MATH5824 Generalised Linear and Additive Models

Robert G Aykroyd

2024-04-22

Table of contents

Weekly schedule

Items will be added here week-by-week and so keep checking when you need up-to-date information on what you should be doing. Note that items specific to *MATH5824M* will be marked accordingly, otherwise items refer to the material common with *MATH3823*.

i Week 9 (22 - 26 April)

- Submit your Computer Practical report on-time!
- **Before next Lecture:** Re-read *Chapter 6: Log-linear Models*.
- **Lecture on Tuesday:** Start *Chapter 7: Extensions to Loglinear models*.
- **Lecture on Thursday:** Continue *Chapter 7: Extensions to Loglinear models*.
- **Lecture on Friday:** Cover *Chapter 6: General Additive Models*.
- **Weekly feedback:** Check any previous Exercises and solutions not completed.

! Advanced notice

Please note that I will not be available after Friday 22 March until Monday 22 April.

⚠ Assessed Practical

- **Module Assessment:** Set on 12 March with submission deadline 23 April (that is after the break). You will be expected to write a short report based on an RStudio practical.
- **Computer classes:** Supervised session on 19/20 March – check your timetable.
- **Generative AI usage within this module:** The assessments for this module fall in the red category for using Generative AI which means you must not use Generative AI tools. The purpose and format of the assessments makes it inappropriate or impractical for AI tools to be used.

i Week 8 (18 - 22 March)

- **Before next Lecture:** Re-read Chapters 4 and 5 ready for practical.
- **Lecture on Tuesday:** Start *Chapter 6: Loglinear Modelling* with *Sections 6.1-6.3*.
- **Computer Practical on Tuesday/Wednesday:** Work on Assessed Practical.
- **Lecture on Thursday:** Complete *Chapter 6: Loglinear Modelling* with *Section*

6.4-6.5.

- **Lecture on Friday:** Complete *Chapter 5: Choosing the smoothing parameter* with *Sections: 5.4-5.6.*
- **Weekly feedback:** Start Exercises for Chapter 6 and check answers with solutions.

Week 7 (11 - 15 March)

- **Before next Lecture:** Re-read *Chapter 5: Sections 5.1-5.3.*
- **Lecture on Tuesday:** Complete *Chapter 5:* by looking at *Section 5.4: Odds Ratio* and *Section 5.5 - Application to dose-response experiments.*
- **Lecture on Thursday:** Consider selected Exercises from previous Chapters and discuss Assessed Practical. If time start *Chapter 6: Log-linear Modelling.*
- **Before next Lecture:** Please re-read *MATH5824 Chapter 4: Smoothing splines.*
- **Lecture on Friday:** Start *Chapter 5: Choosing the smoothing parameter* with *Sections: 5.1-5.3.*
- **Weekly feedback:** Complete Exercises for Chapter 5.

Week 6 (4 - 8 March)

- **Before next Lecture:** Re-read *Chapter 4: Sections 4.1-4.2.*
- **Lecture on Tuesday:** Complete *Chapter 4* with *Sections: 4.3 Model deviance, 4.4 Model Residuals & 4.5 Fitting GLMs in R*
- **Lecture on Thursday:** Start *Chapter 5: Modelling Proportions* with *Sections 5.1 & 5.2.*
- **Before next Lecture:** Please re-read *MATH5824 Section 4.1: Overview* and *Section 4.2: The penalized least-squares criterion.*
- **Lecture on Friday:** Complete *Chapter 4* with *Sections 4.3-4.5.*
- **Weekly feedback:** Complete *Chapter 4 Exercises.*

Week 5 (26 February - 1 March)

- **Before next Lecture:** Re-read all of *Chapter 3.*
- **Computer Practical:** Either on Tuesday or Wednesday, attend supervised computer class – see your timetable. For information see *Week 5 Computer Practical* folder on Minerva.
- **Lecture on Tuesday:** Start *Chapter 4* by covering *Section 4.1: The identically distributed case.*
- **Lecture on Thursday:** *Chapter 4: Section 4.2: The general case.*
- **Lecture on Friday:** Complete *Chapter 4* with *Section 4.1: Overview*, then cover *Section 3.4: Roughness penalties* which was forgotten in the previous lecture. Then, continue with *Section 4.2: The penalized least-squares criterion.*
- **Weekly feedback:** Check your answers on Chapter 3 Exercises with online

solutions and start *MATH5824M Chapter 3 Exercises*.

i Week 4 (19 - 23 February)

- **Before next Lecture:** Be confident with all material up to, and including, *Section 3.2: The GLM structure*.
- **Lecture on Tuesday:** We will cover *Section 3.3: The random part of a GLM*, *Section 3.4: Moments of exponential-family distributions* and, if time permits, *Sections 3.5: The systematic part of the model*.
- **Lecture on Thursday:** Complete Chapter 3 by covering *Section 3.6: The link function*. Then, we will consider selected Exercises from *Section 3.7*.
- **Before next Lecture:** Please re-read the whole of *MATH5824M Chapter 2: Introducing Splines*.
- **Lecture on Friday:** We will start *MATH5824M Chapter 3: Interpolating Splines* by looking at *Sections 3.1-3.3*, *Overview*, *Natural splines* and *Properties of natural splines*. [Edit: Also, *Section 3.4: Fitting interpolating splines in R*.]
- **Weekly feedback:** Complete the first Chapter 3 Quiz and start the *MATH5924M Exercises* in *Section 3.6* and the *MATH3823 Exercises* in *Section 3.7*.

i Week 3 (12 - 16 February)

- **Before next Lecture:** Please re-read *Section 2.5: Model shorthand notation* and *Section 2.6 Fitting linear models in R*, and read *Section 2.7: Ethics in statistics and data science*.
- **Lecture on Tuesday:** We will start Chapter 3 with *Section 3.1: Motivating examples* and *Section 3.2: The GLM structure*.
- **Before next Lecture:** Please re-read *Sections 3.1* and *3.2* carefully.
- **Lecture on Thursday:** CANCELLED due to illness.
- **Lecture on Friday:** CANCELLED due to illness.
- **Weekly feedback:** Complete the *MATH5824M Exercises** in Chapters 1 & 2.

i Week 2 (5 - 9 February)

- **Before next Lecture:** Please re-read *Section 2.1: Overview* and *Section 2.2: Linear models*, and self-study *Section 2.3: Types of normal linear model*.
- **Lecture on Tuesday:** We will cover *Section 2.4: Matrix representation of linear models* and briefly *Section 2.5: Model shorthand notation*.
- **Before next Lecture:** Please re-read *Sections 2.4* and *2.5* carefully.
- **Lecture on Thursday:** We will cover *Section 2.6: Fitting linear models in R* then discuss selected Exercises from Chapters 1 and 2.
- **Before next Lecture:** Please re-read the whole of *MATH5824M Chapter 1:*

Non-parametric Modelling.

- **Lecture on Friday:** We will cover the whole of *MATH5824M Chapter 2: Introducing Splines*.
- **Weekly feedback:** Complete the Chapter 2 Quizzes and complete the Exercises in *Section 2.8*. Also, complete the *MATH5824M Section: 1.4 Exercises*.

Week 1 (29 January - 2 February)

- **Before first Lecture:** Please read the *Overview*.
- **Lecture on Tuesday:** We will briefly cover all material in *Chapter: Introduction*.
- **Before next Lecture:** Please re-read *Chapter 1* carefully, especially any sections not covered in Lectures.
- **Lecture on Thursday:** Start *Chapter 2: Essentials of Normal Linear Models* with *Section 2.1: Overview* & *Section 2.2: Linear models*.
- **Before next Lecture:** Please read the *MATH5824M Overview*.
- **Lecture on Friday:** We will cover the whole of *MATH5824M Chapter 1: Non-parametric Modelling*.
- **Weekly feedback:** Complete the Chapter 1 Quizzes and self-study the Exercises in *Section 1.5* – solutions to be added during Week 1. If you have time, then also self-study the *MATH5824M Exercises* in *Section 1.4*.

Advanced notice

- **Module Assessment:** Set on 14 March with submission deadline 23 April (that is after the break). You will be expected to write a short report based on an RStudio practical.
- **Computer classes:** 27/28 February for Practice and 19/20 March for Assessment – check your timetable.
- **Generative AI usage within this module:** The assessments for this module fall in the red category for using Generative AI which means you must not use Generative AI tools. The purpose and format of the assessments makes it inappropriate or impractical for AI tools to be used.

Provisional Weekly Lecture Schedule

Week 1	Chapter 1	All
Week 2	Chapter 2	All
Week 3	Chapter 3	Sections 3.1-3.3
Week 4		Sections 3.4-3.5
Week 5	Chapter 4	Sections 4.1-4.3
Week 6		Sections 4.4-4.5

Week 7	Chapter 5	Sections 5.1-5.3
Week 8		Sections 5.4-5.6
Easter		
Week 9	Chapter 6	All
Week 10	Exercises	All Exercises
Week 11	Revision	

Overview

Preface

These lecture notes are produced for the University of Leeds module "MATH5824 - Generalized Linear and Additive Models" for the academic year 2023-24. They are based on the lecture notes used previously for this module and I am grateful to previous module lecturers for their considerable effort: Lanpeng Ji, Amanda Minter, John Kent, Wally Gilks, and Stuart Barber. This year, again, I am using Quarto (a successor to RMarkdown) from RStudio to produce both the html and PDF, and then GitHub to create the website which can be accessed at rgaykroyd.github.io/MATH5824/. Please note that the PDF versions will only be made available on the University of Leeds Minerva system. Although I am a long-term user of RStudio, I am a novice at Quarto/RMarkdown and a complete beginner using Github and hence please be patient if there are hitches along the way.

