{\tau^2})$$

Inference can then be carried out efficiently using a Gibbs sampler. This model and many other Bayesian smoothing models are implemented in the software `BayesX`.

Rather than placing a independent inverse-gamma prior on τ^2 we can set $\tau^2 = \sigma^2/\lambda$ and place a prior of our choice on λ. In this model the posterior distribution distribution of λ does not follow a known distribution, but can be evaluated efficiently, as all the other parameters can be integrated out in closed form. The drawback is that the integration over λ would need to be carried out numerically, which suggests that this approach is better suited for an empirical Bayes strategy for estimating λ.

3.4 Splines in more than one dimension

3.4.1 Tensor-product splines

So far we have only covered the construction of spline bases in one dimension. In this section we will see how we can turn a one-dimensional spline basis into a spline basis of any dimension. To keep things simple we shall start with the bivariate case.

Suppose we have two covariates and want to fit a regression model of the form

$$\mathbb{E}(Y_i) = m(x_{i1}, x_{i2}),$$

where $m(\cdot, \cdot)$ is a bivariate surface.

We start by placing a basis on each dimension separately. Denote by $B_1^{(1)}(x_1), \ldots, B_{l_1+r-1}^{(1)}(x)$ the basis functions placed on the first covariate and by $B_1^{(2)}(x_1), \ldots, B_{l_2+r-1}^{(2)}(x)$ the basis functions placed on the second covariate. We now define a set of basis functions

$$B_{jk}(x_1, x_2) = B_j^{(1)}(x_1) \cdot B_k^{(2)}(x_2)$$

for $j \in 1, \ldots, l_1 + r - 1$ and $k \in 1, \ldots, l_2 + r - 1$. Figure 3.20 shows how one such bivariate basis function looks like for different degrees of the underlying univariate B-spline. Figure 3.21 shows all 36 bivariate basis functions resulting from two B-spline bases with six basis functions each.

We will now use the basis expansion

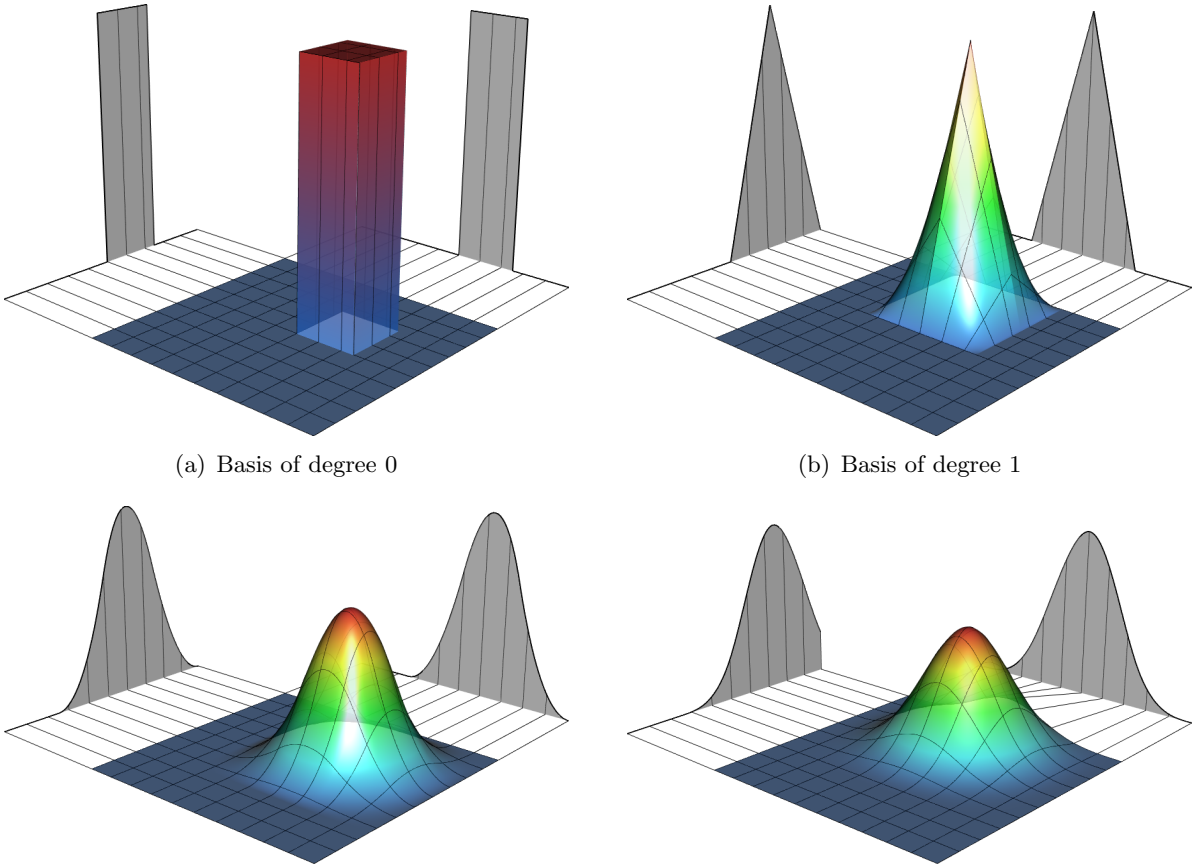
$$m(x_{i1}, x_{i2}) = \sum_{j=1}^{l_1+r-1} \beta_{jk} B_{jk}(x_1, x_2)$$

which corresponds to the design matrix

$$\mathbf{B} = \begin{pmatrix} B_{11}(x_{11}, x_{12}) & \cdots & B_{l_1+r-1,1}(x_{11}, x_{12}) & B_{12}(x_{11}, x_{12}) & \cdots & B_{l_1+r-1,2}(x_{11}, x_{12}) & \cdots & B_{l_1+r-1,l+2+r-1}(x_{11}, x_{12}) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ B_{11}(x_{n1}, x_{n2}) & \cdots & B_{l_1+r-1,1}(x_{n1}, x_{n1}) & B_{12}(x_{n1}, x_{n2}) & \cdots & B_{l_1+r-1,2}(x_{n1}, x_{n2}) & \cdots & B_{l_1+r-1,l+2+r-1}(x_{n1}, x_{n2}) \end{pmatrix}$$

and coefficient vector $\boldsymbol{\beta} = (\beta_{11}, \ldots, \beta_{l_1+r-1,1}, \beta_{12}, \ldots, \beta_{l_2+r-1,2}, \ldots, \beta_{l_1+r-1,l_2+r-1})^\top$.

We can generalise this principle of constructing a basis to dimension p by multiplying all combinations of basis functions of the p covariates.



(a) Basis of degree 0

(b) Basis of degree 1

(c) Basis of degree 2

(d) Basis of degree 3

Figure 3.20. Illustration of the construction of a single bivariate B-spline basis function $B_{jk}(x_1, x_2) = B_j(x_1) \cdot B_k(x_2)$ for B-spline bases of different degree.

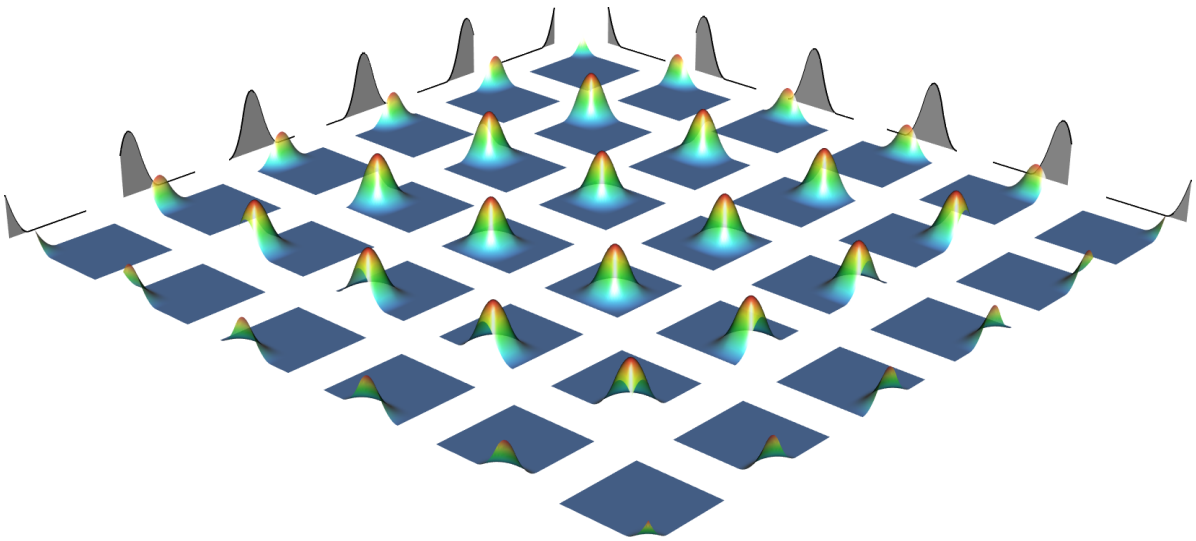


Figure 3.21. Illustration of the construction of a bivariate B-spline basis created from a univariate B-spline basis.

Finally, we need to explain how a penalty matrix can be constructed for this bivariate spline basis. We will explain the basic idea using figure 3.21. A simple way of constructing a roughness penalty consist of applying the univariate roughness penalties to the rows and columns of the basis functions. More mathematically, this corresponds to taking Kronecker products, i.e. using the difference matrix

$$\mathbf{D} = \begin{pmatrix} \mathbf{D}^{(2)} \otimes \mathbf{I}_{l_1+r-1} \\ \mathbf{I}_{l_2+r-1} \otimes \mathbf{D}^{(1)} \end{pmatrix},$$

where $\mathbf{D}^{(1)}$ is the univariate difference matrix used for the first dimension and $\mathbf{D}^{(2)}$ is the univariate difference matrix used for the second dimension.

Example 3.9 (Great Barrier Reef (continued)). Figure 3.22 shows the result of fitting a tensor-product-spline model to the data from example 1.3. The objective is to model a score which represents the composition of the catch as a function of longitude and latitude. ◁

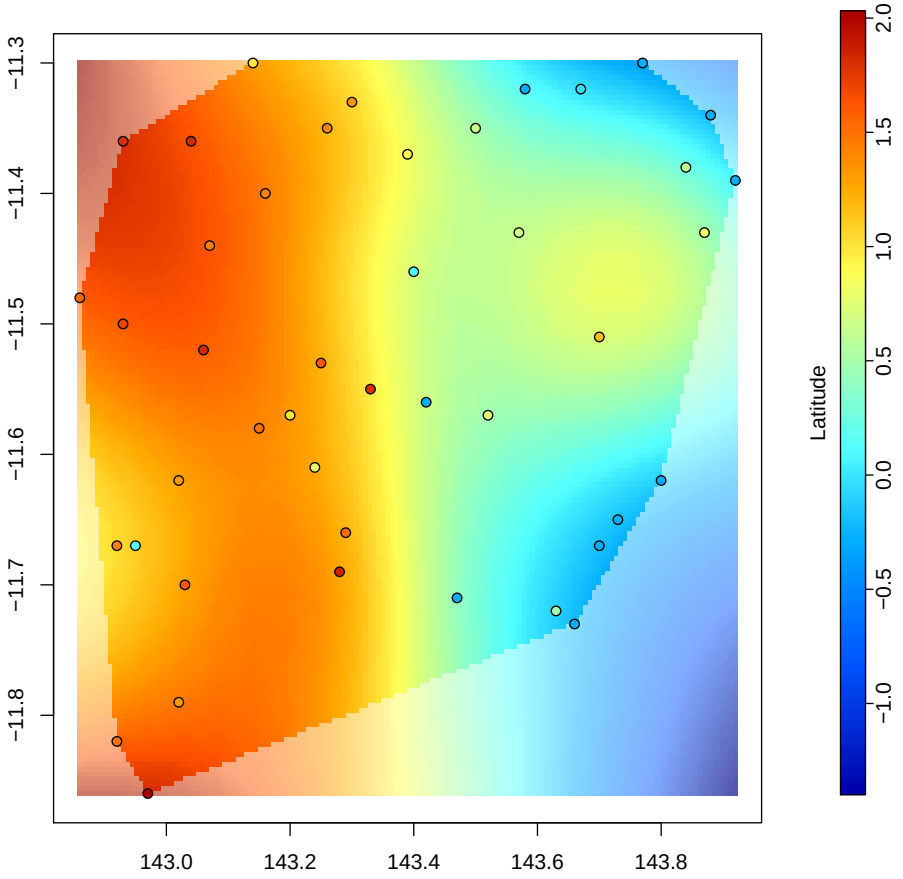


Figure 3.22. Predicted score obtained from a tensor-product-spline model fitted to the Great Barrier Reef data.

3.4.2 Thin-plate splines

In this section we generalise natural cubic splines to the bivariate case, which provides an alternative way of bivariate spline smoothing. In section 3.2.3 we have seen that the minimiser of

$$\underbrace{\sum_{i=1}^{n}(y_i - m(x_i))^2}_{\text{Fit to the data}} + \lambda \underbrace{\int_a^b m''(x)^2\, dx}_{\text{Roughness penalty}},$$

has to be a natural cubic spline.

Generalising this variational problem to the bivariate case leads to the objective function

$$\underbrace{\sum_{i=1}^{n}(y_i - m(x_{i1}, x_{i2}))^2}_{\text{Fit to the data}} + \lambda \underbrace{\int\int \left(\frac{\partial^2}{\partial x_1^2} m(x_1, x_2) + 2\frac{\partial^2}{\partial x_1 \partial x_2} m(x_1, x_2) + \frac{\partial^2}{\partial x_2^2} m(x_1, x_2)\right)^2 dx_2\, dx_1}_{\text{Roughness penalty}},$$

The roughness penalty can be interpreted as the bending energy of thin plate of metal. One can show that the solution to his problem has to be a so-called *thin-plate* spline of the form

$$m(\xi_1, \xi_2) = \beta_0 + \beta_1 \xi_1 + \beta_2 \xi_2 + \sum_{i=1}^{n} \beta_{2+i} K\left((\xi_1, \xi_2), (x_{i1}, x_{i2})\right),$$

where $K\left((\xi_1, \xi_2), (\zeta_1, \zeta_2)\right) = \frac{1}{2}\left((\zeta_1 - \xi_1)^2 + (\zeta_2 - \xi_2)^2\right) \cdot \log\left((\zeta_1 - \xi_1)^2 + (\zeta_2 - \xi_2)^2\right)$.

Similar to what we have discussed in section 3.3 we can estimate the coefficients β_j using a penalised least squares criterion. In fact, we need to minimise the objective function

$$\sum_{i=1}^{n} \left(y_i - m(x_{i1}, x_{i2})\right)^2 + \lambda \boldsymbol{\beta}' \mathbf{P} \boldsymbol{\beta}$$

subject to the constraints that $\sum_{i=1}^{n} \beta_{2+i} = \sum_{i=1}^{n} x_{i1}\beta_{2+i} = \sum_{i=1}^{n} x_{i2}\beta_{2+i} = 0$, where

$$\mathbf{P} = \begin{pmatrix} 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & K\left((x_{11}, x_{12}), (x_{11}, x_{12})\right) & \dots & K\left((x_{11}, x_{12}), (x_{n1}, x_{n2})\right) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & K\left((x_{n1}, x_{n2}), (x_{11}, x_{12})\right) & \dots & K\left((x_{n1}, x_{n2}), (x_{n1}, x_{n2})\right) \end{pmatrix}$$

Example 3.10 (Great Barrier Reef (continued)). Figure 3.23 shows the result of fitting a tensor-product-spline model to the data from example 1.3. ◁

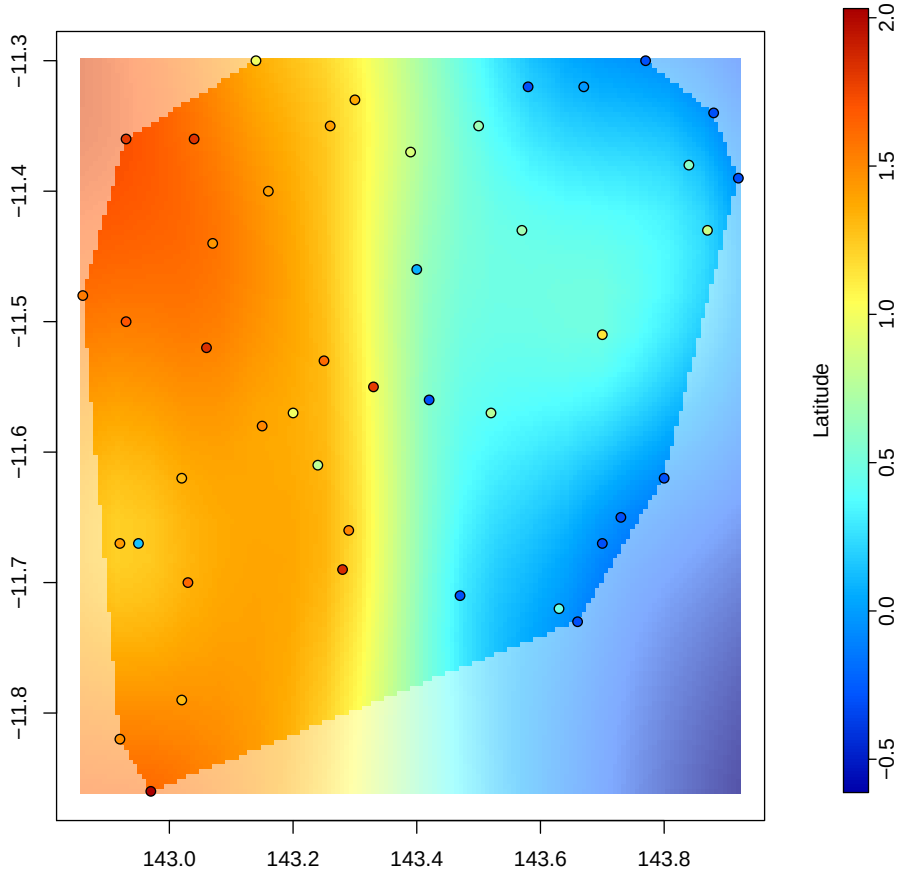
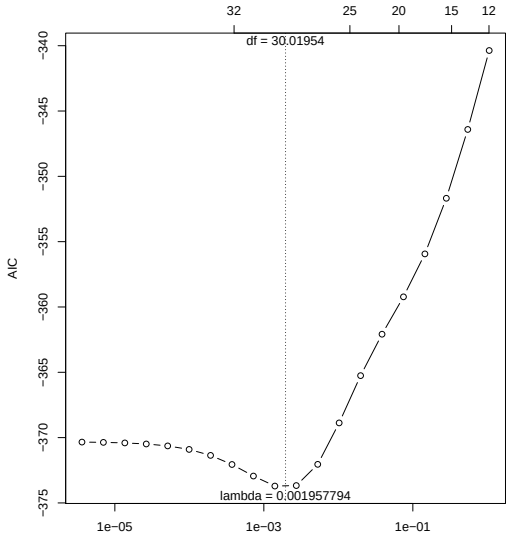


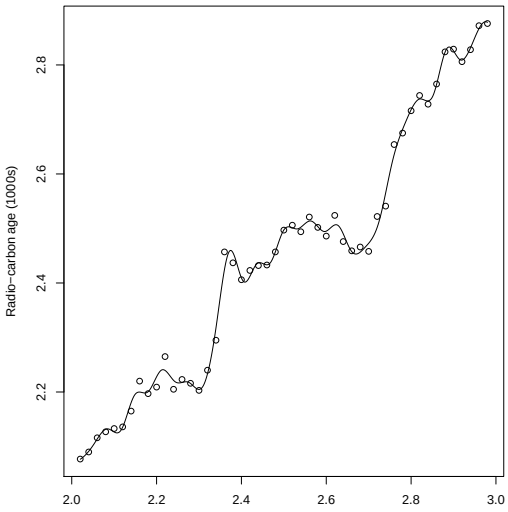
Figure 3.23. Predicted score obtained from a thin-plate-spline model fitted to the Great Barrier Reef data.

3.5 How much to smooth

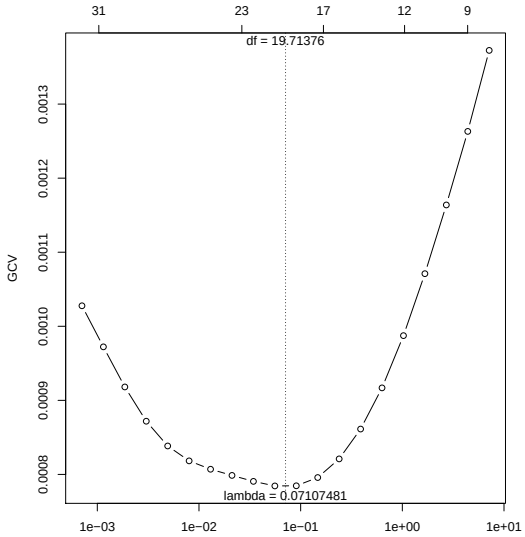
In 2.5 we have introduced criteria for selecting the optimal amount of smoothness. In this section we will compare AIC and GCV to the empirical Bayesian approach set out in this chapter. Figure 3.24 shows both the profile of the criterion as well as the resulting model fit when choosing the supposedly optimal value. The example shows that the criteria can give markedly different result and that care should be taken when blindly using criteria such as the AIC, which in our example leads to a pronounced overfit.



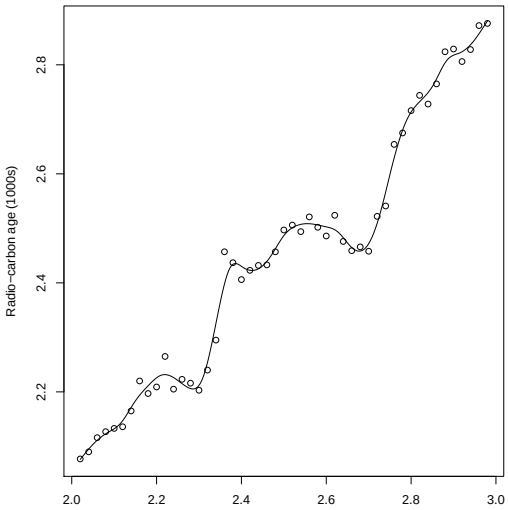
(a) Profile plot of AIC.



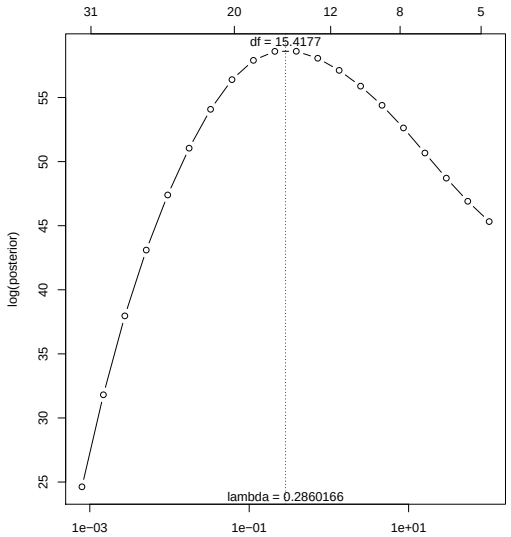
(b) Regression function obtained using λ chosen by AIC.



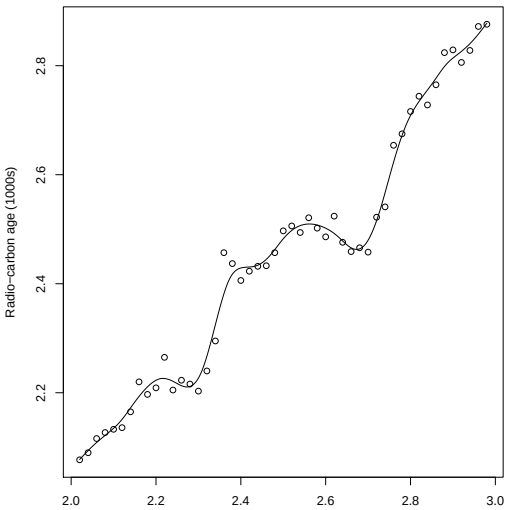
(c) Profile plot of GCV.



(d) Regression function obtained using λ chosen by GCV.



(e) Profile plot of the log-posterior.



(f) Regression function obtained using λ chosen by the empirical Bayes appraoch.

Figure 3.24. Comparison of the different criteria for choosing the smoothing parameter lambda λ.

4

More general models and inference

4.1 Reference bands

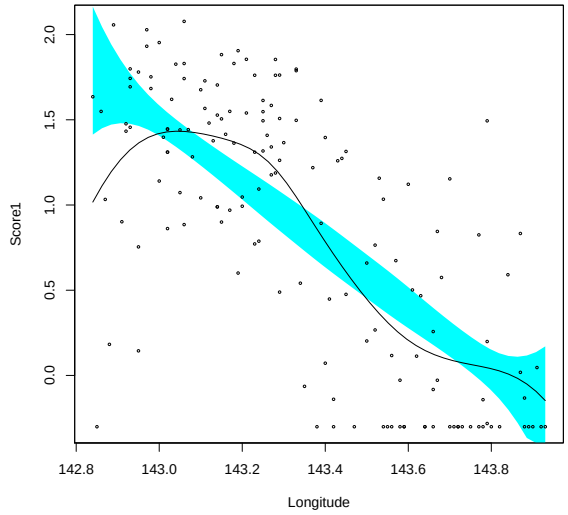
In chapter 2, standard errors of nonparametric regression curves were discussed and these can provide helpful information on the variability of the estimate. However, there are situations where we might wish to examine the suitability of particular parametric models by comparing this with a nonparametric model and here careful use of standard errors can be particularly helpful.

For example, consider the case of a single covariate x where interest lies in comparing a linear model whose mean function is $\alpha+\beta x$ with the nonparametric model whose mean function is the smooth curve $m(x)$. Under the linear model, the estimate of the regression function at a particular point x can be expressed as a simple linear function $\sum_i l_i y_i$, where the weights l_i come from the 'hat' matrix. In a similar manner, the nonparametric estimate has the form $\sum_i s_i y_i$, where the weights s_i come from the 'smoothing matrix'. Instead of using the standard errors about $\hat{m}(x)$ to judge the suitability of a linear regression, a more direct approach is to consider the standard errors of the difference between the two models. This is easily calculated as

$$\text{s.e.}\Big\{\hat{\alpha}+\hat{\beta}x-\hat{m}(x)\Big\}=\sqrt{\sum_i (l_i-s_i)^2}\,\sigma.$$

By plugging in an estimate of σ and highlighting a band, centred at the linear model, whose width is ± 2 s.e.'s, a *reference band* is created. This shows where we should expect a nonparametric estimate to lie most of the time, when the true model is linear.

The procedure is illustrated below on the Reef data, where there is clear, informal evidence that the linear model is not an adequate description of the data.



Bowman and Azzalini (1997) give further examples of reference bands and inferential methods.

4.2 Comparing models

A more global method for comparing models would be useful. Again, an analogy with linear models is helpful. There we would compute the residual sums-of-squares (RSS_0, RSS_1) and degrees of freedom (ν_0, ν_1) of each model, where model 0 is nested within model 1, and construct an F-statistic as

$$\frac{(\mathrm{RSS}_0 - \mathrm{RSS}_1)/(\mathrm{df}_1 - \mathrm{df}_0)}{\mathrm{RSS}_1/(n - \mathrm{df}_1)}.$$

This would then be compared to an $F_{\mathrm{df}_1-\mathrm{df}_0,\mathrm{n}-\mathrm{df}_1}$ distribution. In the nonparametric setting, all of the RSS's and df's are available and an approximate F-test would be carried out in the same manner.

While this is a useful benchmark, the distribution is no longer F in the nonparametric case. In some cases, a more accurate probability calculation can be performed with a little more effort. Suppose again that we have two competing models, linear and nonparametric, whose fit is expressed in the RSS's and df's. Each RSS is a quadratic form.

$$\begin{aligned} \mathrm{RSS}_0 &= y^T(I-P)^T(I-P)y = y^T(I-P)y^T \\ \mathrm{RSS}_1 &= y^T(I-S)^T(I-S)y \end{aligned}$$

Ignoring the degrees of freedom factors, the F-statistic can then be written as

$$F = \frac{y^T B y}{y^T A y},$$

where $A = (I-S)^T(I-S)$ and $B = I-P-A$. Helpfully, Johnson and Kotz (1972) give general results about quadratic forms in normal random variables, where the matrices involved are required only to be symmetric. A p-value can then be written as

$$p = \mathbb{P}\left\{ \frac{y^T B y}{y^T A y} > F_{\text{obs}} \right\} = \mathbb{P}\left\{ y^T C y > 0 \right\},$$

where $C = B - F_{\text{obs}} A$. In the case of both local linear and b-spline estimators, the estimator is unbiased under a linear regression. The problem then reduces to the calculation

$$p = \mathbb{P}\left\{ \varepsilon^T C \varepsilon > 0 \right\},$$

where ε has a multivariate normal distribution. The distribution of this type of quadratic form can be conveniently and accurately approximated by an $a\chi^2_b + c$ distribution. The cumulants of the quadratic form are given by

$$\kappa_j = 2^{j-1}(j-1)!\,\text{tr}\left\{ (VC)^j \right\},$$

where V is the covariance matrix of the multivariate normal distribution. This leads to the approximating parameters

$$a = |\kappa_3|/(4\kappa_2), \quad b = 8\kappa_2^3/\kappa_3^2, \quad c = \kappa_1 - ab.$$

When these calculations are applied to the Reef data, to assess a linear model in longitude, the p-value is effectively 0, confirming that there is clear evidence of non-linearity in the effect of longitude.