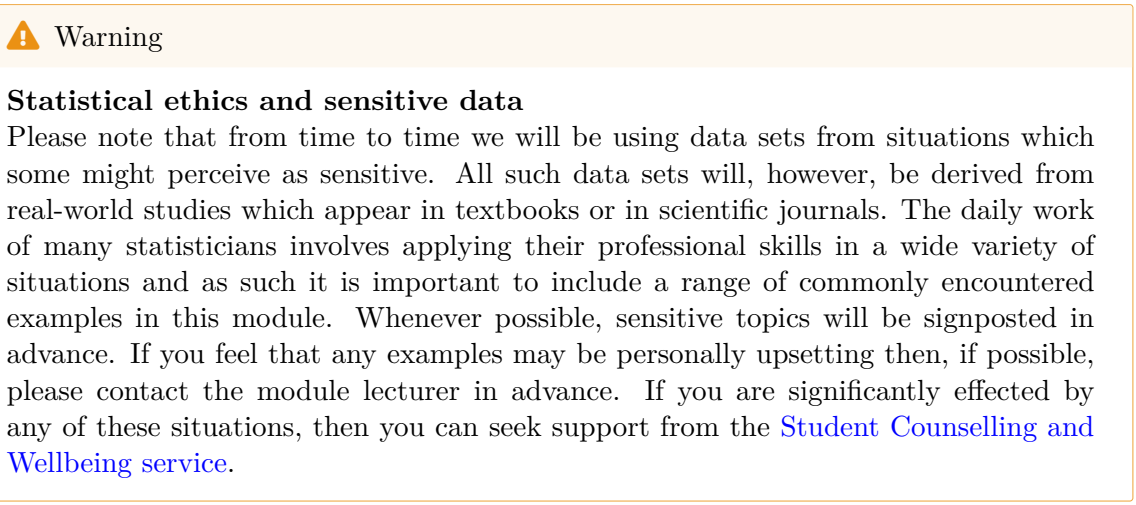
RG Aykroyd, Leeds, January 22, 2024

Changes since last year

Feedback from the students last year was very positive, but there were consistent comments regarding two issues: (1) a shortage of practice exercises and the opportunity to discuss these in class, and (2) limited RStudio support in preparation for the assessment. For the first of these, additional exercises have been prepared and are included in the learning material. Also, I am trying some short quizzes so that you can check your basic knowledge. Further, I intend to set-aside some lecture time for us to discuss selected exercises. For the second, an additional computer session has been added, in Week 5 (26 February - 1 March), this is 3 weeks before the assessed practice in Week 8 (18 - 22 March). Further, a few new instructional videos will be available addressing some RStudio topics. Together, these represents a considerable about of extra work for me, but I hope that they are helpful and so please give your feedback whenever there is an opportunity.

Generative AI usage within this module

The assessments for this module fall in the red category for using Generative AI which means you must not use Generative AI tools. The purpose and format of the assessments makes it inappropriate or impractical for AI tools to be used.



Official Module Description

Module summary

Linear regression is a tremendously useful statistical technique but is limited to normally distributed responses. Generalised linear models extend linear regression in many ways - allowing us to analyse more complex data sets. In this module we will see how to combine continuous and categorical predictors, analyse binomial response data and model count data.A further extension is the generalised additive model. Here, we no longer insist on the predictor variables affecting the response via a linear function of the predictors, but allow the response to depend on a more general smooth function of the predictor.

Objectives

On completion of this module, students should be able to:

- carry out regression analysis with generalised linear models including the use of link functions, deviance and overdispersion;
- fit and interpret the special cases of log linear models and logistic regression;
- compare a number of methods for scatterpot smoothing suitable for use in a generalised additive model;
- use a backfitting algorithm to estimate the parameters of a generalised additive model;
- interpret a fitted generalised additive model;
- use a statistical package with real data to fit these models to data and to write a report giving and interpreting the results.

Syllabus

Generalised linear model; probit model; logistic regression; log linear models; scatterplot smoothers; generalised additive model.

University Module Catalogue

For any further details, please see MATH5824 Module Catalogue page

1 Non-parametric Modelling

1.1 Introduction

Here is a short video [3 mins] to introduce the chapter.

In the Level 3 component of this module, we extend the simple linear regression model to the generalised linear model which can cope with non-normally distributed response variables, in particular data following binomial and Poisson distributions. However, we still just use linear functions of the predictor variables. A further extension of the linear model is the generalised additive model. Here, we no longer insist on the predictor variables affecting the response via a linear function of the predictors, but allow the response to depend on a more general smooth function of the predictor. In the Level 5 component of this module, we study splines and their use in interpolating and smoothing the effects of explanatory variables in the generalised linear models of the Level 3 component of this module (see separate Lecture Notes accompanying MATH3823). Towards the end of the material, we will learn that the fitting of generalised additive models is a straightforward extension of what is learnt in MATH3823.

Outline of the additional material in MATH5824 compared to MATH3823:

1. Interpolating and smoothing splines.
2. Cross-validation and fitting splines to data.
3. The generalised additive model.

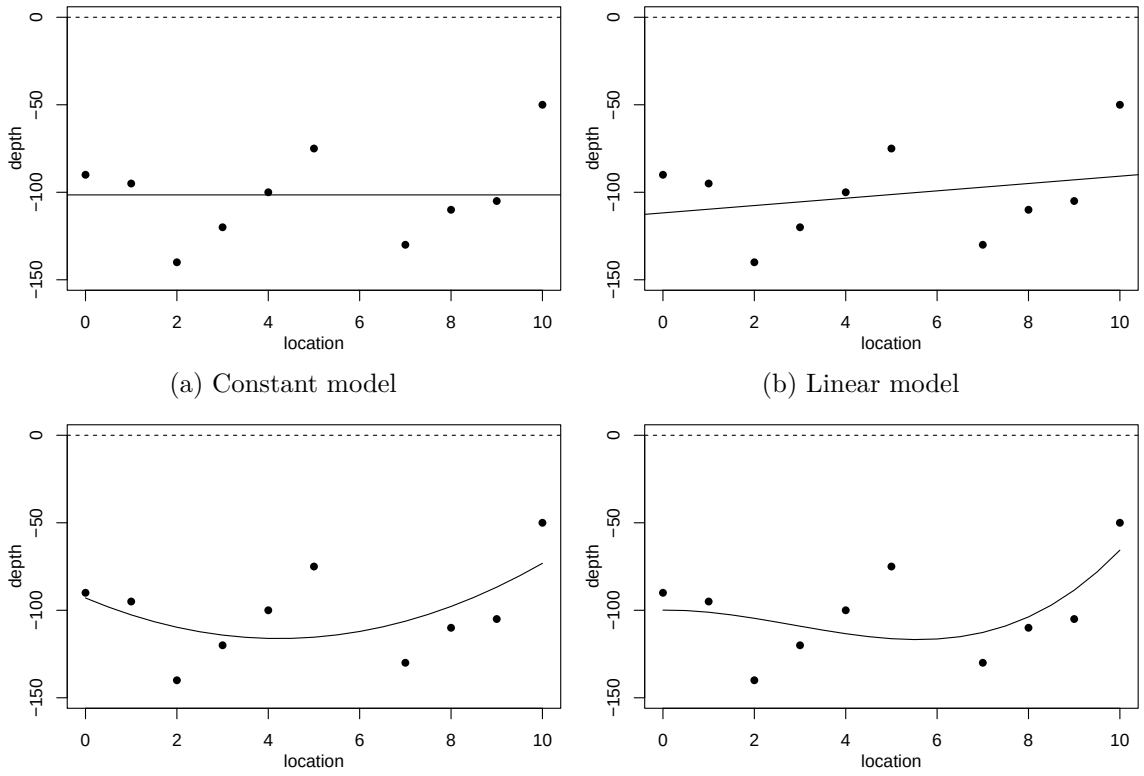
1.2 Motivation

Table 1.1 reports on the depth of a coal seam determined by drilling bore holes at regular intervals along a line. The depth y at location $x = 6$ is missing: could we estimate it?

Table 1.1: Coal-seam depths (in metres) below the land surface at intervals of 1 km along a linear transect.

Location, x	0	1	2	3	4	5	6	7	8	9	10
Depth, y	-90	-95	-140	-120	-100	-75	NA	-130	-110	-105	-50

Figure 1.1 plots these data, superimposed with predictions from several polynomial regression models.



(a) Constant model

(b) Linear model

(c) Quadratic model

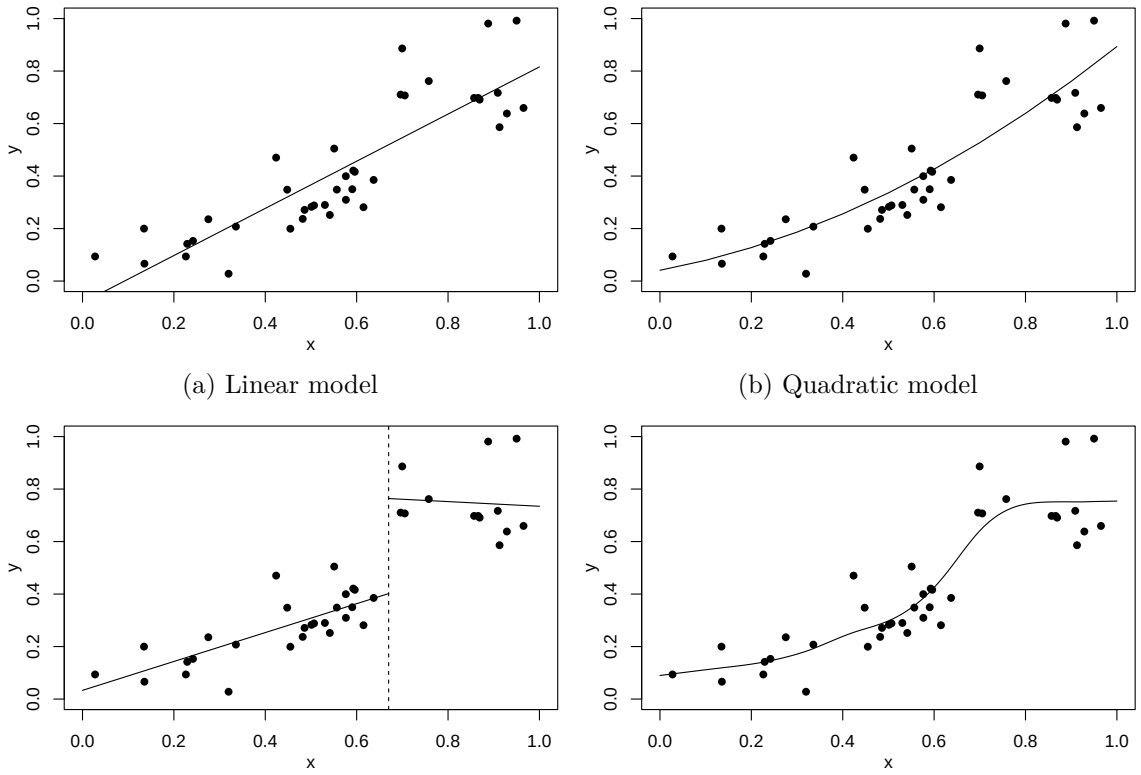
(d) Cubic model

Figure 1.1: The coal-seam data superimposed with predictions from polynomial regression models.

Each of these models would predict a different value for the missing observation y_6. We do not know the accuracy of the depth measurements, so in principle any of these curves could be correct. Clearly, the residual variance is largest for the constant-depth model in Figure 1.1a, and smallest for the cubic polynomial in Figure 1.1c – the residual sums of squares are: 6252.5, 5773.05, 4489.84, 4242.89. However, none of these models produces a convincingly good fit. Moreover, these models are not particularly believable, since we know that geological pressures exerted over very long periods of time cause the landscape and its underlying layers of rock to undulate and fracture. This suggests we need a different strategy.