4.3 Smoothing binary data

Example 4.1 (Ascaris lumbricoides). A survey of the occurence of the human parasitic worm infection *Ascaris lumbricoides* was carried out in residents of a rural community in China. The variables are:

Age	age of the resident
Infection	presence (1) or absence (0) of infection
Sex	male (1) or female (2)

The background to the data, and an analysis, are described by Weidong et al. (1996), *Ascaris, people and pigs in a rural community of Jiangxi province, China*, Parasitology 113, 545-57. ◁

A natural extension to non-normal data is to apply weights to the log-likelihood rather than the sum-of-squares as

$$\sum_i \ell_i(\alpha, \beta)\, w(x_i - x; h),$$

where $\ell_i(\alpha, \beta)$ is the contribution to the usual log-likelihood from the ith observation. For example, in the case of logistic regression for binary responses, we have the model

$$\log\left(\frac{p_i}{1-p_i}\right) = \alpha + \beta x_i \qquad (i = 1, \ldots, n)$$

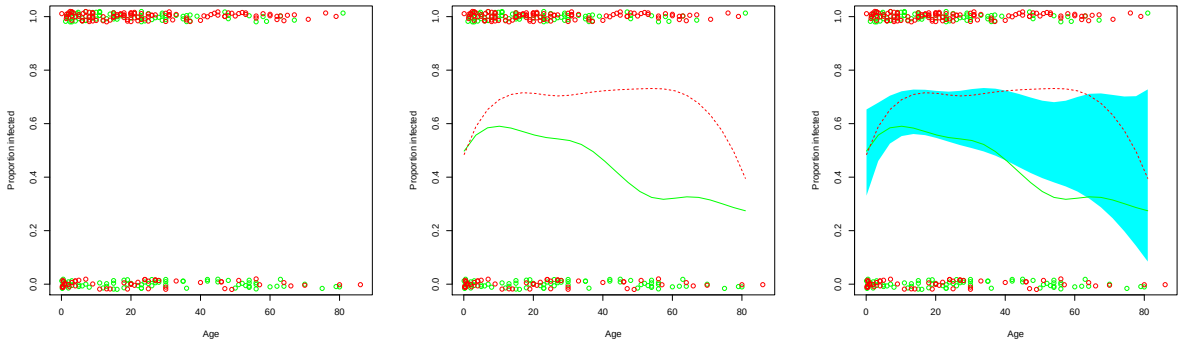
and then the likelihood contributions

$$\ell_i(\alpha, \beta) = y_i \log\left(\frac{p_i}{1-p_i}\right) + \log(1 - p_i).$$

Here p_i denotes the probability of a 1 at design point x_i, and as usual the logit link function is assumed to relate p_i to a linear predictor in x_i. Maximisation of the weighted likelihood with respect to (α, β) provides local estimates $(\hat{\alpha}, \hat{\beta})$ and by solving the model definition for p we obtain a fitted value

$$\hat{m}(x) = \frac{\exp(\hat{\alpha} + \hat{\beta}x)}{1 + \exp(\hat{\alpha} + \hat{\beta}x)}.$$

This sounds a bit complicated but it is really just fitting a glm locally by using weights to pay attention only to data points which are near the point of interest x. The procedure will be illustrated below on the worm data.



The form of smoothing described above corresponds to the 'local fitting' approach to nonparametric modelling. The penalty approach is also available. Recall from earlier APTS work that a generalised linear model involves a linear predictor plus an error distribution, usually from the exponential family, and that the model can be fitted by maximum likelihood through an iteratively reweighted least squares algorithm. In the nonparametric setting, we can simply replace the linear predictor by a linear term in spline bases and fit the model by a similar algorithm, but this time adding to the log-likelihood a penalty term based on differencing of the spline coefficients. Wood (2006) gives the details of this.

4.4 A simple additive model

Now that we have tools available to estimate smooth curves and surfaces, linear regression models can be extended to *additive models* as

$$y_i = \beta_0 + m_1(x_{1i}) + \ldots + m_p(x_{pi}) + \varepsilon_i, \qquad i = 1, \ldots, n,$$

where the m_i are functions whose shapes are unrestricted, apart from an assumption of smoothness. This gives a very flexible set of modelling tools. To see how these models can be fitted, consider the case of only two covariates,

$$y_i = \beta_0 + m_1(x_{1i}) + m_2(x_{2i}) + \varepsilon_i, \qquad i = 1, \ldots, n,$$

A rearrangement of this as $y_i - \beta_0 - m_2(x_{2i}) = m_1(x_{1i}) + \varepsilon_i$ suggests that an estimate of component m_1 can then be obtained by smoothing the residuals of the data after fitting $\hat{m}_2$,

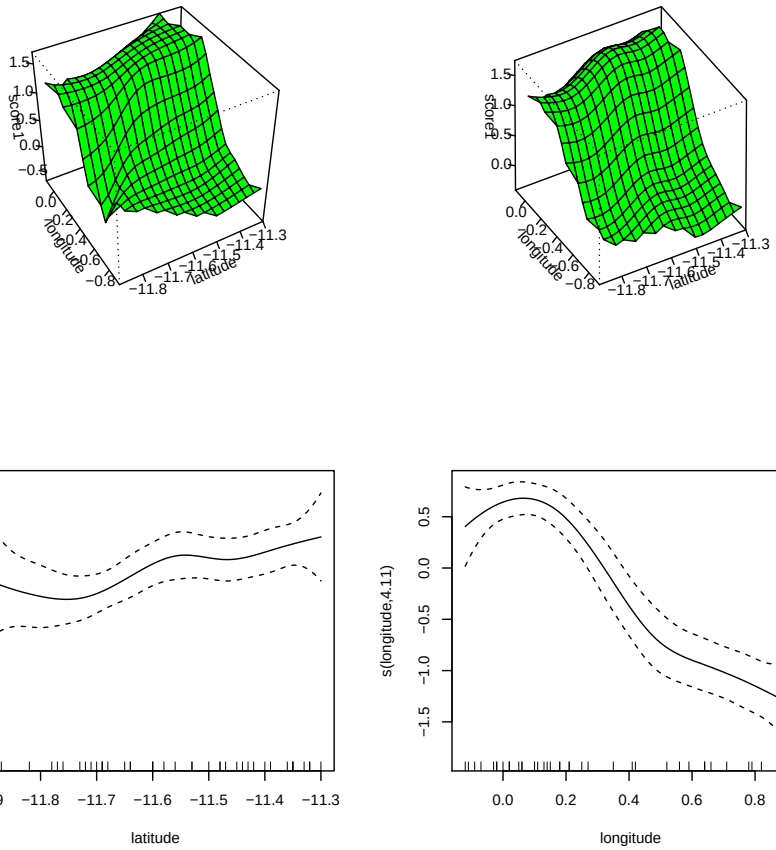
$$\hat{m}_1 = S_1(y - \bar{y} - \hat{m}_2)$$

and that, similarly, subsequent estimates of m_2 can be obtained as

$$\hat{m}_2 = S_2(y - \bar{y} - \hat{m}_1).$$

Repetition of these steps gives a simple form of the *backfitting* algorithm. The same idea applies when we have more than two components on the model. At each step we smooth over a particular variable using as response the y variable with the current estimates of the other components subtracted.

A simple example of an additive model for the Reef data was shown in Chapter 1 and this is repeated below.



4.5 More general additive models

For the more general model

$$y_i = \alpha + m_1(x_{1i}) + \ldots + m_p(x_{pi}) + \varepsilon_i.$$

a simple extension of the steps outlined for two covariates gives a form of the *backfitting* algorithm. In order to ensure identifiability, we assume that $\sum_i m_j(x_{ji}) = 0$, for each j. At each step we smooth over a particular variable using as response the y variable with the current estimates of the other components subtracted.

The backfitting algorithm can be expressed as:

$$\hat{m}_j^{(r+1)} = S_j \left(y - \hat{\alpha}\mathbf{1} - \sum_{k<j} \hat{m}_k^{(r+1)} - \sum_{k>j} \hat{m}_k^{(r)} \right).$$

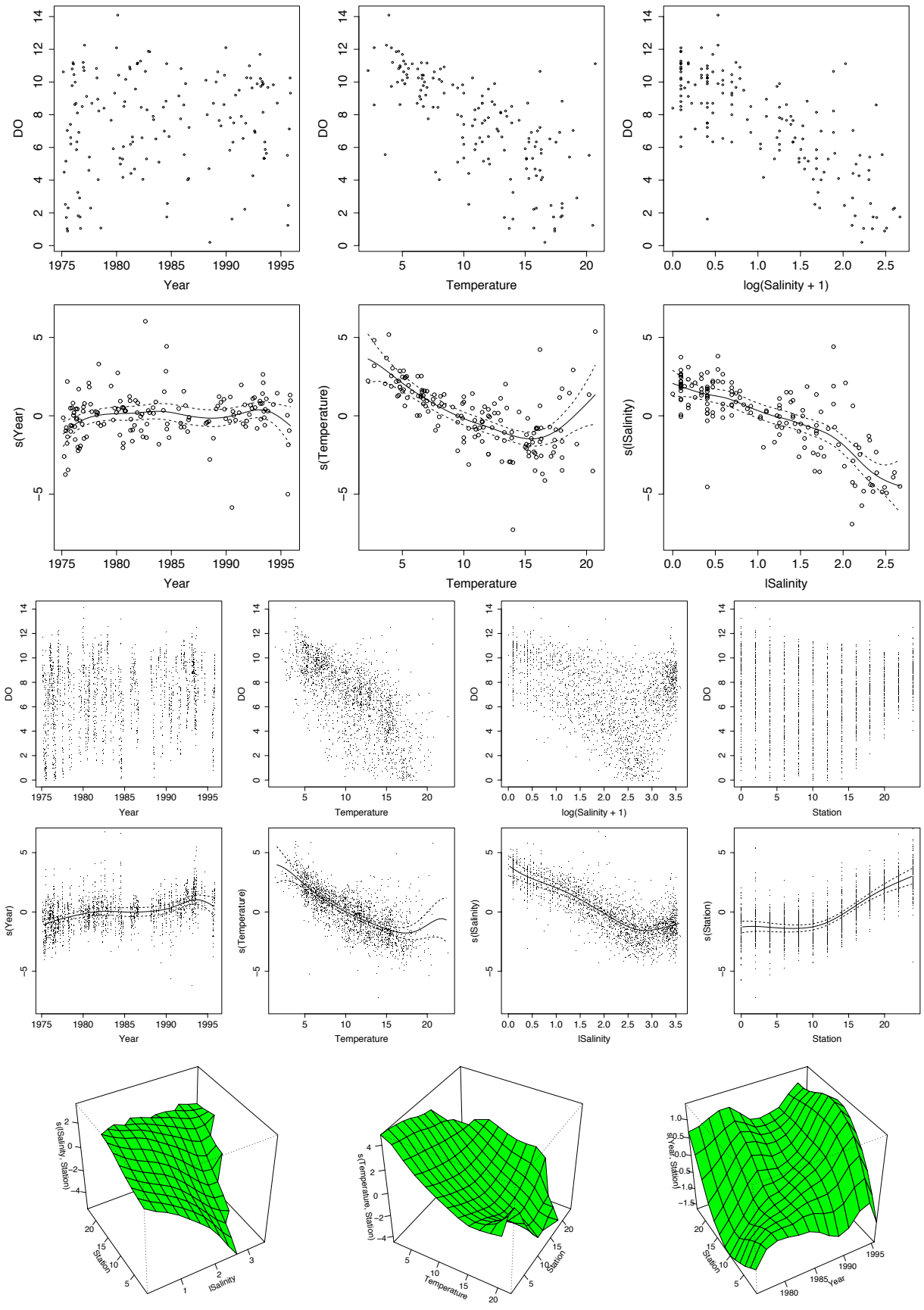
We can also express these in terms of projection matrices.

$$P_j^{(l)} = (I_n - P_0) S_j (I_n - \sum_{k<j} P_k^{(l)} - \sum_{k>j} P_k^{(l-1)}),$$

$$\hat{y} = Py = (P_0 + \sum_{j=1}^{p} P_j) y$$

If a regression splines or p-splines model is adopted, the each of the functions $m_i(x)$ is represented by a linear expression and so the model itself remains linear. It can then be fitted be standard linear regression, incorporating a set of penalties in the p-splines case. This has the advantage of direct, rather than iterative, fitting but it has the potential disadvantage of needing to invert very large matrices if the model has many terms.

The plots below show data from a survey of dissolved oxygen (DO) in the RIver Clyde at a single sampling station, related to potential explanatory variables of interest. The additive terms usefully capture the underlying trends. The following plots build a model for the whole river, using data at many sampling stations. Some care has to be taken here because of the repeated measures nature of the data. Also, we are likely to need interaction terms and these are shown in the surface plots.


DO
Year
Temperature
log(Salinity + 1)
s(Year)
s(Temperature)
s(lSalinity)
lSalinity
Station
s(Station)
s(lSalinity, Station)
s(Temperature, Station)
s(Year, Station)

4.6 Comparing additive models

While models of this type provide very flexible and visually informative descriptions of the data, it is also necessary to consider how models can be compared and inferences drawn. Hastie and Tibshirani (1990) recommend the use of residual sums-of-squares and their associated approximate degrees of freedom to provide guidance for model comparisons.

For an additive model, the residual sum-of-squares can easily be defined as

$$\mathrm{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2,$$

where $\hat{y}_i$ denotes the fitted value, produced by evaluating the additive model at the observation x_i. We can write the residual sum-of-squares as

$$\mathrm{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = y^\top (I - P)^\top (I - P) y,$$

where P denotes the projection matrix discussed earlier. The approximate degrees of freedom for error can be defined as

$$\mathrm{df} = \mathrm{tr}\left\{ (I - P)^\top (I - P) \right\}.$$

In an obvious notation, comparisons of two models can expressed quantitatively in

$$F = \frac{(\mathrm{RSS}_2 - \mathrm{RSS}_1)/(\mathrm{df}_2 - \mathrm{df}_1)}{\mathrm{RSS}_1/\mathrm{df}_1},$$

by analogy with the F-statistic used to compare linear models. Unfortunately, this analogy does not extend to distributional calculations and no general expression for the distribution of (**??**) is available. However, Hastie and Tibshirani (1990, sections 3.9 and 6.8) suggest that at least some approximate guidance can be given by referring the observed nonparametric F-statistic to an F distribution with $(\mathrm{df}_2 - \mathrm{df}_1)$ and df_1 degrees of freedom. Wood (2006) gives a formulation in terms of testing whether relevant spline coefficients might be 0.

The reef data provide a simple illustration of how model comparisons may be made. There are three obvious models of interest.

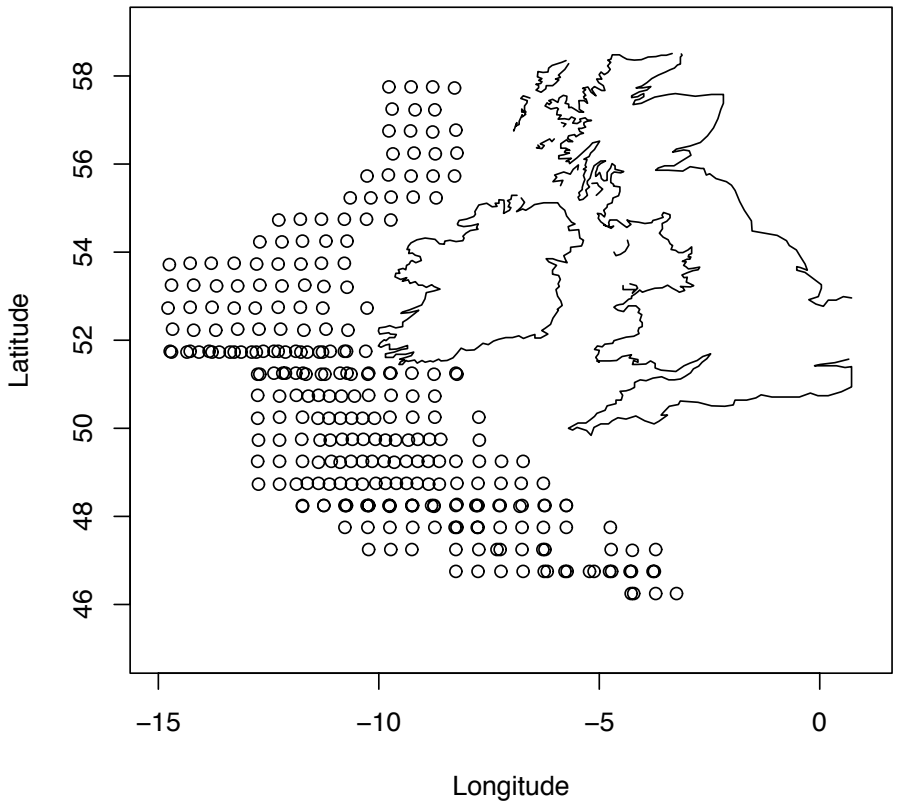
Model	RSS	df
1: $\beta_0 + m_1(\text{Latitude}) + m_2(\text{Longitude})$	4.541	33.99
2: $\beta_0 + m_2(\text{Longitude})$	6.128	37.46
3: $\beta_0 + m_1(\text{Latitude})$	27.306	37.54

The observed F-statistic for the latitude component is

$$\frac{(6.128 - 4.541)/(37.46 - 33.99)}{4.541/33.99} = 3.42.$$

Referring this to an $F_{3.47,33.99}$ distribution produces an approximate p-value of 0.023. This therefore suggests that there is some evidence that the underlying regression surface changes with latitude.

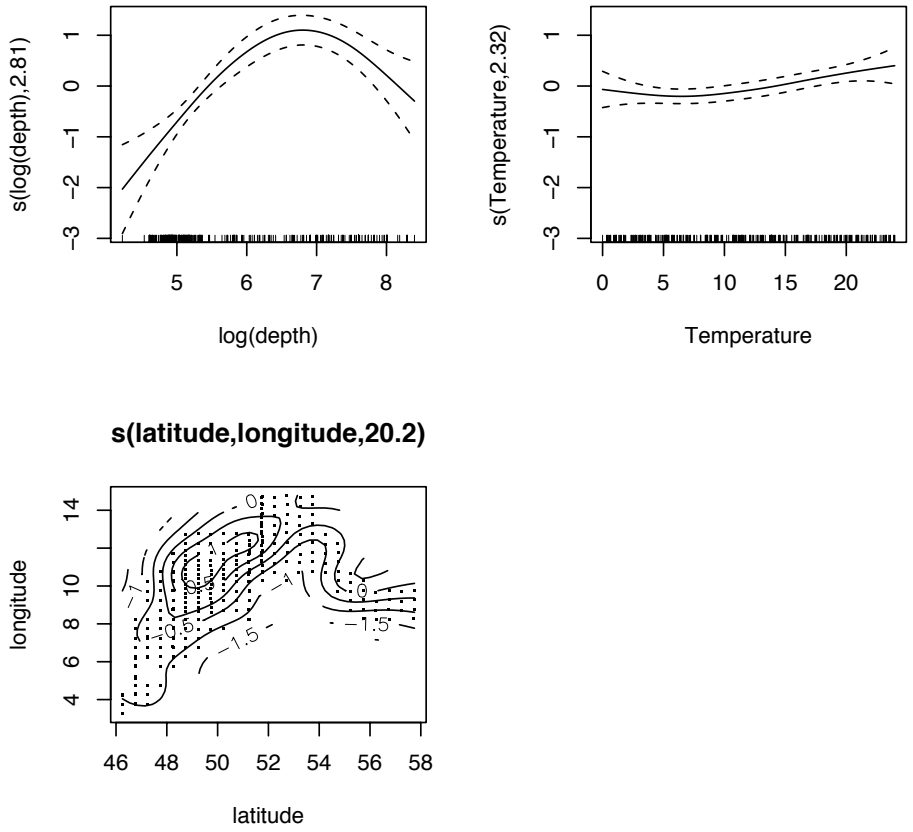
The observed F-statistic for the longitude component is 48.08 on 3.54 and 33.99 degrees of freedom, which is highly significant, and so confirms the importance of the effect of longitude.



A further example uses data from a multi-country survey of mackerel eggs in the Eastern Atlantic. An additive model for egg density might reasonably contain terms for depth and temperature, plus a joint term for latitude and longitude, to reflect spatial position. This leads to the model

$$y = \beta_0 + m_{12}(x_1, x_2) + m_3(x_3) + m_4(x_4) + \varepsilon,$$

where m_{12} represents a smooth two-dimensional function of latitude (x_1) and longitude (x_2), and m_3 and m_4 represent additive terms of the usual type for depth (x_3) and temperature (x_4). Two-dimensional terms require restrictions to define the functions uniquely, as in the one-dimensional case. A simple convention is $\sum_{i=1}^{n} m_{12}(x_{1i}, x_{2i}) = 0$.



Model	RSS	df
1: $\beta_0 + m_{12}(\text{Lat, Long}) + m_3(\log(\text{Depth})) + m_4(\text{Temp})$	261.13	262.80
2: $\beta_0 + m_{12}(\text{Lat, Long}) + m_4(\text{Temp})$	360.24	266.51
3: $\beta_0 + m_{12}(\text{Latitude, Longitude}) + m_3(\log(\text{Depth}))$	272.08	266.10
4: $m_3(\text{Depth}) + m_4(\text{Temp})$	335.53	270.99

The large change in residual sum-of-squares between models 1 and 2 confirms that depth is an important variable. Similarly, the change between models 1 and 4 shows that there are additional geographical effects which should be accounted for in the model by the presence of the term involving latitude and longitude. However, the F-value for the temperature effect, namely

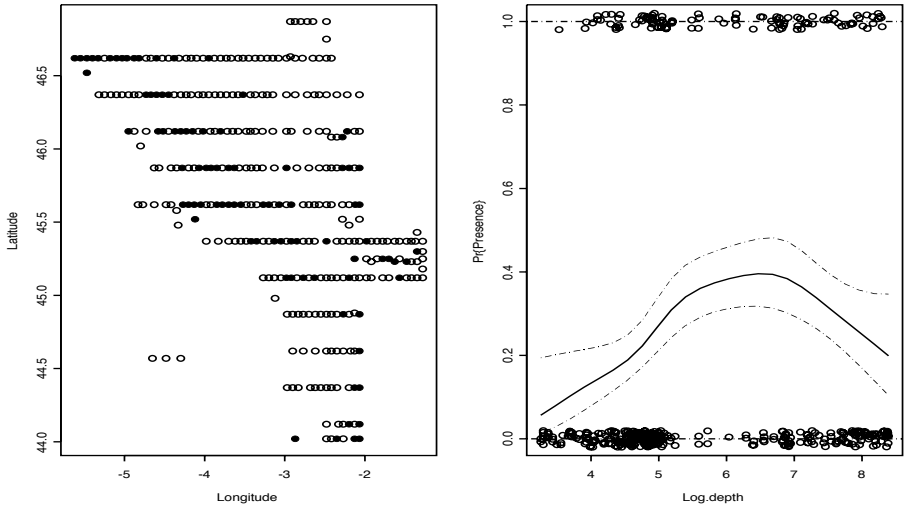
$$\frac{(272.08 - 261.13)/(266.10 - 262.80)}{262.13/262.80} = 3.33$$

when compared to an $F_{3.30,262.80}$ distribution suggests that the effect of temperature may also be significant.

4.7 Generalised additive models

In the mackerel survey above, the data collected by Spanish vessels in the Bay of Biscay exhibit rather different features from the remainder of the survey. One of these features

is that no eggs were detected at all at a substantial number of the sampling points. The sampling positions, together with an indication of the presence or absence of eggs, are displayed below. Depth and sea surface temperature are again available as potential covariates.



The right panel shows a local logistic regression curve fitted to the relationship between presence and depth, on a log scale. As in the earlier investigation of the density of eggs, log(depth) appears to have an approximately quadratic effect on presence, with an optimal depth around $\exp(6) \approx 400$m.

The framework of generalized linear models is extremely useful in providing a unified description of a very wide class of parametric models, along with methods of fitting and analyses. In the nonparametric setting a corresponding framework of generalised additive models provides a very flexible form of extension.

Excellent overviews of this area are provided in the texts by Hastie and Tibshirani (1990), Green and Silverman (1994) and Wood (2006). A brief introduction to the main ideas is provided in this section, based on an illustration. Another very useful text is Ruppert *et al.* (2003) which discusses *semiparametric regression.*

In the case of logistic regression, a linear model with four covariates takes the form

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4,$$

A logistic form of a generalised additive model therefore extends this by replacing each linear component with a nonparametric one.

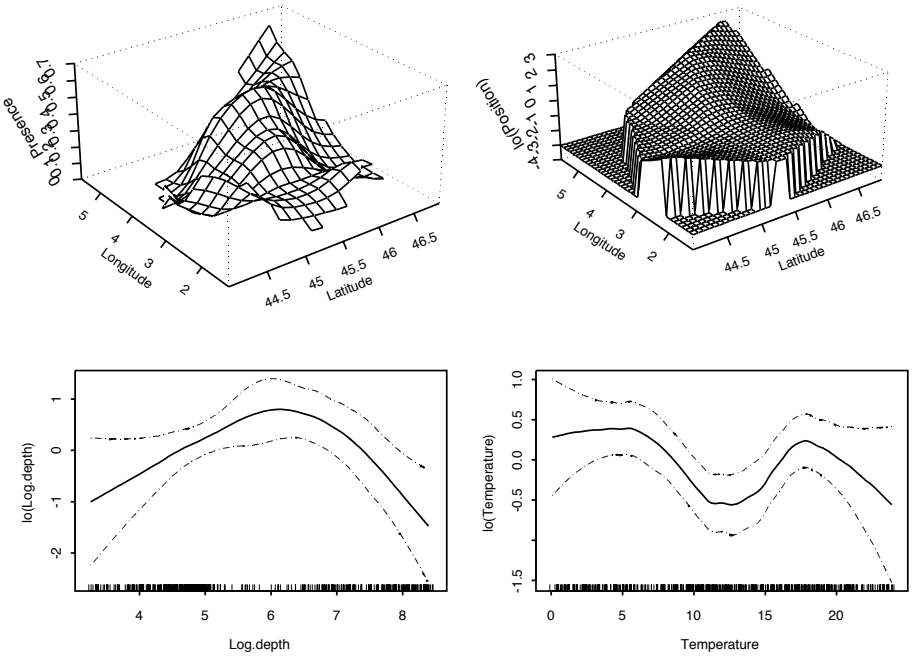
For the Spanish mackerel data, where the covariates represent latitude, longitude, log(depth) and temperature respectively, a natural model is

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + m_{12}(x_1, x_2) + m_3(x_3) + m_4(x_4),$$

since latitude and longitude merely define a convenient two-dimensional co-ordinate system.

In order to fit a generalised additive model the corresponding fitting procedure for generalised linear models again provides a helpful guide. The likelihood function is the natural starting point and a Newton-Raphson, or Fisher scoring, procedure allows the parameter estimates to be located by an iterative algorithm. Each step of these algorithms can be formulated as a weighted least squares linear regression. This immediately provides provides a natural analogue in the generalised additive model setting, by employing a weighted nonparametric regression at each step.

Hastie & Tibshirani (1990), Green & Silverman (1994) and Wood (2006) provide detailed discussion of this approach.



The expected concave effect of log(depth) is apparent. A more complex non-linear curve describes the effect of temperature and a two-dimensional function in latitude and longitude captures the additional spatial variation.

In order to explore which of these terms can be regarded as evidence of an underlying systematic effect, rather than random variation, different models for the data can be compared. A deviance can also be defined in an analogous manner for a generalised additive model. As with additive models for data with normal responses, general distribution theory for model comparisons is not available, even asymptotically. However, by applying suitable quadratic approximations, degrees of freedom can be associated with each model, and so some guidance on model comparisons can be taken by comparing differences in deviance to χ^2 distributions indexed by the difference in the approximate degrees of freedom.

For the Spanish mackerel data, the deviances for a number of models of interest are shown below.

Model	Deviance	df
1: $\beta_0 + m_{12}(\text{Lat, Long}) + m_3(\log(\text{Depth})) + m_4(\text{Temp})$	384.22	401.30
2: $\beta_0 + m_{12}(\text{Lat, Long}) + m_4(\text{Temp})$	394.32	404.75
3: $\beta_0 + m_{12}(\text{Lat, Long}) + m_3(\log(\text{Depth}))$	398.43	404.68
4: $m_3(\text{Depth}) + m_4(\text{Temp})$	431.95	409.20

An analysis of this type suggests that all of the three components in model 1 contribute significantly. The temperature effect looks rather implausible from a scientific point of view and should perhaps be treated with some caution. Hastie and Tibshirani (1986, section 6.10) provide some discussion on the dangers of over-interpreting additive fits in models of this kind.

5

Kernel methods, Gaussian Processes and Support Vector Machines

5.1 Making linear methods nonlinear

Most of the basic statistical methods are "linear". There are many examples of such methods

– In linear regression the mean response is assumed to be a linear function $\mathbb{E}(Y_i) = \mathbf{x}_i^\top \boldsymbol{\beta}$.
– In linear discriminant analysis the different classes are separated by linear hyperplanes.
– In linear principal component analysis the linear projection of the data which has the largest variance is sought.