Next, consider the simulated example in Figure 1.2. At first look we might be happy with the fitted curves in Figure 1.2a or Figure 1.2b. The data, however, are created with a *change-point* at $x = 0.67$ where the relationship changes from linear with slope 0.6 to a constant value of 0.75. This description is completely lost with these two models.

Figure 1.2c shows the result of fitting one linear function to the data below 0.67 and a second linear function above. Clearly, this fits well but it has assumed that the change-



(a) Linear model

(b) Quadratic model

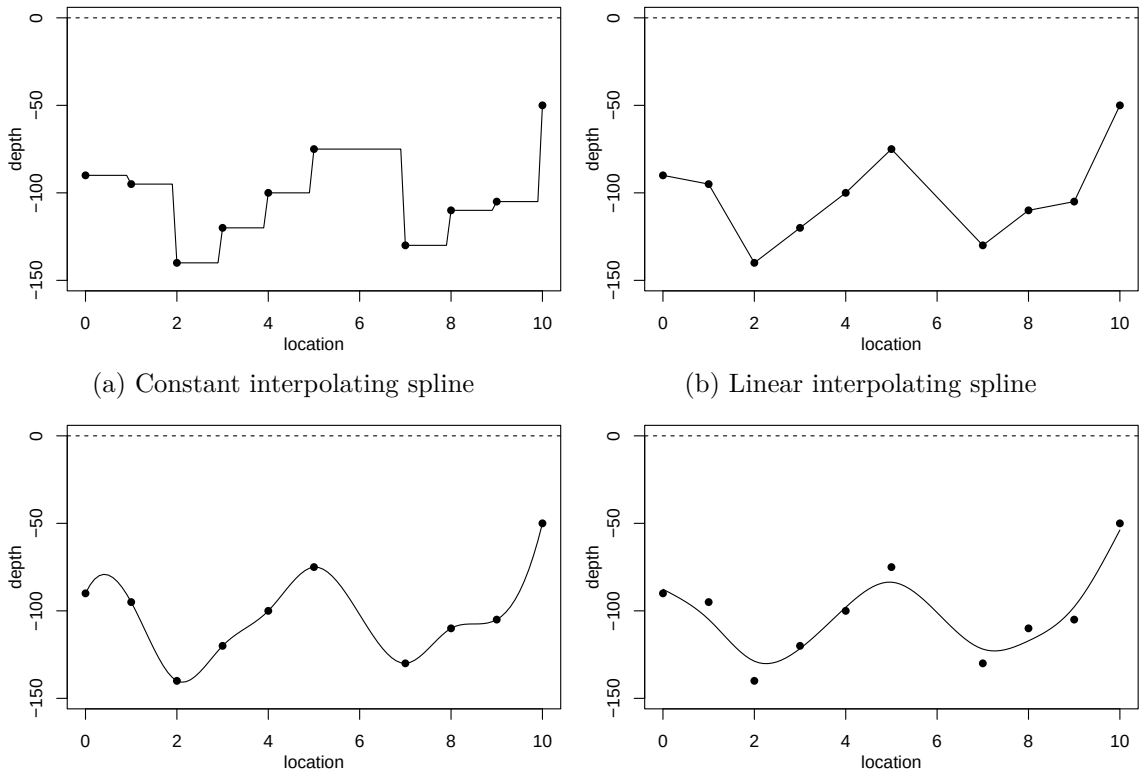
(c) Piecewise linear model

(d) Cubic smoothing spline

Figure 1.2: Simulated data superimposed with predictions from various models.

point location is known – which is unrealistic. Finally, Figure 1.2d shows a fitted *cubic smoothing spline* to the data – we will studies these models later. This shows an excellent fit and leads to appropriate conclusions. That is, the relationship is approximately linear for small values, then there is a rapid increase, and finally a near constant value for high values. Of course, this is not exactly as the true relationship with a discontinuity at $x = 0.67$ but it would definitely suggest something extreme occurs between about 0.6 to 0.7. Full details will follow later, but the cubic spline fits local cubic polynomials which are constrained to create a continuous curve.

Now returning to the coal seam data. Figure 1.3 shows the data again, superimposed with predictions from methods which are not constrained to produce such smooth curves.



(a) Constant interpolating spline

(b) Linear interpolating spline

(c) Cubic interpolating spline

(d) Cubic smoothing spline

Figure 1.3: The coal-seam data superimposed with predictions from various spline models.

The simplest method, *constant-spline interpolation*, assumes that the dependent variable remains constant between successive observations, with the result shown in Figure 1.3a. However, the discontinuities in this model make it quite unreliable. A better method, whose results are shown in Figure 1.3b, is *linear-spline interpolation*, which fits a straight line between successive observations. Even so, this method produces discontinuities in the *gradient* at each data point. A better method still, shown in Figure 1.3c, is *cubic spline interpolation*, which fits a cubic polynomial between successive data points such that both the gradient and the curvature at each data point is continuous.

A feature of all these interpolation methods is that they fit the data exactly. Is this a good thing? The final method assumes that there may be some measurement error in the observations, which justifies fitting a smoother cubic spline than the cubic interpolating spline, but as we see in Figure 1.3d which does not reproduce the data points exactly. Is this a bad thing? We will see during this module how to construct and evaluate these curves. Here, the results are presented only for motivation.

Focus on polynomials quiz

Test your knowledge recall and comprehension to reinforce basic ideas about continuity and differentiability.

1. Which of the predicted curves in Figure 1.3 are continous?

- (A) None of the models
- (B) Model (a) only
- (C) Models (c) and (d) only
- (D) All models

2. Which of the predicted curves in Figure 1.3 have a continous first derivative?

- (A) None of the models
- (B) Model (b) only
- (C) Models (c) and (d) only
- (D) All models

3. Which of the predicted curves in Figure 1.3 have a continous second derivative?

- (A) None of the models
- (B) Model (b) only
- (C) Models (c) and (d) only
- (D) All models

4. Which of the predicted curves in Figure 1.3 have a continous third derivative?

- (A) None of the models
- (B) Model (b) only
- (C) Models (c) and (d) only
- (D) All models

4. Which of the predicted curves in Figure 1.3 has the highest residual sum of squares?

- (A) None of the models
- (B) Model (a)
- (C) Model (b)
- (D) Model (c)
- (E) Model (d)
- (F) All model

1.3 General modelling approaches

We wish to model the dependence of a response variable y on an explanatory variable x, where y and x are both continuous. We observe y_i at each time x_i, for $i = 1, \ldots, n$, where the observation locations are ordered: $x_1 < x_2 < \ldots < x_n$. We imagine that the y's are noisy versions of a smooth function of x, say $f(x)$. That is,

$$y_i = f(x_i) + \epsilon_i, \tag{1.1}$$

where the $\{\epsilon_i\}$ are i.i.d:

$$\epsilon_i \sim \mathrm{N}(0, \sigma^2). \tag{1.2}$$

We suppose we do not know the correct form of function f: how can we estimate it?

It is useful to divide modelling approaches into two broad types: parametric and non-parametric.

Parametric models

By far the most common parametric model is simple linear regression, for example, $f(x) = \alpha + \beta x$, where parameters α and β are to be estimated. This is, of course, the simplest example of the polynomial model family, $f(x) = \alpha + \beta x + \gamma x^2 + \cdots + \omega\ x^p$, where p is the *order* of the polynomial and where all of $\alpha, \beta, \gamma, \ldots, \omega$ are to be estimated. This has as special cases: quadratic, cubic, quartic, and quintic polynomials models. Also common are exponential models, for example $f(x) = \alpha e^{-\beta x}$, where α, β are to be estimated – do not confuse this with the exponential probability density function.

Note that the polynomial models are all linear functions *of the parameters*. They are standard forms in regression modelling, as studied in MATH3714 (Linear regression and Robustness) and MATH3823 (Generalised linear models). The exponential model, however, is an example of a model which is non-linearly in the parameters – it is an example of a *non-linear regression model*.

Although very many parametric models exist, they are all somewhat inflexible in their description of f. They cannot accommodate arbitrary fluctuations in $f(x)$ over x because they contain only a small number of parameters (degrees-of-freedom).

Non-parametric models

In such models, f is assumed to be a smooth function of x, but otherwise we do not know what f looks like. A *smooth function* f is such that $f(x_i)$ is close to $f(x_j)$ whenever x_i is close to x_j. To characterise and fit f we will use an approach based on *splines*. In practice, different approaches to characterizing and fitting smooth f lead to similar fits to the data. The spline approach fits neatly with normal and generalised linear models (NLMs and GLMs), but so do other approaches (for example, kernel smoothing and wavelets). Methods of fitting f based on kernel smoothing and the Nadaraya–Watson estimator are studied in the Level 5 component of MATH5714 (Linear regression, robustness and smoothing) where the choice of bandwidth in kernel methods is analogous to the choice of smoothing parameter value in spline smoothing.

Piecewise polynomial models

A common problem with low-order polynomials is that they can often fit well for part of the data but have unappealing features elsewhere. For example, although none of the models in Figure 1.1 fit the data at all well, we might imagine that three short linear segments might be a good fit to the coal-seam data. Also, the piecewise linear model was a good description of the data in Figure 1.2c. This suggests that local polynomial models might be useful. In some situation, for example when we know that the function f is continuous, jumps in the fitted model, as in Figure 1.2c, are unacceptable. Alternatively, we may require differentiability of f. Such technical issues lead to the use of *splines,* which is introduced in the next chapter.

1.4 Exercises

1.1 Consider the first three models fitted in Figure 1.1 and let the data be denoted, $\{(x_i, y_i) : i = 1, 2, \ldots, n\}$. These three models can be written

$$\begin{aligned} (a) \quad & y = \alpha + \epsilon \\ (b) \quad & y = \alpha + \beta x + \epsilon \\ (c) \quad & y = \alpha + \beta x + \gamma x^2 + \epsilon \end{aligned}$$

where ϵ represents normally distributed random error. Use the principle of least squares, or otherwise, to obtain estimates of the model parameters.

Click here to see hints.

For each, start by defining the residual sum of squares (RSS), $RSS = \sum(y_i - \hat{y}_i)^2$ where, in turn (a) $\hat{y}_i = \alpha$, (b) $\hat{y}_i = \alpha + \beta x_i$, (c) $\hat{y}_i = \alpha + \beta x_i + \gamma x_i^2$. Then, find he parameter values which minimise the RSS by (possibly partial) differentiation.

1.2 Discuss possible approaches to fitting an exponential model, $y = \alpha e^{\beta x}$, to data. Note that no actual algebraic derivation, nor numerical coded algorithm is expect.

Click here to see hints.

There is more than one approach. Can least squares be used? Could a simple transformation of the data and the fitted model make solving the problem easier? Is there an algebraic solution? Is there a purely numerical solution?