This assumption of linearity typically yields simple computations and allows for deriving a simple theoretical framework. In this course you have already seen two ways of turning a linear method into a non-linear method.

Basis expansions This approach is based on considering a function $\mathbf{B}(\mathbf{x})$ rather than $(\mathbf{1}, \mathbf{x})$ itself when constructing the linear combination, i.e. we consider $\mathbf{B}(\mathbf{x})^\top \boldsymbol{\beta}$ instead of $(\mathbf{1}, \mathbf{x})^\top \boldsymbol{\beta}$. Because $\mathbf{B}(\cdot)$ is a non-linear function $\mathbb{E}(Y_i)$ is now a non-linear function of x_i. Polynomial regression and splines are an example of this method.

Local methods In chapter 2 we have studied locally linear methods. The basic idea was to use a weighted linear model in which the weights change for every prediction. Just like in density estimation we have used a kernel function to give observed data close to the observation fort which we want to predict a larger weight that observed data which is further away.

In this chapter we will focus on kernel methods. Kernel methods are a class of methods which were proposed in the Machine Learning community in the early 1990s. Just like kernel density estimation and local linear methods these methods use a kernel function,

which often is the Gaussian kernel. However in kernel methods the kernel is used as as inner product, i.e. as a measure of angle and lengths. In linear methods we often compute quantities like $\mathbf{x}_i^\top \mathbf{x}_j$, which is a special case of an inner product. Often (but not always) all computations done by a linear method can be written only using such inner products. We can turn such a linear method into nonlinear method by using a different inner product, i.e. by measuring angles and lengths differently using the kernel function instead of the standard inner product.

Kernel functions - notation

In section 1.2 we have seen kernel density estimation. We have estimated the density at x_0 by

$$m(\hat{x}_0) = \frac{\sum_{i=1}^n w(x_i - x_0; h)}{n}$$

We have referred to the function $w(\cdot)$, which takes the scaled difference between x_i and x_0 as argument as *kernel function*. A popular choice of $w(\cdot)$ is the Gaussian kernel, which corresponds to $w(t; h) = \frac{1}{\sqrt{2\pi}} \exp(-t^2/2h)$.

Because kernel methods were first proposed in the machine learning community, kernel methods use a slightly different notation. We will also use this notation in this chapter. Basically the difference is that we write the kernel function as function with two arguments. In this notation we would write the kernel density estimate as

$$m(\hat{x}_0) = \frac{\sum_{i=1}^n K_h(x_i, x_0)}{n}$$

If we set $K_h(x_1, x_2) = w(x_1 - x_2; h)$ we can recover the notation from chapters 1 and 2. In this new notation the univariate Gaussian kernel is $K_h(x_1, x_2) = \frac{1}{\sqrt{2\pi}} \exp(-(x_1 - x_2)^2/2h)$.

This new notation is more general as it allows kernels which are not a function of the difference between x_1 and x_2, such as $K(x_1, x_2) = x_1 x_2$.

5.2 Kernelisation

5.2.1 Kernel ridge regression

Ridge regression revisited Recall from section 3.3 that the ridge regression estimate of the regression coefficient in linear regression is given by

$$\hat{\boldsymbol{\beta}}_{\text{ridge}} = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_p)^{-1} \mathbf{X}^\top \mathbf{y}$$

Using the identity $(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_p)^{-1} \mathbf{X}^\top = \mathbf{X}^\top (\mathbf{X}\mathbf{X}^\top + \lambda \mathbf{I}_n)^{-1}$, which holds for $\lambda > 0$, we can rewrite $\hat{\boldsymbol{\beta}}_{\text{ridge}}$ as

$$\hat{\boldsymbol{\beta}}_{\text{ridge}} = \mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I}_n)^{-1}\mathbf{y}.$$

The prediction for a new observation with covariates $\mathbf{x}_0^\top$ can thus be written as

$$\hat{y}_0 = \mathbf{x}_0^\top\hat{\boldsymbol{\beta}}_{\text{ridge}} = \mathbf{x}_0^\top\mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I}_n)^{-1}\mathbf{y} = \mathbf{k}_0(\mathbf{K} + \lambda\mathbf{I}_n)^{-1}\mathbf{y},$$

where $\mathbf{k}_0 = \mathbf{x}_0^\top\mathbf{X}^\top$ and $\mathbf{K} = \mathbf{X}\mathbf{X}^\top$. At first sight it seems unclear whether this new way of writing down the estimate from ridge regression is of any benefit to us. The most obvious difference is that instead of inverting a $p \times p$ matrix, we now need to invert a $n \times n$ matrix, which unless $p > n$ is not an advantage. In order to realise the benefits of this new way of writing down the solution to ridge regression we need to take a closer look at the entries of the vector $\mathbf{k}_0$ and the matrix $\mathbf{K}$.

$$\mathbf{k}_0 = \mathbf{x}_0^\top\mathbf{X}^\top = \left(\sum_{j=1}^p x_{0j}x_{1j}, \ldots, \sum_{j=1}^p x_{0j}x_{1j}\right) = \left(\mathbf{x}_o^\top\mathbf{x}_1, \ldots, \mathbf{x}_o^\top\mathbf{x}_n\right) = \left(\langle\mathbf{x}_o, \mathbf{x}_1\rangle, \ldots, \langle\mathbf{x}_o, \mathbf{x}_n\rangle\right)$$

$$\mathbf{K} = \mathbf{X}\mathbf{X}^\top = \begin{pmatrix} \sum_{j=1}^p x_{1j}^2 & \ldots & \sum_{j=1}^p x_{1j}x_{nj} \\ \vdots & \ddots & \vdots \\ \sum_{j=1}^p x_{nj}x_{1j} & \ldots & \sum_{j=1}^p x_{nj}^2 \end{pmatrix} = \begin{pmatrix} \mathbf{x}_1^\top\mathbf{x}_1 & \ldots & \mathbf{x}_1^\top\mathbf{x}_n \\ \vdots & \ddots & \vdots \\ \mathbf{x}_n^\top\mathbf{x}_1 & \ldots & \mathbf{x}_n^\top\mathbf{x}_n \end{pmatrix}$$

$$= \begin{pmatrix} \langle\mathbf{x}_1, \mathbf{x}_1\rangle & \ldots & \langle\mathbf{x}_1, \mathbf{x}_n\rangle \\ \vdots & \ddots & \vdots \\ \langle\mathbf{x}_n, \mathbf{x}_1\rangle & \ldots & \langle\mathbf{x}_n, \mathbf{x}_n\rangle \end{pmatrix}$$

We can see that both the entries of the matrix $\mathbf{k}_0$ and the matrix $\mathbf{K}^\top$ and thus also $\hat{y}_0$ only depend on the covariates through inner products, i.e. measures of angles and length. Thus we can kernel ridge regression a nonlinear regression technique by using a different inner product.

Non-linear ridge regression We start by using the same idea as we have used when looking at basis expansions. Rather than working with the data $\mathbf{x}_i$ itself, we work with a basis expansion $\boldsymbol{\phi}(\mathbf{x}_i)$, i.e. an extended design matrix $\mathbf{B} = \begin{pmatrix} \boldsymbol{\phi}(\mathbf{x}_1)^\top \\ \vdots \\ \boldsymbol{\phi}(\mathbf{x}_n)^\top \end{pmatrix}$.[1] Now $\mathbf{k}_0$ and $\mathbf{K}$ become, using $\mathbf{b}_0 = \boldsymbol{\phi}(\mathbf{x}_0)$,

$$\mathbf{k}_0 = \mathbf{b}_0^\top\mathbf{B}^\top = \left(\langle\boldsymbol{\phi}(\mathbf{x}_o), \boldsymbol{\phi}(\mathbf{x}_1)\rangle, \ldots, \langle\boldsymbol{\phi}(\mathbf{x}_o), \boldsymbol{\phi}(\mathbf{x}_n)\rangle\right)$$

$$\mathbf{K} = \mathbf{B}\mathbf{B}^\top = \begin{pmatrix} \langle\boldsymbol{\phi}(\mathbf{x}_1), \boldsymbol{\phi}(\mathbf{x}_1)\rangle & \ldots & \langle\boldsymbol{\phi}(\mathbf{x}_1), \boldsymbol{\phi}(\mathbf{x}_n)\rangle \\ \vdots & \ddots & \vdots \\ \langle\boldsymbol{\phi}(\mathbf{x}_n), \boldsymbol{\phi}(\mathbf{x}_1)\rangle & \ldots & \langle\boldsymbol{\phi}(\mathbf{x}_n), \boldsymbol{\phi}(\mathbf{x}_n)\rangle \end{pmatrix}$$

[1] In the lecture on basis expansions we have called the functions $\mathbf{B}(\cdot)$, rather than $\boldsymbol{\phi}(\cdot)$. In the kernel literature the function is however almost always called $\boldsymbol{\phi}(\cdot)$, so we will use this notation as well.

The function $\boldsymbol{\phi}(\cdot)$ appears only inside inner products, i.e. we do not need to know much about $\boldsymbol{\phi}(\cdot)$, except for how to compute inner products

$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle .$$

Actually, we don't even need to specify $\boldsymbol{\phi}(\cdot)$, we can simply write down the function $k(\cdot, \cdot)$. Before we spend more time looking at possible choices for this kernel function $k(\cdot, \cdot)$ we rewrite the solution to kernel ridge regression so that we can see an interesting theoretical pattern. Using this new kernel function $k(\cdot, \cdot)$ we can write

$$\mathbf{k}_0 = (k(\mathbf{x}_o, \mathbf{x}_1), \ldots, k(\mathbf{x}_o, \mathbf{x}_n)) \qquad \mathbf{K} = \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \ldots & k(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & \ldots & k(\mathbf{x}_n, \mathbf{x}_n) \end{pmatrix}$$

Using $\boldsymbol{\alpha} = (\mathbf{K} + \lambda \mathbf{I}_n)^{-1}\mathbf{y}$ we can rewrite $\hat{y}_0$ as

$$\hat{y}_0 = \mathbf{k}_0^\top \underbrace{(\mathbf{K} + \lambda \mathbf{I}_n)^{-1}\mathbf{y}}_{=\boldsymbol{\alpha}=(\alpha_1,\ldots,\alpha_n)} = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_0, \mathbf{x}_i),$$

i.e. the prediction $\hat{y}_0$ is just a linear combination of kernel functions. This simple form is no coincidence. Later on in this chapter we will derive a result that states that optimal solutions to penalised fitting criteria must be of this simple form.

Inner products

An inner product space is a vector space with a function $\langle \cdot, \cdot \rangle$, called inner product, which satisfies the following three properties.

(Conjugate) symmetry $\langle \mathbf{x}, \mathbf{y} \rangle = \overline{\langle \mathbf{y}, \mathbf{x} \rangle}$ (in case of a real-valued inner products $\langle \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{y}, \mathbf{x} \rangle$)

Linearity $\langle \alpha \mathbf{x}_1 + \beta \mathbf{x}_2, \mathbf{y} \rangle = \alpha \langle \mathbf{x}_1, \mathbf{y} \rangle + \beta \langle \mathbf{x}_2, \mathbf{y} \rangle$.

Together with the (conjugate) symmetry, linearity in the first argument implies (conjugate) symmetry in the second argument.

Positive-definiteness $\langle \mathbf{x}, \mathbf{x} \rangle \geq 0$ with equality if and only if $\mathbf{x} = \mathbf{0}$.

One can show that then $\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$ is a norm and thus $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\| = \sqrt{\langle \mathbf{x} - \mathbf{y}, \mathbf{x} - \mathbf{y} \rangle}$ is a distance. Examples of inner products are:

- In the vector space of $\mathbb{R}^p$ the classical inner product $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{p} x_i y_i$ satisfies the above definition.
- In the vector space of $\mathbb{C}^p$ the classical inner product $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{p} x_i \bar{y}_i$ satisfies the above definition.
- In the vector space of random variables with finite variance, the covariance $\mathrm{Cov}(X, Y)$ satisfies the properties of an inner product.

5.2.2 Choosing a kernel

We have seen above that the predictions in ridge regression only depend on the covariates through inner products $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle$, so rather than choosing $\boldsymbol{\phi}(\cdot)$ we can choose $k(\cdot, \cdot)$. There are however some constraints as $k(\mathbf{x}_i, \mathbf{x}_j)$ has to be a valid inner product.

An important result from functional analysis, called Mercer's theorem[2], guarantees that if

i. $k(\cdot, \cdot)$ is symmetric in its arguments, i.e. $k(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_j, \mathbf{x}_i)$, and
ii. $k(\cdot, \cdot)$ is positive semi-definite, i.e. for any choice of $\mathbf{x}_1, \ldots, \mathbf{x}_n$ the matrix

$$\mathbf{K} = \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \ldots & k(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & \ldots & k(\mathbf{x}_n, \mathbf{x}_n) \end{pmatrix}$$

is positive semi-definite,

then there exists a unique basis expansion ("feature map") $\boldsymbol{\phi}(\cdot)$ such that $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle$.

Examples of kernel functions are the usual dot product in $\mathbb{R}^p$

$$k(\mathbf{x}_i, \mathbf{x}_j) := \mathbf{x}_i^\top \mathbf{x}_j.$$

Using this kernel simply corresponds to linear ridge regression. Other choices of kernels are the homogeneous polynomial kernel ($q \in \mathbb{N}$)

$$k(\mathbf{x}_i, \mathbf{x}_j) := (\mathbf{x}_i^\top \mathbf{x}_j)^q$$

and the Gaussian kernel ($\gamma > 0$)

$$k(\mathbf{x}_i, \mathbf{x}_j) := \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2\right).$$

Using the polynomial kernel or the Gaussian kernel leads to a non-linear fitted function. For the Gaussian kernel the corresponding $\boldsymbol{\phi}(\cdot)$ is infinite-dimensional and cannot be written down in closed form.

There are of course many other possible kernel functions. The properties a valid covariance function of a stochastic process must have are the same as the properties a kernel function must have, so every covariance function can be a kernel and vice versa.

The kernel trick also allows incorporating non-numeric covariates. All we need to be able to do is construct a function $k(\cdot, \cdot)$ which expresses some of distance between its

[2] named after James Mercer FRS (1883 –1932), a British mathematician.

two arguments. Kernels have, for example, been constructed that work on DNA data and text.

Many linear methods commonly used in Statistics can be kernelised: principal component analysis, (penalised) discriminant analysis, etc. There are however some exceptions: linear regression, which is nothing other than ridge regression with $\lambda = 0$, cannot be kernelised.

5.2.3 Reproducing Kernel Hilbert Spaces and the Representer Theorem

We have seen that the optimal solution in kernel ridge can be written as $\hat{m}(\mathbf{x}_0) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_0, \mathbf{x}_i)$. In this section we will generalise this result.

Let $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ be a kernel. We have then seen that we can construct a mapping $\boldsymbol{\phi}(\cdot)$ from $\mathcal{X}$ to a suitable inner product space such that

$$\langle \phi(x), \phi(x') \rangle = k(x, x') \qquad \forall x, x' \in \mathcal{X}.$$

Furthermore the Riesz representation theroem[3] tells us that in a reproducing kernel Hilbert space[4] k is the so-called representer of evaluation, i.e. for all $x \in \mathcal{X}$ and all functions $f \in \mathcal{H}$ we have that

$$m(x) = \langle f, k(\cdot, x) \rangle$$

Having set out the mathematical foundations we will now return to the our statistical problem. In ridge regression we optimise the criterion

$$\underbrace{\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2}_{\text{empirical loss on training data}} + \lambda \underbrace{\|\boldsymbol{\beta}\|^2}_{\text{penalty}},$$

We will now generalise this result using a generic loss function L instead of the least squares loss $\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$. More precisely, let

$$\begin{aligned} L \;:\; & \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}, \\ & (y_1, \ldots, y_n, m(\mathbf{x}_1), \ldots, m(\mathbf{x}_n)) \mapsto L(y_1, \ldots, y_n, m(\mathbf{x}_1), \ldots, m(\mathbf{x}_n)) \end{aligned}$$

be a pointwise defined loss function which associates a loss L to a set of predictions $m(\mathbf{x}_1), \ldots, m(\mathbf{x}_n)$ with respect to the observed responses $y_1, \ldots, y_n$ ($\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$, $f \in \mathcal{H}$).

[3] named after Frigyes Riesz (1880 – 1956), a Hungarian mathematician

[4] A Hilbert space is an inner product space which is complete, i.e. every Cauchy sequence converges. A reproducing kernel Hilbert space is a Hilbert space of functions where the linear map δ_x which maps each function $m(\cdot)$ to the value $m(x)$ it takes at some x is continuous for every choice of x. Essentially, a reproducing kernel Hilbert space is a "reasonably well behaved" Hilbert space.

Similarly we will consider a more general penalty $\Omega(\|f\|^2)$, where $\Omega : [0, +\infty] \to \mathbb{R}$ is a strictly increasing function.

We now want to find the function $m(\cdot)$ in the Hilbert space $\mathcal{H}$ induced by the kernel $k(\cdot, \cdot)$ which minimises the regularised loss ($\lambda \in \mathbb{R}^+$)

$$L(y_1, \ldots, y_n, m(\mathbf{x}_1), \ldots, m(\mathbf{x}_n)) + \lambda \cdot \Omega(\|f\|^2). \tag{5.1}$$

This is similar to the tasked studied is section 3.2.3.

We will now show that even though the space $\mathcal{H}$ might be of infinite dimension, we will show that the minimisation problem is actually finite-dimensional as any minimiser of (5.1) in $\mathcal{H}$ admits the representation

$$\hat{m}(\mathbf{x}_0) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x}_0) \tag{5.2}$$

for suitable $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$. In other words, the minimisation problem is only finite-dimensional as we just have to find the right $\alpha_1, \ldots, \alpha_n$. This is an extremely powerful result, which was first derived in he late 1970s by Kimeldorf and Wahba.

In the remainder of this section we will prove this important result. We start by assuming that we have a minimiser $f \in \mathcal{H}$ and will show that we can write it as $m(\cdot) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \cdot)$.

i. As $k(x_i, \cdot) \in \mathcal{H}$, we can decompose any function f into a part $f_\parallel \in \text{span}\,(k(x_1, \cdot), \ldots, k(x_n, \cdot))$ and a part $f_\perp$ that is orthogonal to the span (i.e. $\langle f_\perp, k(x_i, \cdot)\rangle = 0$). Furthermore we can find $\alpha_1, \ldots, \alpha_n$ such that $f_\parallel = \sum_{i=1}^{n} \alpha_i k(x_i, \cdot)$. Thus

$$m(\cdot) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \cdot) + f_\perp(\cdot), \tag{5.3}$$

ii. We will now show that at any of the training points $\mathbf{x}_j$ we can compute $m(\cdot)$ as

$$m(\mathbf{x}_j) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x}_j).$$

We can show this by using the representer property

$$\begin{aligned} m(\mathbf{x}_j) &= \langle f, k(\cdot, \mathbf{x}_j)\rangle = \left\langle \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \cdot) + f_\perp(\cdot), k(\cdot, \mathbf{x}_j) \right\rangle = \\ &= \sum_{i=1}^{n} \alpha_i \underbrace{\langle k(\mathbf{x}_i, \cdot), k(\cdot, \mathbf{x}_j)\rangle}_{=k(\mathbf{x}_i, \mathbf{x}_j)} + \underbrace{\langle f_\perp(\cdot), k(\cdot, \mathbf{x}_j)\rangle}_{=0} \\ &= \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x}_j) \end{aligned}$$

iii. We can now rewrite the penalty as

$$\Omega(\|f\|^2) = \Omega\Bigg(\underbrace{\left\|\sum_{i=1}^n \alpha_i k(\mathbf{x}_i,\cdot) + f_\perp(\cdot)\right\|^2}_{=\|\sum_{i=1}^n \alpha_i k(\mathbf{x}_i,\cdot)\|^2 + \underbrace{\|f_\perp(\cdot)\|^2}_{\geq 0}}\Bigg) \geq \Omega\left(\left\|\sum_{i=1}^n \alpha_i k(\mathbf{x}_i,\cdot)\right\|^2\right)$$

iv. We have just seen in ii. that $f_\perp(\cdot)$ has no influence on $m(\mathbf{x}_j)$ at the training points $\mathbf{x}_j$, so $f_\perp(\cdot)$ has no influence on the training loss $L(y_1,\ldots,y_n,m(\mathbf{x}_1),\ldots,m(\mathbf{x}_n))$. We have also seen in iii. that $f_\perp(\cdot)$ only makes $\Omega(\|f\|^2)$ bigger, so a minimser must have $f_\perp(\cdot) = 0$, i.e. $f \in \text{span}\,(k(\mathbf{x}_1,\cdot),\ldots,k(\mathbf{x}_n,\cdot))$, which is nothing other than $m(\cdot) = \sum_{i=1}^n \alpha_i k(\mathbf{x}_i,\cdot)$.

Note that the proof is very similar to the proof of theorem 3.5 — just view $\int_a^b m''(x)g''(x)\,dx$ as an inner product between the two functions $m(\cdot)$ and $g(\cdot)$.

5.3 Gaussian processes

5.3.1 Ridge regression from a Bayesian point of view

In this section we will study Gaussian processes. Gaussian processes have been used for decades in geostatistics where they are often referred to kriging models[5]. About 10 to 15 years ago they became popular in the machine learning community.

Just like in section 5.2.1 we will start with ridge regression, however in this session we will focus on its Bayesian interpretation.

Remember that in ridge regression we wanted to find the regression parameters $\boldsymbol{\beta}$ which minimise the penalised least squares criterion

$$\sum_{i=1}^n (y_i - \mathbf{x}_i^\top \boldsymbol{\beta})^2 + \frac{\sigma^2}{\tau^2}\sum_{j=1}^p \beta_j^2,$$

where we have written $\lambda = \frac{\sigma^2}{\tau^2}$. The objective function of ridge regression can be given a Bayesian interpretation. Suppose that we want to use a normal distribution with mean 0 and variance τ^2 as prior distribution for the regression coefficients β_j and assume that each observation has, given $\boldsymbol{\beta}$, a normal distribution with mean $\mathbf{x}_i^\top\boldsymbol{\beta}$ and variance σ^2, i.e.

$$\boldsymbol{\beta} \sim \mathsf{N}(\mathbf{0},\tau^2\mathbf{I}) \tag{5.4}$$

$$y_i|\boldsymbol{\beta} \sim \mathsf{N}(\mathbf{x}_i^\top\boldsymbol{\beta},\sigma^2) \tag{5.5}$$

[5] named after Daniel Gerhardus Krige, a South African mining engineer and professor at the University of the Witwatersrand, who first suggested kriging to model mineral deposits.

To keep things simple we shall assume for the moment that the variance σ^2 is known.[6] Then we can write the p.d.f. of the posterior distribution of $\boldsymbol{\beta}$ as

$$f(\boldsymbol{\beta}|y_1,\ldots,y_n) \propto \underbrace{\left(\prod_{i=1}^{n} f(y_i|\boldsymbol{\beta})\right)}_{\text{Likelihood}} \cdot \underbrace{f(\boldsymbol{\beta})}_{\text{prior}}$$
$$= \left(\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - \mathbf{x}_i^\top\boldsymbol{\beta})^2}{2\sigma^2}\right)\right) \cdot \left(\frac{1}{\sqrt{2\pi\tau^2}}\right)^p \exp\left(-\frac{\sum_{j=1}^{p}\beta_j^2}{2\tau^2}\right)$$

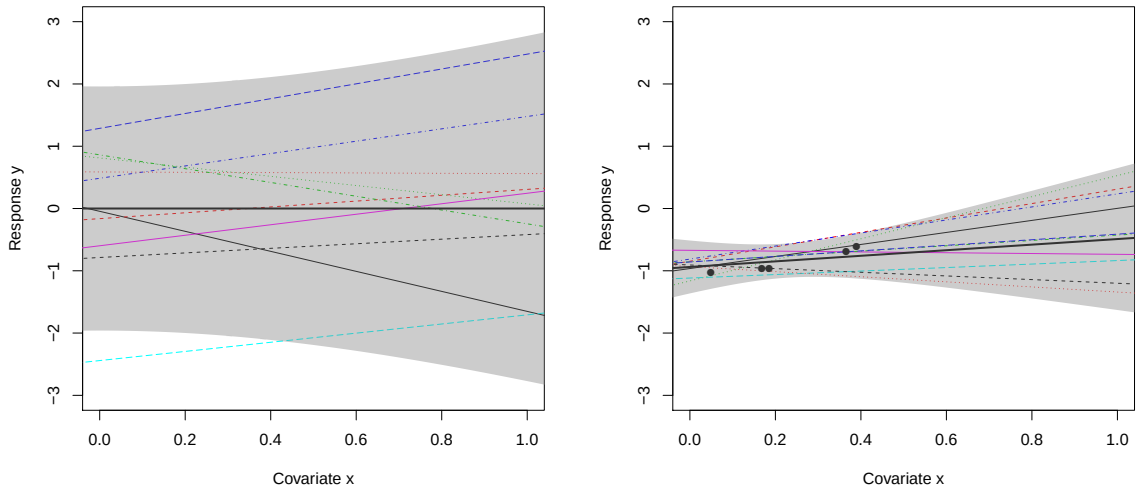
Collecting terms, taking logs and keeping only terms involving $\boldsymbol{\beta}$ yields the log-posterior density

$$\log f(\boldsymbol{\beta}|y_1,\ldots,y_n) = \text{const} - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^\top\boldsymbol{\beta})^2 - \frac{1}{2\tau^2}\sum_{j=1}^{p}\beta_j^2,$$

which is, up to a multiplicative constant, the objective function used in ridge regression. The estimated regression coefficient $\hat{\boldsymbol{\beta}}^{\text{ridge}} = (\mathbf{X}^\top\mathbf{X} + \frac{\sigma^2}{\tau^2}\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}$ is thus the Bayesian maximum-a-posteriori (MAP) estimate of $\boldsymbol{\beta}$.

In full Bayesian inference we do not just look for the maximiser of the posterior, but at the entire posterior distribution of $\boldsymbol{\beta}$. One can show (by completing the square) that the posterior distribution of $\boldsymbol{\beta}$ is

$$\boldsymbol{\beta}|y_1,\ldots,y_n \sim \mathsf{N}\left(\left(\mathbf{X}^\top\mathbf{X} + \frac{\sigma^2}{\tau^2}\mathbf{I}\right)^{-1}\mathbf{X}^\top\mathbf{y}, \left(\mathbf{X}^\top\mathbf{X} + \frac{\sigma^2}{\tau^2}\mathbf{I}\right)^{-1}\right)$$



(a) Samples from the prior distribution. (b) Data and samples from the posterior distribution.

Figure 5.1. Draws from the prior distribution and the posterior distribution of a Bayesian linear model. The bold line corresponds to the mean, the shaded area corresponds to pointwise 95% credible intervals.

[6] To incorporate an unknown variance σ^2 into the model we could use a normal-inverse gamma prior jointly placed on $(\boldsymbol{\beta}, \sigma^2)$.

Figure 5.1 illustrates this idea of Bayesian inference for a linear model with design matrix $\mathbf{X} = \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}$. Panel (a) shows ten draws from the prior distribution, whereas panel (b) shows draws from the posterior distribution given the data.

Suppose we are not interested in the regression coefficients $\boldsymbol{\beta}$, but only in predictions for the training data or unseen data. Thus we will try to re-express the model behind ridge regression without reference to the parameter $\boldsymbol{\beta}$. Essentially we have to combine (5.4) and (5.5) to find the marginal distribution of $\mathbf{y}$. The theory of the normal distribution tells us that the marginal distribution of $\mathbf{y}$ is also a normal distribution, so we only need to find its expected values and its variance.

$$\begin{aligned}
\mathbb{E}(\mathbf{y}) &= \mathbb{E}_{\boldsymbol{\beta}}\left(\mathbb{E}_{\mathbf{y}|\boldsymbol{\beta}}(\mathbf{y})\right) = \mathbb{E}_{\boldsymbol{\beta}}(\mathbf{X}\boldsymbol{\beta}) = \mathbf{X}\mathbb{E}_{\boldsymbol{\beta}}(\boldsymbol{\beta}) = \mathbf{0} \\
\mathbf{Var}(\mathbf{y}) &= \mathbf{Var}_{\boldsymbol{\beta}}\left(\mathbb{E}_{\mathbf{y}|\boldsymbol{\beta}}(\mathbf{y})\right) + \mathbb{E}_{\boldsymbol{\beta}}\left(\mathbf{Var}_{\mathbf{y}|\boldsymbol{\beta}}(\mathbf{y})\right) \\
&= \mathbf{Var}_{\boldsymbol{\beta}}(\mathbf{X}\boldsymbol{\beta}) + \mathbb{E}_{\boldsymbol{\beta}}\left(\sigma^2\mathbf{I}\right) = \mathbf{X}\mathbf{Var}_{\boldsymbol{\beta}}(\boldsymbol{\beta})\mathbf{X}^\top + \sigma^2\mathbf{I} \\
&= \tau^2\mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I}
\end{aligned}$$

thus ridge regression corresponds to assuming that

$$\mathbf{y} \sim \mathsf{N}\left(\mathbf{0}, \tau^2\mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I}\right).$$

What we have achieved by eliminating $\boldsymbol{\beta}$[7] is moving the linear model assumption of ridge regression from the mean into the covariance of the Gaussian distribution. This is the key idea which allows us to generalise the Bayesian linear model to Gaussian processes.