1.3 In Figure 1.2c, discuss how you might fit a two-part linear model for the case where the change-point is *unknown*. Note that no actual algebraic derivation, nor numerical coded algorithm is expect.

Click here to see hints.

With many similar problems, imaging breaking the problem down into steps. If you know the location of the change-point then what should you do? Can you then try different possible change-point locations?

1.4 Discuss the four fitted models in Figure 1.3. Can you give positive and negative properties of each model? Which do you think is best and which worst? Do you think which is best/worst, depends on the data? Justify your answers.

Click here to see hints.

Don't get too stuck on the data used here, but think of general issues: ease of use, reliability of the data, is there error with the data or is it very reliable? What if the response it discrete?

> **Note**
>
> Exercise 1 Solutions can be found here.

2 Introducing Splines

Here is a short video [3 mins] to introduce the chapter.

2.1 Basic definitions

Let $t_1 < t_2 < ... < t_m$ be a fixed set of *sites* or *knots* which need not correspond to observation locations, as in Figure 2.1.

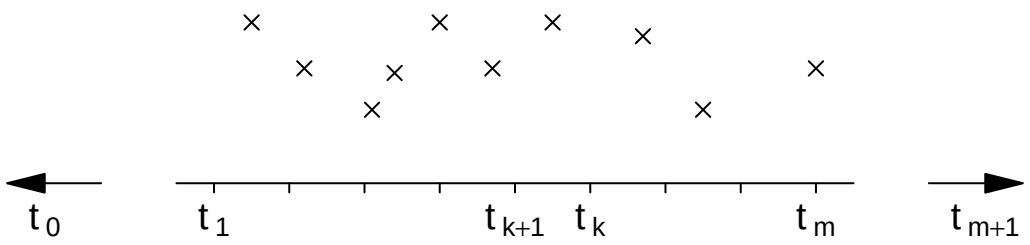


Figure 2.1: Diagram of knots and data points.

Note that we use the symbol t, rather than x, so that we do not confuse knots and observation locations.

A spline of order $p \geq 1$ is a piecewise-polynomial of order p which is $(p-1)$ times differentiable at the knots. Thus there are coefficients $\{a_{k\ell},\ k = 0, ..., m,\ \ell = 0, ..., p\}$ such that

$$f(t) = \sum_{\ell=0}^{p} a_{k\ell}\, t^{\ell}, \qquad \text{for } t_k \leq t < t_{k+1}, \tag{2.1}$$

where we take $t_0 = -\infty$ and $t_{m+1} = +\infty$.

If we are using cubic polynomials, $(p = 3)$, then f is given by the following equations:

$$f(t) = a_{00} + a_{01}t + a_{02}t^2 + a_{03}t^3, \quad t_0 \leq t < t_1$$

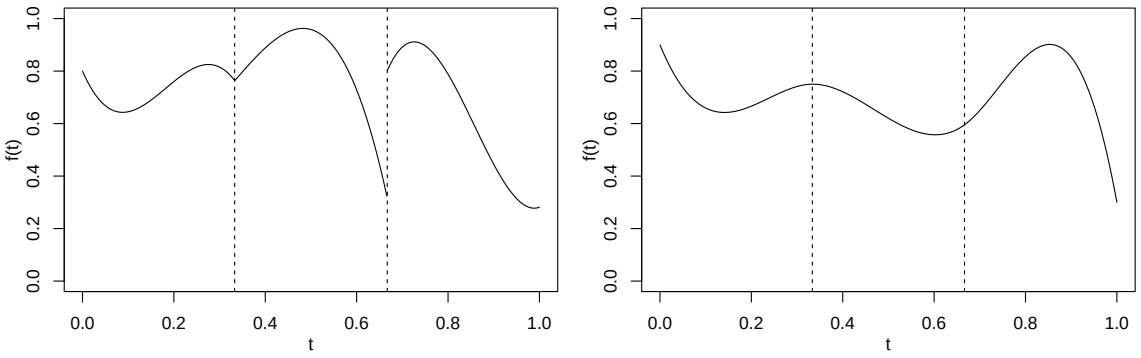
to the left of the first knot,

$$f(t) = a_{10} + a_{11}t + a_{12}t^2 + a_{13}t^3, \quad t_1 \leq t < t_2$$

between the first and second knots, and so on until

$$f(t) = a_{m0} + a_{m1}t + a_{m2}t^2 + a_{m3}t^3, \quad t_m \leq t < t_{m+1}$$

to the right of the final knot. This is illustrated in Figure 2.2a with $m = 2$.



(a) No smoothness constraints

(b) With smoothness constraints

Figure 2.2: Piecewise-cubic functions in three intervals with knot positions indicated with vertical lines.

Focus on splines quiz

Test your knowledge recall and comprehension to reinforce basic ideas about splines.

1. Which of the following best describes the relationship between knots and data locations.

- (A) Knots are the response and data locations are the explanatory variable
- (B) There must be fewer knots than data locations
- (C) Knots and data locations are both marked on the x-axis
- (D) Knots and data points are exactly the same

2. How many polynomial equations are need to define a cubic spline with four knots?

- (A) 3
- (B) 4

- (C) 5
- (D) 12
- (E) None of the above

3. What is the highest power possible in a cubic spline?

- (A) 0
- (B) 1
- (C) 2
- (D) 3
- (E) None of the above

4. How many times can a spline of order 5 be differentiated at the knots?

- (A) Cannot be differentiated
- (B) Twice differentiable
- (C) Four times differentiable
- (D) Infinitly differentiable

5. Which of the following is NOT a property of cubic splines?

- (A) Piecewise cubic polynomial
- (B) Twice differentiable
- (C) May contain change-points
- (D) Continuous at the knots

2.2 Imposing smoothness

Because of the use of polynomials, f is smooth *between* each successive pair of knots. At the knots, however, f might not be continuous and it might not be differentiable – in such cases we would say that the function is not smooth.

To ensure that f is also smooth at each of the knots, we impose smoothness constraints which control continuity of the function and its derivatives at the knots.

Let $f^{(\ell)}$ be the ℓ-th order derivative, with $f^{(0)} = f$ being the function itself, $f^{(1)} = f'$ is the first derivative and $f^{(2)} = f''$ the second derivative. Further, let $f^{(\ell)}(t-\epsilon)$ and $f^{(\ell)}(t+\epsilon)$, for $\epsilon \geq 0$, denote evaluation of the function or its derivative at points just below and just above t – we will be interested in their relative values as $\epsilon \to 0$.

To impose smoothness, we require that

$$\lim_{\epsilon \to 0} f^{(\ell)}(t_k - \epsilon) = \lim_{\epsilon \to 0} f^{(\ell)}(t_k + \epsilon), \tag{2.2}$$

for all $k = 1, \ldots, m$ and for $\ell = 0, \ldots, (p-1)$.

In other words we say that f is smooth if the limits, from below and from above, of the function and its $(p-1)$ derivatives exist and are equal.

The meaning of these smoothness constraints is illustrated in Figure 2.2. In Figure 2.2a, a piecewise cubic function with two knots has been plotted. The first derivative, f', is discontinuous at the first knot and the function itself, f, is discontinuous at the second knot. Figure 2.2b shows a similar shaped cubic spline with two knots. This time, the function f and its first two derivatives are continuous at both knots.

The smoothness conditions in Equation 2.2 induce constraints on the coefficients $\{a_{k\ell}\}$. A polynomial of order p has $p+1$ coefficients, and there are $m+1$ intervals when we have m knots. This leads to $(p+1) \times (m+1)$ coefficients but there are p constraints at each of the m knots. Thus the total *degrees of freedom* of the system is

$$\mathrm{df}_{\mathrm{spline}} = (p+1)(m+1) - pm = m + p + 1. \tag{2.3}$$

These degrees of freedom provide the necessary flexibility in the spline.

Note that f is infinitely differentiable everywhere, except at the knots where it is $p-1$ times differentiable. In particular, for $p = 1$, f is a linear spline comprising linear pieces constrained to be continuous at the knots, although the slope of f is discontinuous at the knots. Also, for $p = 3$, f is a cubic spline comprising cubic polynomial pieces continuous at the knots; where the first and second derivatives of f are also continuous, but the third derivative is discontinuous at the knots.

Focus on smoothness quiz

Test your knowledge recall and comprehension to reinforce basic ideas about smoothness.

1. Which of the following best describes the motivation for using smooth fitted models?

- (A) Makes model fitting easier
- (B) Calculations are easy to do in R
- (C) It produces nice graphs
- (D) Reduces the effect of measurement error
- (E) None of the above

2. Which of the following best describes the motivation for using piecewise polynomial components?

- (A) Can involve change-points
- (B) Well understood and easy to use
- (C) Can model jumps well
- (D) They lead to normally distributed errors
- (E) None of the above

3. Is the following a true statement? 'The higher the order of the polynomial components the smoother the spline' TRUE / FALSE

4. If a model fitted to a particular data set has zero degrees of freedom, then which of the following statements about the solution is most likely to be true?

- (A) Does not fit the data well
- (B) No solution
- (C) Unique solution
- (D) Multiple solutions

5. Which of the following best describes spline modeling?

- (A) The only non-parametric method available
- (B) It is a parametric approach
- (C) Requires high-level coding
- (D) A flexible non-parametric approach

2.3 Exercises

2.1 Why is it not sensible to define a smooth function made-up of constant components? Similarly, why is not sensible to create a differentiable function from linear splines?

Click here to see hints.

For each case, think about the number of parameters for each component and the implications of any constraints.

2.2 In the situation illustrated in Figure 2.2b, where $p = 3$ and $m = 2$, clearly identify the $(p + 1) \times (m + 1) = 12$ model parameters and the $pm = 6$ smoothness constraints in terms of the cubic polynomials and their derivatives.

Click here to see hints.

Define a cubic polynomial for each interval and consider continuity and differentiability.

2.3 Further consider the situation illustrated in Figure 2.2b. Suppose now that we require the splines to pass through specified coordinates $(t_1, f(t_1))$ and $(t_2, f(t_2))$. What is the degrees of freedom for this model? How many such cubic splines would satisfy these constraints? Discuss potential additional constraints which would lead to a unique fitted model. Do you think having a unique solution is a positive or negative property?

Click here to see hints.

Think about the degrees of freedom, that is the total number of parameters and the number of constraints, including forcing the spine to pass through two points. Think about the implications of having zero and non-zero degrees of freedom. There are very many (infinitely many?) potential additional constraints, but suggest one or two which sound a good idea.

2.4 For a general problem, what would be the effect of requiring additional constraints of the form of Equation 2.2 but with $\ell = p$? Would this lead to an acceptable fitted cubic spline model? Justify your answer.

Click here to see hints.

Think about the implication of this on the curvature of neighbouring components, and hence on overall curvature.

> **Note**
>
> Exercise 2 Solutions can be found here.

3 Interpolating Splines

Here is a short video [3 mins] to introduce the chapter.