Recall that $\mathbf{X}\mathbf{X}^\top = \begin{pmatrix} \langle \mathbf{x}_1, \mathbf{x}_1 \rangle & \dots & \langle \mathbf{x}_1, \mathbf{x}_n \rangle \\ \vdots & \ddots & \vdots \\ \langle \mathbf{x}_n, \mathbf{x}_1 \rangle & \dots & \langle \mathbf{x}_n, \mathbf{x}_n \rangle \end{pmatrix}$, which suggests that we can resort to kernelisation again. This is what we will look at in the next section, however we will interpret the kernel matrix as a covariance matrix.

5.3.2 Gaussian processes

Basic idea In the Bayesian linear model we have placed a prior distribution on the regression coefficients and Bayesian inference yields a posterior distribution on the regression coefficients. The coefficients of a linear function fully determine a linear function (and vice versa), so we have done nothing other than having placed a prior distribution on all linear functions.

[7] To be mathematically more precise, we have integrated out $\boldsymbol{\beta}$.

Gaussian processes generalise this idea by placing a prior distribution on a much more general space of functions.

We start by defining what a Gaussian process actually is. We define a Gaussian process to be a collection of random variables $y_i = y(\mathbf{x}_i)$ $(i = 1, 2, 3, \ldots)$ depending on covariates $\mathbf{x}_i$ such that any *finite* subset of random variables $\mathbf{y} = (y_1, \ldots, y_n) = (y(\mathbf{x}_1), \ldots, y(\mathbf{x}_n))$ has a multivariate Normal distribution.

A Gaussian process is fully specified by the mean and the covariance of this Gaussian distribution. To keep things simple (and without loss of generality) we shall assume that the mean of this Gaussian distribution is always $\mathbf{0}$. With a bit of rewriting we have that $\mathbf{y}$ is from a Gaussian process if and only if

$$\mathbf{y} \sim \mathsf{N}(\mathbf{0}, \tau^2 \mathbf{K}(\mathbf{x}_1, \ldots, \mathbf{x}_n) + \sigma^2 \mathbf{I})$$

To keep the notation simple and consistent I have, without loss of generality, added $\sigma^2\mathbf{I}$ to the covariance and added the factor τ^2.

The Bayesian linear model is a special case of a Gaussian process, because for any set of $\mathbf{x}_1, \ldots, \mathbf{x}_n$ we have seen that $\mathbf{y} \sim \mathsf{N}\left(\mathbf{0}, \tau^2 \mathbf{X}\mathbf{X}^\top + \sigma^2 \mathbf{I}\right)$, thus $\mathbf{K}(\mathbf{x}_1, \ldots, \mathbf{x}_n) = \mathbf{X}\mathbf{X}^\top$. Other covariance functions (kernel functions) give nonlinear regression functions. Covariance functions have been covered in the APTS course on Spatial Statistics and are thus not discussed here.

5.3.3 Predictions from Gaussian processes

Conditionals of Gaussian distributions

Assume that

$$\begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix} \sim \mathsf{N}\left(\begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix} \right)$$

Then the conditional distribution of $\mathbf{y}_2$ given $\mathbf{y}_1$ is

$$\mathbf{y}_2 | \mathbf{y}_1 \sim \mathsf{N}\left(\boldsymbol{\mu}_2 + \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}(\mathbf{y}_1 - \boldsymbol{\mu}_1), \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}\right)$$

We can compute predictions for a new test case with covariates $\mathbf{x}_0$ by looking at the joint distribution

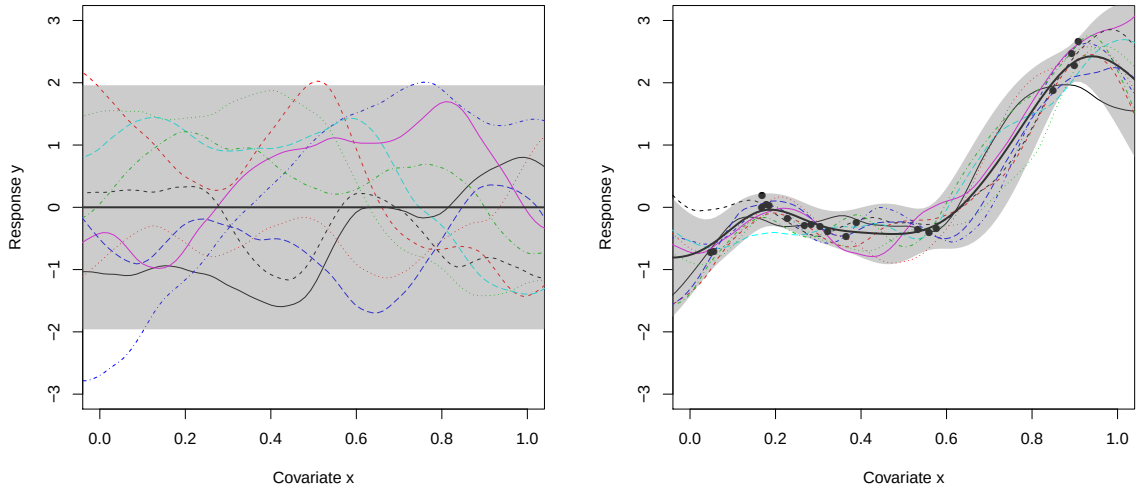
$$\begin{pmatrix} \mathbf{y} \\ y_0 \end{pmatrix} \sim \mathsf{N}\left(\begin{pmatrix} \mathbf{0} \\ 0 \end{pmatrix}, \begin{pmatrix} \tau^2\mathbf{K} + \sigma^2\mathbf{I} & \tau^2\mathbf{k}_0 \\ \tau^2\mathbf{k}_0^\top & \tau^2 k_{00} + \sigma^2 \end{pmatrix} \right),$$

where $\mathbf{K}$ is as defined is the preceding section, $\mathbf{k}_0 = (k(\mathbf{x}_0, \mathbf{x}_1), \ldots, k(\mathbf{x}_0, \mathbf{x}_n))$ is the covariance between the training data and the test case and $k_{00} = k(\mathbf{x}_0, \mathbf{x}_0)$. Then using the formula for the conditional distribution of a Gaussian we obtain

$$y_0|\mathbf{y} \sim \mathsf{N}\left(\mathbf{k}_0^\top \left(\mathbf{K} + \frac{\sigma^2}{\tau^2}\mathbf{I}\right)^{-1} \mathbf{y}, \tau^2 \left(k_{00} - \mathbf{k}_0^\top \left(\mathbf{K} + \frac{\sigma^2}{\tau^2}\mathbf{I}\right)^{-1} \mathbf{k}_0\right) + \sigma^2\right)$$

The mean of the posterior distribution of y_0 is nothing other than the point estimate of y_0 obtained from kernel ridge regression. The formula above gives the variance to be used for a prediction interval for a new observation. If we want to get the variance for a confidence interval for its mean we have to omit the σ^2 accounting for the error on the unseen data, i.e. the variance of the predicted mean is $\tau^2 \left(k_{00} - \mathbf{k}_0^\top \left(\mathbf{K} + \frac{\sigma^2}{\tau^2}\mathbf{I}\right)^{-1} \mathbf{k}_0\right)$.

Figure 5.2 shows five draws each from the prior distribution (panel (a)) and the posterior distribution (panel (b)) from a simple Gaussian process fitted to data.



(a) Samples from the prior distribution. (b) Samples from the posterior distribution.

Figure 5.2. Draws from the prior distribution and the posterior distribution of a simple Gaussian process (Matérn covariance with $\kappa = 2.5$). The bold line corresponds to the mean, the shaded area corresponds to pointwise 95% credible intervals.

5.3.4 Learning hyperparameters

Given that Gaussian processes are based on a proper probabilistic model can do not have to resort to cross-validation to find the hyperparameters. We can use an empirical Bayes strategy (sometime also referred to as maximum-likelihood) and maximise the posterior density with respect to the hyperparameters.

However, a Gaussian process can use many hyperparameters and there is often little information in the data about the hyperparameters. This is especially true for the parameter κ of the Matérn covariance function. Full Bayesian models thus typically fare better as they take into account the uncertainty about the values of the hyperparameters. However, with the possible exception of σ^2, none of the hyperparameters can be integrated out in closed form, thus one has to resort to either using a discrete grid or sampling techniques such as Markov Chain Monte Carlo (MCMC).

5.3.5 Gaussian processes in R

There are many packages in R (such as `mlegp`, `tgp` or `kernlab`) which implement Gaussian processes in R. We will use the package `tgp` which performs full Bayesian inference using MCMC. We will illustrate the functionality of this package using the motorcycle data.

```
# Load the required libraries
library(mlegp)
library(MASS)
# Fit a GP model (only squared exponential kernels, hyperparameters estimated by MLE)
model <- mlegp(mcycle$times, mcycle$accel)
# Compute predictions
predicted <- predict(model, se.fit=TRUE)
# Plot data
plot(mcycle)
# Add fitted line
lines(mcycle$times, predicted$fit)
# Add CI for mean function
lines(mcycle$times, predicted$fit+qnorm(0.975)*predicted$se.fit, lty=2, col=3)
lines(mcycle$times, predicted$fit-qnorm(0.975)*predicted$se.fit, lty=2, col=3)
# Add prediction interval
lines(mcycle$times, predicted$fit+qnorm(0.975)*(sqrt(model$nugget)+predicted$se.fit),
lines(mcycle$times, predicted$fit-qnorm(0.975)*(sqrt(model$nugget)+predicted$se.fit),
```

The `tgp` package performs full Bayesian inference. This takes into account the uncertainty in the hyperparameters, but does require using a sampling algorithm, which is much slower.

```
# Load the TGP package
library(tgp)
# Fit the model using MCMC
model <- bgp(mcycle$times, mcycle$accel)
# Plot the results
plot(model)
```

The `tgp` packages also implements a generalisation of Gaussian processes, called tree-based Gaussian processes, which implements "piecewise" GP's. These are actually better suited to the motorcycle data, as they can account for the heteroscedasticity of the data. However this would be beyond the scope of the course.

5.3.6 Classification using Gaussian processes and other extensions

So far we have only studied Gaussian processes for regression. Suppose the response variable of interest is z_i, which does not have a Gaussian distributions, i.e. z_i could be a -1/1 indicator, which would correspond to binary classification. We can handle this problem using Gaussian processes by assuming that there is an unobserved Gaussian process $(y_1, \ldots, y_n)$ and that we can only observe a process $z_i = z_i(y_i)$, which is defined

such that z_i only depends on $\mathbf{y}_i$[8]. In the case of binary classification $z_i \in \{-1, 1\}$ and we could assume a probit model, i.e. $\mathbb{P}\{z_i = 1|y_i\} = \Phi(y_i)$, where $\Phi(\cdot)$ is the Gaussian c.d.f.

This idea of latent Gaussian processes can be applied to many others settings.

5.4 Support Vector Machines

5.4.1 Introduction

The history of support vector machines reaches back to the 1960s. The "Generalised Portrait" algorithm, which constructs a separating hyperplane with maximal margin, was originally proposed by the Soviet mathematicians Vapnik and Chernovenkis. Over last decade, support vector machines have become an increasingly popular learning algorithm.

Though support vector machines are mostly used for classification, we will only cover support vector machines for regression.

5.4.2 Robust Statistics

You probably remember from your introductory undergraduate Statistics course that the median is more robust than the mean. In other words, the median is less affected by outliers than the mean. In this section we will relate this to loss functions and use these to propose robust methods for regression.

One can show that the mean $\bar{y} = \frac{1}{n}\sum_{i=1}^n y_i$ of a sample minimises the sum of squares, i.e.

$$\sum_{i=1}^n (y_i - a)^2$$

is minimal for $a = \bar{y}$. Similarly one can show than the median $\tilde{y}$ minimises the sum of absolute differences, i.e.

$$\sum_{i=1}^n |y_i - a|$$

is minimal for $a = \tilde{y}$.

Remember that in standard linear regression we choose the regression coefficients $\boldsymbol{\beta}$ such that

$$\sum_{i=1}^n (y_i - \mathbf{x}_i^\top \boldsymbol{\beta})^2,$$

is minimal, i.e. (standard) linear regression is using a quadratic loss function (just like the mean). One can obtain a robust version of linear regression by choosing $\boldsymbol{\beta}$ such that

[8] More precisely, this corresponds to assuming that the z_i are conditionally independent given the y_i.

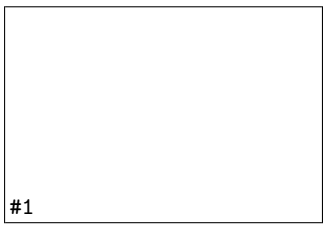
$$\sum_{i=1}^{n} |y_i - \mathbf{x}_i^\top \boldsymbol{\beta}|,$$

is minimal. This robust approach to regression yields an algorithm that can cope much better with outliers. However computing the regression coefficients is computationally more demanding and there is no "nice" theory for tests and confidence / prediction intervals.

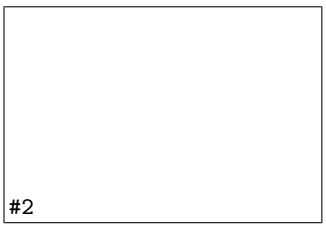
A compromise between two loss functions is Huber's loss function which is defined as

$$\sum_{i=1}^{n} L_\delta^H(y_i - \mathbf{x}_i^\top \boldsymbol{\beta}) \qquad \text{where} \qquad L_\delta^H(z) = \begin{cases} \frac{z^2}{2} & \text{for } -\delta \le z \le \delta \\ \delta(|z| - \delta/2) & \text{otherwise,} \end{cases}$$

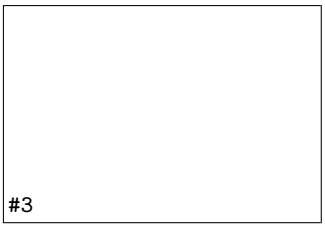
where $\delta > 0$ is a suitably chosen constant. Huber's loss function is implemented in the function `rlm` in `MASS`. Figure 5.3 (a) to (c) compares the three loss functions.



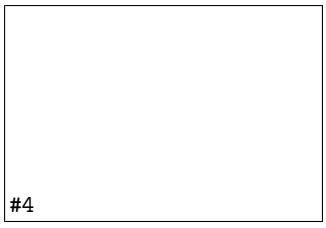
(a) Quadratic loss



(b) Linear loss



(c) Huber's loss



(d) ε-insensitive loss

Figure 5.3. Different loss functions for regression.