3.1 Overview

Chapter 1 considered general limitations of parametric models, and polynomial regression in particular (see Figure 1.1), which motivated the use of the more flexible spline models (see Figure 1.3) – though at that stage no mathematical details were presented. In Chapter 2, basic spline definitions were given, including the notation of smoothness constraints, and these ideas were further explored in the Exercises in Section 3.6. This chapter will now give mathematical details of the interpolating spline problem and consider application to data. A feature of all these interpolation methods is that they fit the data exactly and that the fitted functions are smooth. Figure 3.1, is *cubic spline interpolation*, which fits a cubic polynomial between successive data points such that the function, gradient and the curvature are all continuous at each data point. The solid line shows the fitted values within the range of the data, whereas the dashed line shows the fitted values outside the range of the data – *extrapolation*.

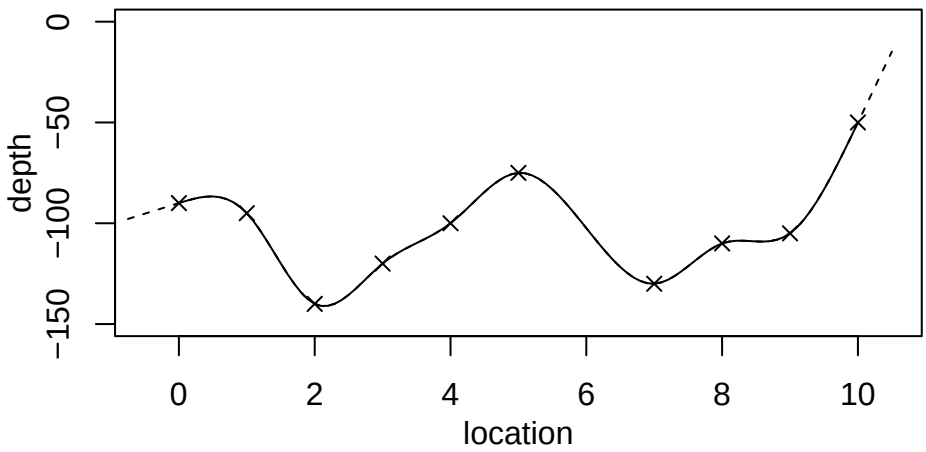


Figure 3.1: A cubic interpolating spline fitted to the coal-seam data, with the dashed line showing extrapolation.

3.2 Natural splines

Suppose we have n observations $\{y_1, \ldots, y_m\}$ at locations $\{t_1, \ldots, x_m\}$. We can construct a cubic spline (that is with $p = 3$) to pass through (interpolate) all the points (t_i, y_i), $i = 1, \ldots, m$. In fact, for any given set of points, there is an infinite number of cubic splines which interpolate them, see Figure 3.2 for examples. Exactly one of these splines has the property that, in the leftmost and rightmost intervals, it is a straight line. Such a spline is called a *natural* cubic spline.

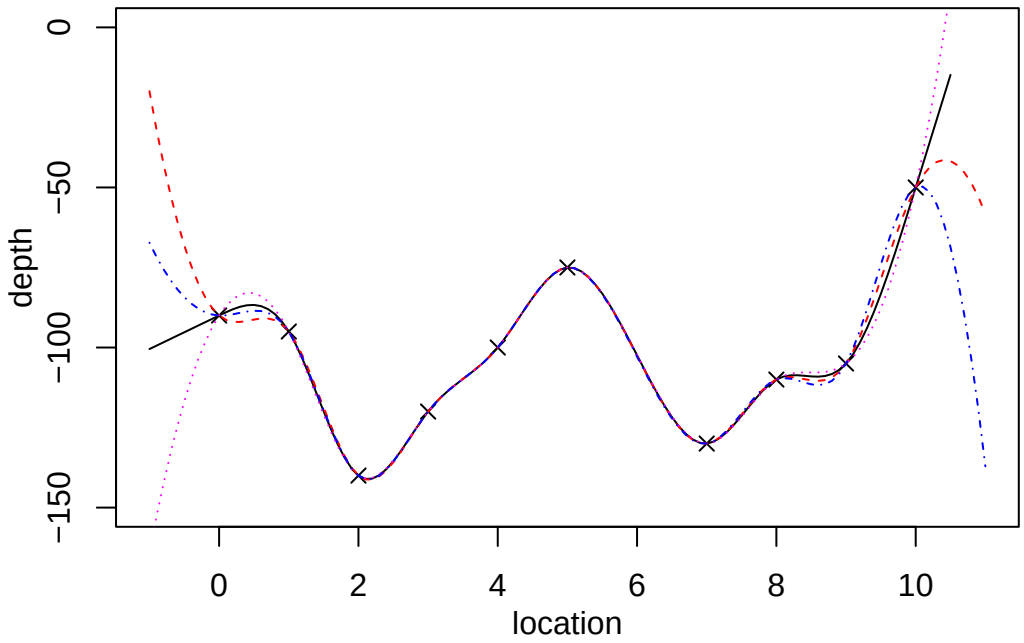
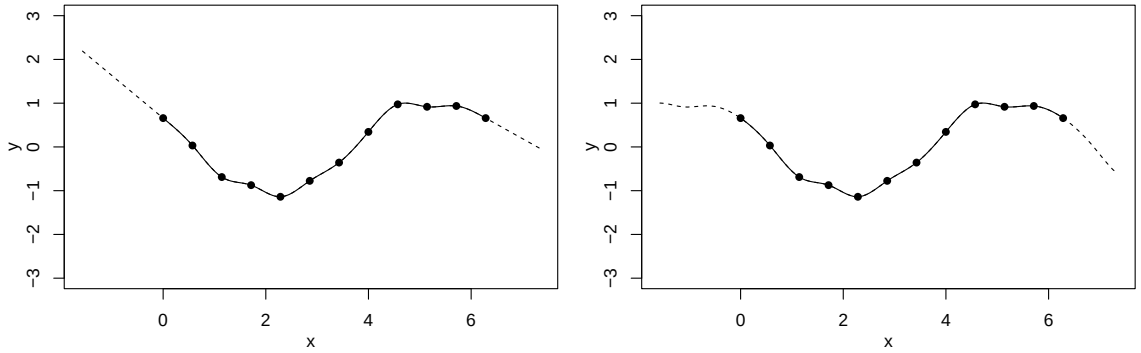


Figure 3.2: Cubic interpolating splines fitted to the coal-seam data, with the dashed lines showing extrapolation – the natural spline is shown in solid black.

Note that natural splines are not the only choice for the spline method – see the R help page for other options – with perhaps the most useful other being `periodic` which can be considered if we expect the unknown function to also be periodic. Figure 3.3 shows the fitted natural and periodic spline fitted to a simulated *cosine* function. The only noticeable difference is outside the range of the data, that is for extrapolation. Great care should be used, however, as imposing such additional restrictions on the fitted function can lead to unforeseen modelling errors when we do not observe the full range of `x` values.

3.3 Properties of natural splines

Natural splines are a special case of polynomial splines of *odd* order p. Thus we have natural linear splines ($p = 1$), natural cubic splines ($p = 3$), etc. A spline is said to be *natural* if,



(a) Using method="natural" (b) Using method="periodic"

Figure 3.3: Example of different interpolating splines.

beyond the boundary knots t_1 and t_m, its $(p+1)/2$ higher-order derivatives are zero:

$$f^{(j)}(t) = 0, \tag{3.1}$$

for $j = (p+1)/2, \dots, p$ and either $t \leq t_1$ or $t \geq t_m$.

Thus a natural spline of order p has the following $p+1$ constraints, in addition to those of Equation 2.2 :

$$\lim_{\epsilon \to 0} f^{(\ell)}(t_1 - \epsilon) = \lim_{\epsilon \to 0} f^{(\ell)}(t_m + \epsilon) = 0, \tag{3.2}$$

for $\ell = (p+1)/2, \dots, p$.

In particular,

- a natural *linear* spline has $p + 1 = 2$ additional constraints:

$$\lim_{\epsilon \to 0} f^{(1)}(t_1 - \epsilon) = \lim_{\epsilon \to 0} f^{(1)}(t_m + \epsilon) = 0, \tag{3.3}$$

  implying that $f(t)$ is constant in the outer intervals of a natural linear spline,
- a natural *cubic* spline has $p + 1 = 4$ additional constraints:

$$\lim_{\epsilon \to 0} f^{(2)}(t_1 - \epsilon) = \lim_{\epsilon \to 0} f^{(2)}(t_m + \epsilon) = 0,$$

$$\lim_{\epsilon \to 0} f^{(3)}(t_1 - \epsilon) = \lim_{\epsilon \to 0} f^{(3)}(t_m + \epsilon) = 0, \tag{3.4}$$

  implying that $f(t)$ is linear in the outer intervals of a natural cubic spline.

The total degrees of freedom of a natural spline is, starting from Equation 2.3, but taking into account the additional $p + 1$ additional constraints is

$$\mathrm{df}_{\text{nat.spline}} = m + p + 1 - (p + 1) = m. \tag{3.5}$$

That is the degrees of freedom for natural splines equals m whatever the value of p.

Proposition 3.1. *Linear and cubic natural splines have the following representations:*

- *Linear natural splines:*

$$f(t) = a_0 + \sum_{i=1}^{m} b_i \left| t - t_i \right| ; \quad \sum_{i=1}^{m} b_i = 0 \tag{3.6}$$

- *Cubic natural splines:*

$$f(t) = a_0 + a_1 t + \sum_{i=1}^{m} b_i \left| t - t_i \right|^3 ; \quad \sum_{i=1}^{m} b_i = \sum_{i=1}^{m} b_i t_i = 0. \tag{3.7}$$

Proof: Not covered here (but may be included later in the module if time allows).

Focus on natural splines quiz

Test your knowledge recall and comprehension to reinforce basic ideas about natural splines.

1. Which of the following is a key property of every spline?

- (A) Has jump discontinuities
- (B) Is a smooth non-parametric function
- (C) Is an example of a parametric model
- (D) Always interpolate data
- (E) Always extrapolate data

2. Which of the following is NOT a key property of a cubic spline function?

- (A) Is always continuous
- (B) Is always infinitely differentiable
- (C) Is a smooth function
- (D) Is a piecewise polynomial
- (E) Has extra contraints at knots

3. Which of the following statements is true ?

- (A) Natual splines are a special case of polynomial splines
- (B) Polynomial splines are a special case of natural splines
- (C) Polynomial splines are identical to natural splines
- (D) Natural splines have fewer constraints than polynomial splines
- (E) Natural splines are smoother than polynomial splines

3. Which of the following statements is a true ?

- (A) Natural splines have fewer constraints in the outer intervals
- (B) Natural splines have additional constraint in the internal intervals
- (C) Natural splines have additional constraints in the outer intervals
- (D) Natural splines are not smooth functions
- (E) Natural splines have fewer constraint in the internal intervals

5 What determines the degrees of freedom of a natural spline?

- (A) Both the number of knots and the polynomial order
- (B) The sample size
- (C) The number of knots
- (D) Both the number of knots and the sample size
- (E) The order of the polynomial used

3.4 Roughness penalties

An aim of spline models is to describe an unknown function using piecewise-polynomials which are smooth. In the previous section, smoothness was imposed by explicitly constraining specified high-order derivatives. An alternative approach is to measure and control the degree of smoothness of the splines. In practice the *roughness* of the spline is usually measured and one definition of roughness is:

$$J_\nu(f) = \int_{-\infty}^{\infty} \left[f^{(\nu)}(t)\right]^2 \mathrm{d}t \tag{3.8}$$

where $\nu \geq 1$ is an integer and $f^{(\nu)}$ denotes the νth derivative of f. Thus $f^{(1)}(t)$ denotes the first derivative and $f^{(2)}(t)$ denotes the second derivative of f.

Intuitively, roughness measures the "wiggliness" of a function.

Aim might be to find the smoothest function which interpolates the data points. Hence, an alternative approach to that in previous sections is to find the function f which minimizes Equation 3.8 and satisfies $f(t_i) = y_i$ for $i = 1, \ldots, m$. We refer to the solutions of this problem as the *optimal interpolating function*.

It turns out that there is a very close link between $J_\nu(\cdot)$ and pth-order natural splines, where $p = 2\nu - 1$ (so p is odd). Important special cases are: $\nu = 1$ and $p = 1$, and $\nu = 2$ and $p = 3$. This relationship is defined in the following proposition.

Proposition 3.2. *The optimal interpolating function is a pth-order natural spline, where $p = 2\nu - 1$. That is, the natural spline f is the unique minimizer of $J_\nu(f)$.*

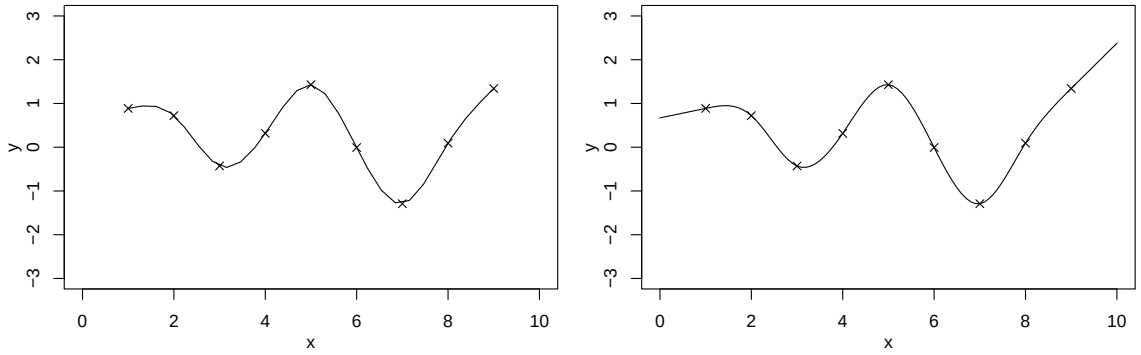
Proof: Not covered here (but may be included later in the module if time allows).

Comments

- Linear and cubic interpolating splines are also of interest in numerical analysis, for example to interpolate tables of numbers.
- The linear interpolating spline is simply the piecewise-linear path connecting the data points.
- Of course, in the linear spline case, knot points are clearly visible as kinks in the interpolating function. But, in the cubic spline case, knots points are invisible to the naked eye. Hence, in general, there is little motivation to use higher-order splines.
- Numerical considerations: the interpolating spline solutions involve matrix inversion. The inversion of an $n \times n$ matrix involves $O(n^3)$ operations – hence it is time consuming if n is large (for example, $n = 1000$ or 10000). Fortunately there are tricks to reduce the computation to $O(n)$.

3.5 Fitting interpolating splines in R

There are two main function within **R** for fitting interpolating splines to data, `spline` which outputs fitted values for specified points or `splinefun` which returns an **R** *function* which can be used directly by other commands, such as `curve`. The following illustrates the two approaches.



(a) Using the spline command

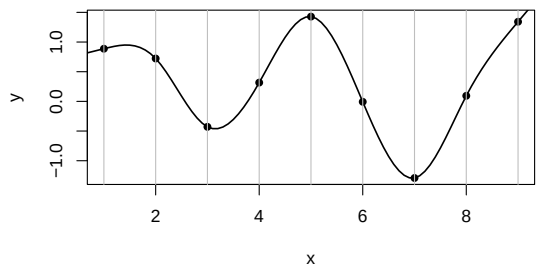
(b) Using the splinefun command

Figure 3.4: R code for cubic interpolating splines.

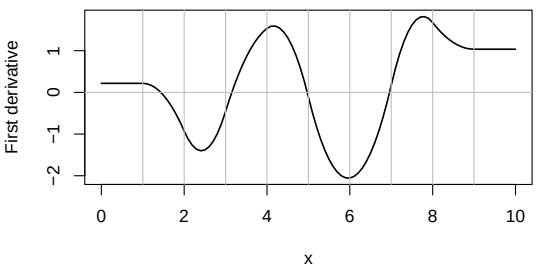
The following, illustrates the different ways to draw the spline and to calculated fitted values.

```
$x
[1] 2.5 7.5

$y
[1]  0.08785792 -0.78655273

[1]  0.08785792 -0.78655273
```

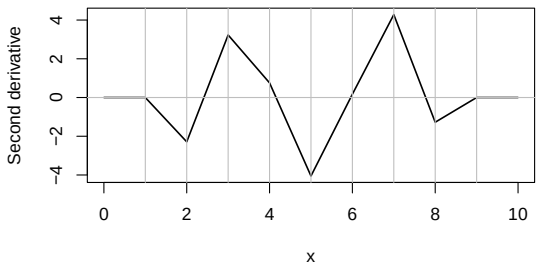
Before finishing, as we saw in Lecture 2, let us consider the derivatives of the fitted spline function. Figure 3.5 shows (a) the fitted natural spline, along with its first three derivatives in (b)-(d). Note that the function and the first two derivatives are continuous everywhere, but that the third derivatives is not continuous but has jumps at the knot locations. Note also that the first derivative, and higher derivatives, are all constant outside the range of the interior knots.



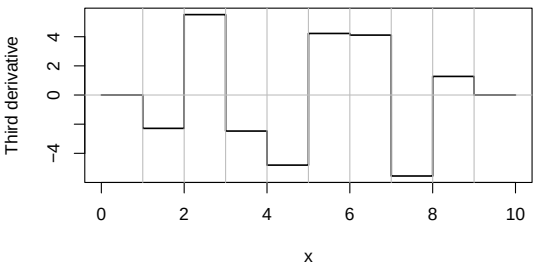
(a) Data and spline function



(b) First derivative



(c) Second derivative



(d) Third derivative

Figure 3.5: Derivatives of cubic interpolating splines.

Focus on fitting splines quiz

Test your knowledge recall and comprehension to reinforce basic ideas about R commands for splines.

1. Which of the following should be first in every data analysis?

- (A) Fit a spline model
- (B) Calculate the correlation
- (C) Add the fitted curve to a plot of the data
- (D) Fit a linear model
- (E) Draw a scatter plot

2. Which of the following is NOT part of a data analysis using interpolating splines?

- (A) Read the data description
- (B) Examine model residuals
- (C) Add the fitted curve to a plot of the data
- (D) Fit a spline model
- (E) Draw a scatter plot of the data

3. What is the main difference between the R commands spline and splinefun?

- (A) splinefun produces a function
- (B) spline interpolates whereas splinefun smooths
- (C) spline is more boring
- (D) Only spline has knots
- (E) They are identical, only the name is different

4. Which of the following is NOT an optional method for the R command spline?

- (A) fmm
- (B) periodic
- (C) natural
- (D) hyman
- (E) smooth

5. When using the output from splinefun, which of the following CANNOT be plotted?

- (A) Fourth derivative
- (B) Function value
- (C) Second derivative
- (D) First derivative
- (E) Third derivative

3.6 Exercises

3.1 For the situation shown in Figure 2.2, but taking $p = 1$, write-down the linear functions for the three intervals and clearly identify all the 6 model parameters. Next, write down the constraints required to make the functions pass through the $m = 2$ data points, and the two constraints which impose continuity of function. What additional constraints are needed to fix the first derivative at zero for the outer two intervals?

Click here to see hints.

You might get two redundant constraints but think carefully about which can be removed.

3.2 Continuing the problem described in Exercise 3.1, write the constraints as a system of 6 linear equations in the 6 unknown model parameters. How might you solve this system to give the parameter values which solve the interpolation problem?

Click here to see hints.

You need to write the six equations as a set of simultaneous equations. A simple matrix inversion is all that is needed for the solution – but don't try to actually do the inversion!

3.3 Continuing the linear system described in Exercise 3.2, create a synthetic problem by choosing two data response values. Then solve the system in **R**, or otherwise, and plot the fitted spline interpolating function.

Click here to see hints.

The choice of the two data point locations (x-values) and function value (y-value) is completely your choice. Then, define the two vectors and the matrix defined in the previous question. R has a function "solve" which will invert the design matrix – look at "help(solve)"

3.4 Again, considering the situation shown in Figure 2.2, but taking $p = 1$. Using the alternative representation in Equation 3.6, write down two constraints involving the data points and the additional constraint on the b_i parameters. Write this linear system of 3 equations in three unknowns in matrix form.

Click here to see hints.

The first part only requires checking the definition of the alternative form in the notes. Then, write it in vector/matrix form.

3.5 Continuing the linear system described in Exercise 3.4, using the same points created in Exercise 3.4, calculate the parameter values in this new parameterization. Check that your two fitted interpolating spline give the same answers. Which approach do you prefer? Justify you answer.

Click here to see hints.

Again, in R, define the vectors and matrix, and use "solve" to find the parameter estimates. You can check for equality by any approach, for example just plot both solutions and see if they match. The choice of which it preferable is yours, no wrong answer, but it's the justification that matters.

3.6 Create you own version of the R code used to produce Figure 3.4 and experiment with the two alternative spline fitting commands. Remove the `set.seed(15342)` command so that you produce different data each time and comment on the similarities and differences when using different data sets.

Click here to see hints.

Simply copy the code from the referred to figure and remove the set.seed command. Then, each time you run the code you will get a different answer.

3.7 Let

$$f(t) = 3 + 2t + 4|t|^3 + |t-1|^3.$$

Write f as a cubic polynomial in each of the intervals $(-\infty, 0)$, $(0, 1)$ and $(1, \infty)$. Verify that f and its first two derivatives are continuous at the knots.

Is f a spline? Is f a natural spline?