Support Vector Regression

Support vector regression is yet another way of performing robust regression. All methods described in the previous section yield estimates of the regression coefficients which depend on all observations. In order to obtain a sparse solution which depends only on a small subset of the observations a modification of the above loss functions is used. This new "ε-insensitive" loss function is defined as

$$L_\varepsilon(z) = (|z| - \varepsilon)_+ = \begin{cases} 0 & \text{for } -\varepsilon \le z \le \varepsilon \\ |z - \varepsilon| & \text{otherwise,} \end{cases}$$

where ε is a suitably chosen constant. Small errors (i.e. errors less than ε) will not incur any loss with this new loss function, thus ε is typically chosen rather small (often as small as 10^{-3}).

As common in the support vector literature we will denote the regression coefficients by $\mathbf{w}$, rather than $\boldsymbol{\beta}$. In linear support vector regression we fit a linear function $b+\langle \mathbf{x}_i, \mathbf{w}\rangle$ to a response y_i by minimising the criterion

$$\underbrace{\frac{1}{2}\|\mathbf{w}\|^2}_{\text{regularisation}} + \underbrace{C\sum_{i=1}^{n} L_\varepsilon\left(y_i - b - \langle \mathbf{x}_i, \mathbf{w}\rangle\right)}_{\text{training loss}}$$

Note that the objective function is almost the same as in ridge regression. The only difference is that we use the ε-insensitive loss function for the training loss rather than the quadratic loss used in ridge regression.

Before we go into the details of solving the optimisation problem we will first generalise the problem to the nonlinear case. Suppose we want to use a feature map $\boldsymbol{\phi}(\cdot)$, i.e. fit the function $b + \langle \boldsymbol{\phi}(\mathbf{x}_i), \mathbf{w}\rangle$. In this case the objective function becomes

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} L_\varepsilon\left(y_i - b - \langle \boldsymbol{\phi}(\mathbf{x}_i), \mathbf{w}\rangle\right)$$

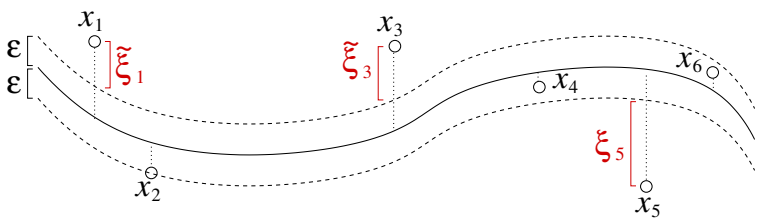


Figure 5.4. Slack variables $\tilde{\xi}_i$ and ξ_i used in support vector regression

The above optimisation can be written as an optimisation problem using slack variables $\tilde{\xi}_i$ and ξ_i. We want to minimise

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}(\tilde{\xi}_i + \xi_i)$$

subject to $y_i - (\langle \boldsymbol{\phi}(\mathbf{x}_i), \mathbf{w}\rangle + b) \leq \varepsilon + \tilde{\xi}_i$ and $(\langle \boldsymbol{\phi}(\mathbf{x}_i), \mathbf{w}\rangle + b) - y_i \leq \varepsilon + \xi_i$ with slack variables $\tilde{\xi}_i, \xi_i \geq 0$. This is illustrated in figure 5.4. The corresponding dual is

$$D(\tilde{\boldsymbol{\alpha}}, \boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i,j}(\tilde{\alpha}_i - \alpha_i)(\tilde{\alpha}_j - \alpha_j)\left\langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j)\right\rangle - \varepsilon\sum_{i=1}^{n}(\tilde{\alpha}_i + \alpha_i) + \sum_{i=1}^{n} y_i(\tilde{\alpha}_i - \alpha_i),$$

which is to be maximised over $\tilde{\alpha}_i, \alpha_i \in [0, C]$ with $\sum_{i=1}^{n}(\tilde{\alpha}_i - \alpha_i) = 0$ and $\tilde{\alpha}_i\alpha_i = 0$. The estimated regression curve can be expressed as a function of $(\tilde{\alpha}_i - \alpha_i)$ and b:

$$\hat{m}(\mathbf{x}_0) = \sum_{i=1}^{r}(\tilde{\alpha}_i - \alpha_i)\left\langle \boldsymbol{\phi}(\mathbf{x}_0), \boldsymbol{\phi}(\mathbf{x}_i)\right\rangle + b$$

Once again the optimisation problem and its solution only depend on a subset of the learning dataset (those vectors having $\tilde{\alpha}_i > 0$ or $\alpha_i > 0$) and only through inner

products, i.e. we can use a kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j)) \rangle$. In this case the estimated regression curve becomes

$$\hat{m}(\mathbf{x}_0) = \sum_{i=1}^{r} (\tilde{\alpha}_i - \alpha_i) k(\mathbf{x}_0, \mathbf{x}_i) + b,$$

which is again of the optimal form suggested by the representer theorem.

Example 5.1. We will now use a support vector machine to fit a smooth function to the motorcycle data. To illustrate its robustness we will set the 111-th observation to a unrealistically large value.

```
# Get hold of the data
library(MASS)
data(mcycle)
# Put in outlier
mcycle$accel[111] <- 500
# Fit the SVM
# (We should have used tune.svm to get good values for the tuning parameters)
my.svm <- svm(accel~times, data=mcycle, type="eps-regression", kernel="radial",
              cost=1, gamma=10, eps=0.01)
# Plot the data
plot(mcycle)
# Plot the fitted SVM
lines(mcycle$times, predict(my.svm, mcycle))
```

◁

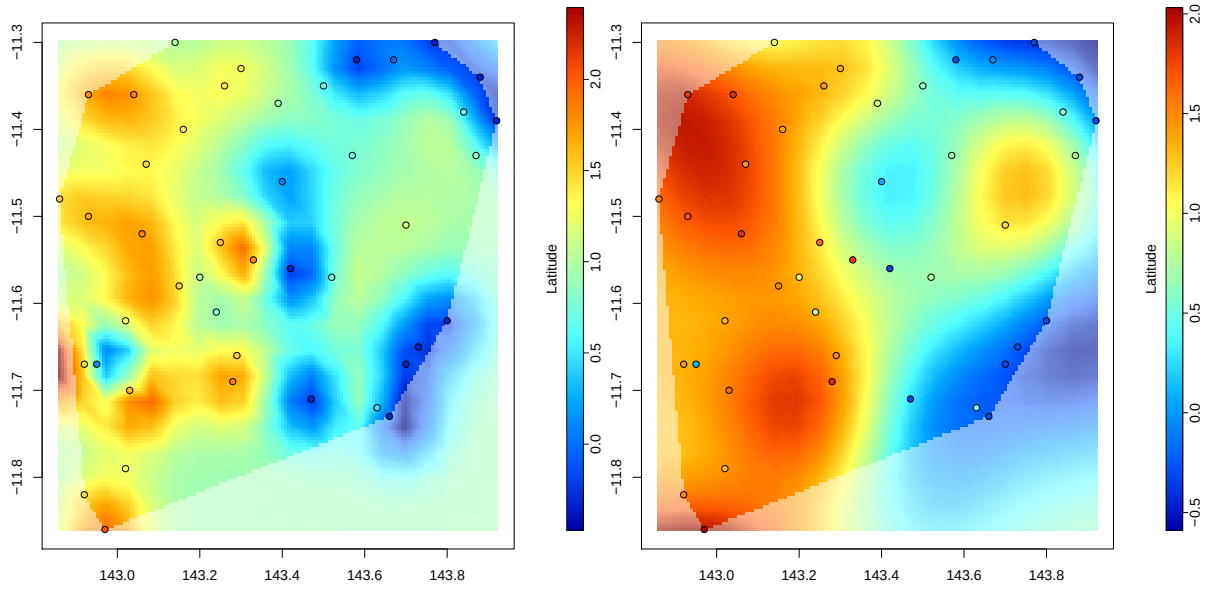
Example 5.2 (Great Barrier Reef (continued)). Figure 5.5 shows support-vector-regression fits to the Great Barrier Reef data for different values of the hyperparameters. ◁

5.4.3 Practical considerations

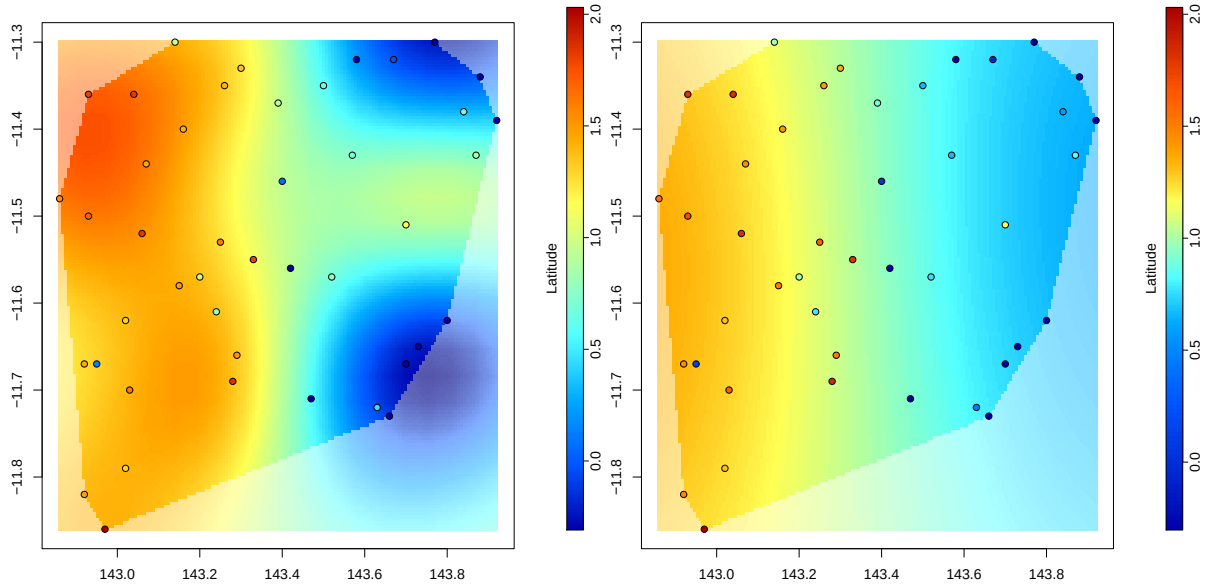
The performance of a support vector machines depends crucially on the hyperparameters chosen: Big values of C generally yield an overfit to the data. The bigger the degree q of the polynomial or the smaller the "width" γ of the Gaussian kernel is selected, the more wigglier the fitted function will be and the more likely the fit will result in an overfit. The hyperparameters are typically tuned using either a validation set or cross-validation.

Support vector machines are not scale-invariant, so it is necessary to scale the data beforehand. However most implementations of SVM (like the one used in `R`) perform the standardisation automatically.

Support vector machines have been successfully trained with huge amounts of data. This applies to the number of observations as well as to the number of covariates.



(a) $C = 0.1$, $\gamma = 0.1$ (b) $C = 1$, $\gamma = 1$



(c) $C = 10$, $\gamma = 1$ (d) $C = 10$, $\gamma = 10$

Figure 5.5. Suppport vector machine fits to the Great Barrier Reef data for different values of the hyperparameters.

Support vector machines do not make any model assumption. This makes them a very versatile tool, but it makes assessing the uncertainty difficult to impossible: we cannot define proper confidence intervals or compute criteria like the AIC or the BIC. There are some probabilistic upper bounds like the Vapnik-Chernovenkis bound, but these bounds are typically very loose and thus only of limited use.

5.4.4 SVMs and Gaussian processes compared

Support vector regression machines and Gaussian processes perform very similar tasks, however they have different strengths and weaknesses.

+ Gaussian processes are based on proper probabilistic model, so hyperparameters can be learnt using standard statistical techniques such as maximum likelihood (or "Empirical Bayes") or full Bayesian inference. There is little need to resort to ad-hoc techniques such as cross-validation.
  Gaussian processes also give confidence/credible intervals.
- SVMs are much faster, as they only rely on the support vectors. Gaussian processes require the inversion of a large covariance matrix, which can be very slow. However there are many approximation techniques which speed up computations for Gaussian processes a lot.
- SVMs are also more robust: they are based on the robust ε-insensitive loss function, whereas Gaussian processes are based on the quadratic loss functions. However, Gaussian processes can be robustified by assuming a more complex model for variance: each observation in this model is assumed to have its own variance, which is drawn from an inverse χ^2-distribution, which corresponds to assuming t-distributed, rather than Gaussian noise, which is much more robust.

5.4.5 ν-SVMs

So far we have stated all objective functions in terms of a cost C which is used to scale the training loss. Interpreting C and choosing an appropriate value for C is however difficult. There exists an alternative formulation of support vector machines, called the ν-SVM, which introduces a parameter ν instead of the cost C. ν which takes values in the interval $(0, 1)$ has an easier interpretation.

– The proportion of support vectors will be at least ν.
– The proportion of observations lying on the "wrong" side of the margin is at most ν.

The optimisation problem (and theory) behind ν-SVMs however is more difficult. For this reason we have only studied C-SVMs so far. It is also worth noting that the opti-

misation problem is not always feasible for low values of ν, unless the kernel is positive definite (like for example the Gaussian kernel).

The R package `e1071` also implements ν-SVMs

6

Case studies

The final lecture will give some examples of the use of smoothing in addressing modelling issues which arise in real applications. These are likely to include:

– the patterns of SO_2 pollution over Europe;
– the dispersion of pollution in groundwater;
– the changes in pollution over a river network;
– the modelling of human facial shape.

References

Bowman, A. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis*. Oxford: Oxford University Press.

Fan, J. and Gijbels, I. (1996). *Local polynomial modelling and its applications*. London: Chapman and Hall.

Green, P. and Silverman, B. (1994). *Nonparametric Regression and Generalized Linear Models*. Chapman and Hall.

Hastie, T. and Tibshirani, R. (1986). Generalized additive models (with discussion). *Statistical Science 1*, 297–318.

Hastie, T. and Tibshirani, R. (1990). *Generalized Additive Models*. London: Chapman and Hall.

Johnson, N. and Kotz, S. (1972). *Distributions in Statistics: Continuous Univariate Distributions*, Volume 2. New York: Wiley.

Ruppert, D., Wand, M. P., and Carroll, R. (2003). *Semiparametric regression*. London: Cambridge University Press.

Scott, D. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization*. New York: John Wiley.

Silverman, B. (1986). *Density estimation for statistics and data analysis*. London: Chapman and Halland Hall.

Simonoff, J. S. (1996). *Smoothing methods in statistics*. New York: Springer.

Wand, M. P. and Jones, M. C. (1995). *Kernel smoothing*. London: Chapman and Hall.

Wood, S. (2006). *Generalized Additive Models: an introduction with R*. London: Chapman and Hall/CRC.