Click here to see hints.

To show this is a spline, you need to re-write the equations as polynomials. That is consider what happens to the absolute values in each interval. Then, substitute in the knot locations

(0 and 1) into the equations for function, derivative and second derivative and make sure all matches. There is also a solution based on the equation of the alternative form – see if you can argue it this way also.

3.8 Let

$$f(t) = 3 + |t| - |t - 2|.$$

Show by direct calculation that

$$\int_{-\infty}^{\infty} \{f'(t)\}^2 \, dt = 8.$$

Show that this integral can also be written in the form $-2\mathbf{b}^T K \mathbf{b}$, where you should define $\mathbf{b}$ and K.

Click here to see hints.

Use a similar approach as in the previous equation to "remove" the absolute value. You should then see that the integral is trivial. For the matrix form, go back to the definitions of each part. Multiplying out will then give the same answer as the integral.

> **Note**
>
> Exercise 3 Solutions can be found here.

4 Smoothing Splines

Here is a short video [4 mins] to introduce the chapter.

4.1 Overview

In Section 1.3 we described a general statistical model with a response variable y and an explanatory variable x. We observe y_i at each location x_i, for $i = 1, \ldots, n$. We imagined that the y's are noisy versions of a smooth function of t, say $f(\cdot)$ where the errors follow a normal distribution with constant variance. That is

$$y_i = f(x_i) + \epsilon_i, \quad \epsilon_i \sim \mathrm{N}(0, \sigma^2),$$

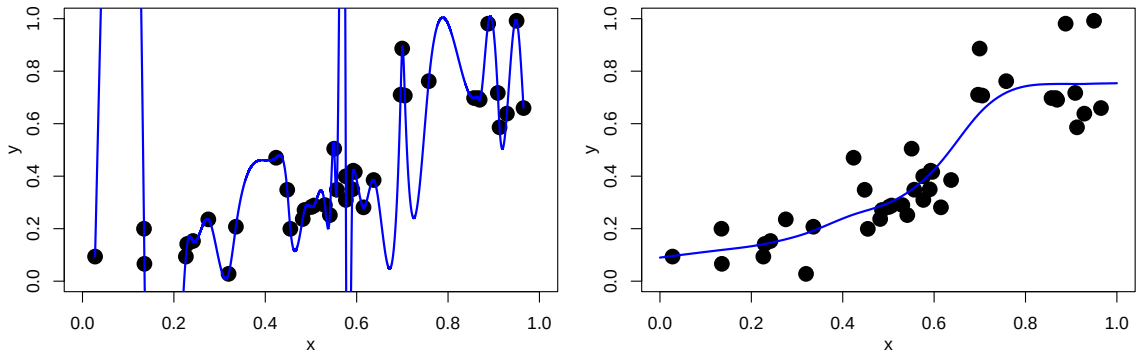
for $i = 1, \ldots, n$, where f is smooth, the ϵ_i are i.i.d., and f and σ^2 are unknown.
The log-likelihood for this situation is:

$$l(f; \mathbf{y}) = -\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - f(x_i))^2 - n \log \sigma \tag{4.1}$$

and we wish to estimate f for a given data set $\mathbf{y} = \{y_1, \ldots, y_n\}$. With no constraints on f, the log-likelihood would be maximized by setting $f(x_i) = y_i$ for all i, and we would estimate the noise variance as $\hat{\sigma}^2 = 0$. This takes no account of randomness in the data and f would in general need to be quite wiggly to achieve this fit.

Figure 4.1a shows such an interpolation of noisy data. This would be of little use for explanation, interpolation or prediction.

We do not expect, or even want, the fitted function f to pass exactly through the data points $\{(x_i, y_i), i = 1, \ldots, n\}$, but merely to lie close to them. We would rather trade-off goodness-of-fit against smoothness of f. Figure 4.1b shows a smoothing spline fit to the same data. This is much better as there is a clear explanation of the relationship, it could be used reasonably well for interpolation and prediction.



(a) Interpolating spline

(b) Smoothing spline

Figure 4.1: Comparison of interpolating and smoothing methods applied to a noisy dat set.

4.2 The penalized least-squares criterion

Noting that maximizing Equation 4.1 is equivalent to minimizing the residual sum of squares: $\sum_{i=1}^{n} (y_i - f(x_i))^2$, we can achieve this trade-off by minimizing a *penalized* sum of squared residuals:

$$R_\nu(f, \lambda) = \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda J_\nu(f), \tag{4.2}$$

where $J_\nu(f)$, as first defined in Equation 3.8, penalizes the roughness of f. The *smoothing parameter* $\lambda \geq 0$ controls the severity of this penalty. For now we will assume λ, which absorbs the σ^2 in Equation 4.1, is known.

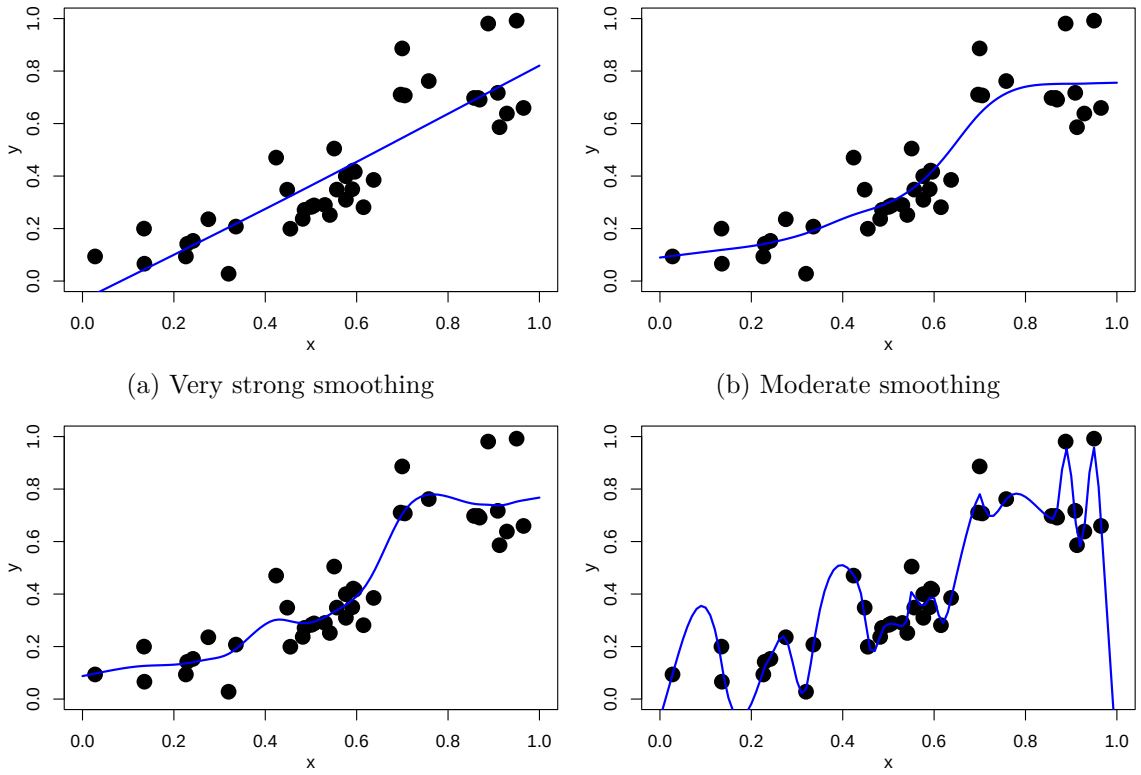
Figure 4.2 shows example smoothing spline fits using a range of smoothing parameters, λ. In Figure 4.2a the fit is essentially a straight line, perhaps Figure 4.2b and Figure 4.2c show acceptable fits. Figure 4.2d, with a very small λ value is close to an interpolating spline fit and is clearly unacceptable.

As the smoothing parameter λ increases, the optimal f becomes smoother. In particular, it can be shown that as $\lambda \to \infty$, the vector of coefficients $\mathbf{b} \to \mathbf{0}$, and $\mathbf{a}$ tends to the usual least-squares estimate of the regression parameters. Thus, the smoothing spline converges to the sample mean $f(x) = \bar{y}$ when $\nu = 1$, and to the ordinary least squares fitted line, $f(x) = \hat{\alpha} + \hat{\beta}t$, when $\nu = 2$. In the other direction, as $\lambda \to 0$ the smoothing solution converges to the interpolating spline.

Focus on smoothing splines quiz

Test your knowledge recall and comprehension to reinforce basic ideas about smoothing splines.

1. Which of the following statements is NOT true?



(a) Very strong smoothing

(b) Moderate smoothing

(c) Weak smoothing

(d) Very weak smoothing

Figure 4.2: Comparison of interpolating and smoothing methods applied to a noisy dat set.

- (A) The smoothing spline parameter controls the amount of smoothing
- (B) Interpolating splines pass through all data points
- (C) Smoothing splines should be used for all problems
- (D) Smoothing splines are better for noisy data
- (E) Both interpolating and smoothing splines are piecewise polynomial functions

2 Which of the following statements is true about splines?

- (A) The roughness measures the wiggliness of a function
- (B) The roughness can be integrated to give the smoothness
- (C) The roughness defines the likelihood function
- (D) The roughness is a measure of goodness of fit
- (E) The roughness measures the lack of fit

3. Which of the following best describes the result of using a very large value of in a cubic smoothing spline?

- (A) The spline is approximately quadratic
- (B) The spline is approximately constant
- (C) The spline is approximately linear
- (D) The spline is approximately cubic
- (E) The spline interpolates the data

4. Which of the following best describes the result of using a very small value of in a linear smoothing spline?

- (A) The spline is approximately quadratic

- (B) The spline is approximately cubic
- (C) The spline is approximately linear
- (D) The spline is approximately constant
- (E) The spline interpolates the data

5. Which of the following statements best describes how to choose the value of ?

- (A) To balance closeness to data and roughness
- (B) To minimise the roughness
- (C) To maximise the likelihood
- (D) To maximise the roughness
- (E) To better fit the data with a high roughness

4.3 Relation to interpolating splines

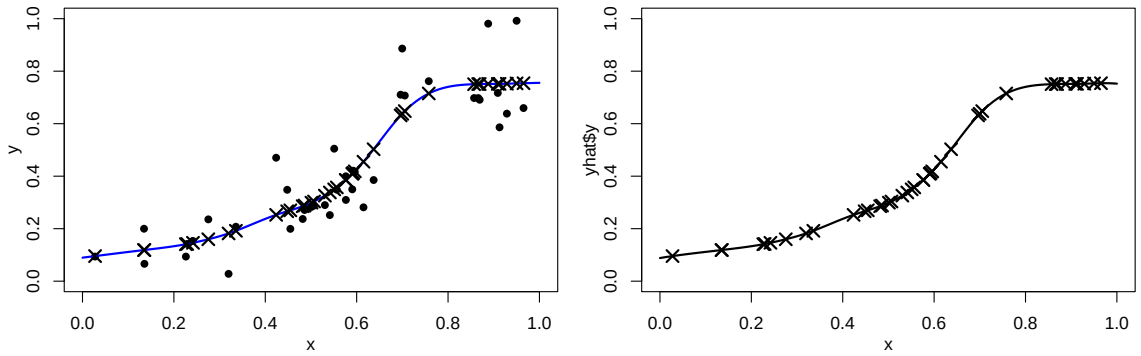
We show in the following proposition that the function f which minimizes Equation 4.2 is the interpolating spline of its fitted values.

Proposition 4.1. *Suppose $\hat{f}$ minimizes $R_\nu(f, \lambda)$ and let $\hat{y}_i = \hat{f}(x_i), i = 1, \ldots, n$ denote the corresponding fitted values. Then, $\hat{f}$ solves the interpolation problem for the* artificial *data set $(x_i, \hat{y}_i), i = 1, \ldots, n$. That is, $\hat{f}$ minimizes $J_\nu(f)$ over functions f satisfying $\hat{f}(x_i) = \hat{y}_i$, $i = 1, \ldots, n$. Consequently, $\hat{f}$ is a p^{th}-order natural spline, where $p = 2\nu - 1$. This means that, when solving the smoothing problem, we only need consider spline functions given by the representations in Proposition 3.1.*

Proof: Suppose the assertion is not true. In this case, we must be able to find a function $\hat{f}^\star$, say, which also interpolates the artificial data $(x_i, \hat{y}_i), i = 1, \ldots, n$, but which has a smaller roughness penalty. That is

$$J_\nu(\hat{f}^\star) < J_\nu(\hat{f}) \text{ with } \hat{f}^\star(x_i) = \hat{y}_i, \ i = 1, \ldots, n.$$

Note that the fitted values from the function $\hat{f}^\star$ are also equal to $\hat{y}_i, i = 1, \ldots, n$ as it interpolates the same artificial data as $\hat{f}$.



(a) Smoothing spline of data shown as dots and with fitted values shown as crosses

(b) Interpolating spline of fitted values from smoothing problem

Figure 4.3: Illustration of proposition showing that solution of the smothing problem is a natural interpolating spline.

Now, from Equation 4.2,

$$\begin{aligned} R_\nu(\hat{f}^\star, \lambda) &= \sum_i (y_i - \hat{y}_i)^2 + \lambda J_\nu(\hat{f}^\star) \\ &< \sum_i (y_i - \hat{y}_i)^2 + \lambda J_\nu(\hat{f}) = R_\nu(\hat{f}, \lambda). \end{aligned}$$

Hence $R_\nu(\hat{f}^\star, \lambda) < R_\nu(\hat{f}, \lambda)$. But, by construction $\hat{f}$ minimizes $R_\nu(f, \lambda)$, which is a contradiction. Hence, it must not be possible to find a function $\hat{f}^\star$ which also interpolates the artificial data but which has a smaller roughness penalty.

We have shown that $\hat{f}$ is the optimal interpolant of the fitted values $\hat{y}_i,\ i = 1, \ldots, n$, so it follows from Proposition 4.1 that $\hat{f}$ is a natural spline of order $p = 2\nu - 1$.

4.4 The smoothing problem in matrix notation

We have just proved that the function $\hat{f}$ that minimizes $R_\nu(f, \lambda)$ must be a natural spline (linear if $\nu = 1$, cubic if $\nu = 2$) with knots at $\{t_i,\ i = 1, \ldots, n\}$. That is, as in Proposition 3.1, we can write:

$$\hat{f}(t) = \sum_{i=1}^{n} b_i \, |t - t_i|^p + \begin{cases} a_0, & \nu = 1 \\ a_o + a_1 t, & \nu = 2, \end{cases} \tag{4.3}$$

where $p = 2\nu - 1$ and constraints

$$\sum_{i=1}^{n} b_i = 0 \ \text{ for } \ \nu = 1 \quad \sum_{i=1}^{n} b_i = \sum_{i=1}^{n} b_i \, t_i = 0 \ \text{ for } \ \nu = 2. \tag{4.4}$$

However, we have not yet figured out how to calculate the parameter values $\hat{a}_0, \ldots, \hat{a}_{\nu-1}$, $\hat{b}_1, \ldots, \hat{b}_n$ in Equation 4.3 which optimally fit the data $y_1, \ldots, y_n$. For this, it is convenient to re-express the penalized sum of squared residuals Equation 4.2 in matrix notation.

Proposition 4.2. *The roughness of a natural linear spline ($\nu = 1$, i.e. $p = 1$) is*

$$J_1(f) = -2 \sum_{i=1}^{n} \sum_{k=1}^{n} b_i b_k \, |t_i - t_k| = c_1 \, \mathbf{b}^T \, K_1 \, \mathbf{b}, \tag{4.5}$$

and the roughness of a natural cubic spline ($\nu = 2$, i.e. $p = 3$) is

$$J_2(f) = 12 \sum_{i=1}^{n} \sum_{k=1}^{n} b_i b_k \, |t_i - t_k|^3 = c_2 \, \mathbf{b}^T \, K_2 \, \mathbf{b}, \tag{4.6}$$

where the constants are given by

$$c_1 = -2, \quad c_2 = 12. \tag{4.7}$$

Here $\mathbf{b} = (b_1, \ldots, b_n)^T$ is the vector of spline coefficients and K_ν is the $n \times n$ matrix whose (i,k)th element is $|t_i - t_k|^p$, where $p = 2\nu - 1$ that is

$$K_\nu = \begin{bmatrix} |t_1 - t_1|^p & |t_1 - t_2|^p & \ldots & |t_1 - t_n|^p \\ |t_2 - t_1|^p & |t_2 - t_2|^p & \ldots & |t_2 - t_n|^p \\ \vdots & \vdots & \ddots & \vdots \\ |t_n - t_1|^p & |t_n - t_2|^p & \ldots & |t_n - t_n|^p. \end{bmatrix}$$

Proof: To be covered later if there is sufficient time.

From Proposition 4.2 the roughness penalty satisfies:

$$J_\nu(\hat{f}) = c_\nu \, \mathbf{b}^T K_\nu \mathbf{b} \tag{4.8}$$

where $c_1 = -2$; $c_2 = 12$; $\mathbf{b} = (b_1, \ldots, b_n)^T$; and K_ν is the $n \times n$ matrix whose (i,k)th element is $|t_i - t_k|^p$.

From Equation 4.3, the value of $\hat{f}$ at knot t_k is

$$\hat{f}(t_k) = \sum_{i=1}^{n} b_i \, |t_k - t_i|^p + \begin{cases} a_0, & \nu = 1 \\ a_0 + a_1 t_k, & \nu = 2 \end{cases}. \tag{4.9}$$

In matrix form, this is

$$\hat{\mathbf{f}} = \begin{bmatrix} \hat{f}(t_1) \\ \vdots \\ \hat{f}(t_n) \end{bmatrix} = K_\nu \mathbf{b} + L_\nu \mathbf{a}_\nu \tag{4.10}$$

where

$$L_1 = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}, \quad \mathbf{a}_1 = a_0, \qquad \text{and} \qquad L_2 = \begin{bmatrix} 1 & t_1 \\ \vdots & \vdots \\ 1 & t_n \end{bmatrix}, \quad \mathbf{a}_2 = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix}.$$

Thus, from Equation 4.4 – Equation 4.10, the penalized least-squares criterion Equation 4.2 reduces to a quadratic function of the parameters $\mathbf{a}$ and $\mathbf{b}$:

$$R_\nu(f, \lambda) = (\mathbf{y} - K_\nu \mathbf{b} - L_\nu \mathbf{a})^T (\mathbf{y} - K_\nu \mathbf{b} - L_\nu \mathbf{a}) + \lambda c_\nu \mathbf{b}^T K_\nu \mathbf{b} \tag{4.11}$$

subject to

$$L_\nu^T \mathbf{b} = 0, \tag{4.12}$$

where $\mathbf{y} = (y_1, \ldots, y_n)^T$.

To find the explicit values for $\mathbf{b}$ and $\mathbf{a}$, we must minimize the quadratic function Equation 4.11 subject to the linear constraints Equation 4.12.

Proposition 4.3. *The solution to the smoothing spline problem is given by*

$$\begin{bmatrix} \hat{\mathbf{a}} \\ \hat{\mathbf{b}} \end{bmatrix} = \begin{bmatrix} 0 & L_\nu \\ L_\nu^T & K_\nu + \lambda^* I_n \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{y} \end{bmatrix} \tag{4.13}$$

and $\lambda^* = c_\nu \lambda$.

Proof: Omitted.

4.5 Smoothing splines in R

Fitting a smoothing spline to data involved estimating values of $\mathbf{a}$ and $\mathbf{b}$ to minimize the penalized roughness measure in Equation 4.11 subject to the constraints in Equation 4.12 which yields the explicit equation in Equation 4.13.

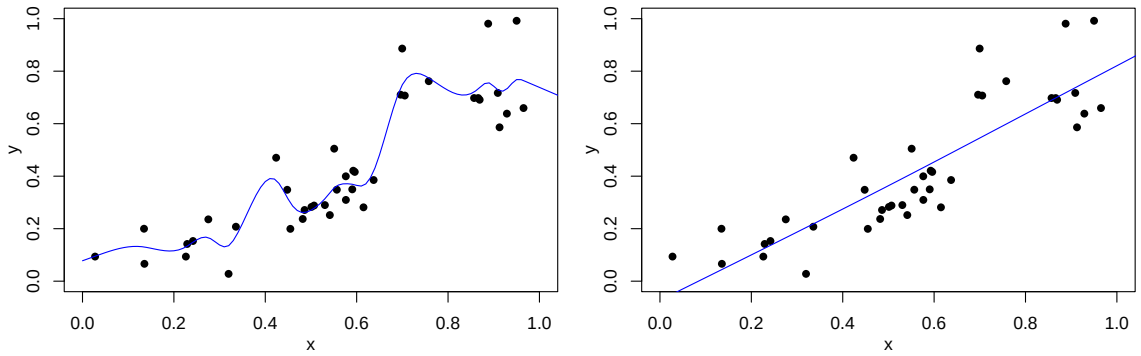
Of course, it is possible to code the solution of this matrix system, but it is usual instead to use in-built commands in software such as **R**.

In **R** the basic command is:

where `x` and `y` are vectors of data coordinates, and `lambda` specifies the value of the smoothing parameter to use – note that the effect of a given value of λ depends on the problem considered.

For example, consider the synthetic data set which contains a break-point at $x = 2/3$ first met in Figure 1.2. Fitted spline curves with $\lambda = 0.0001$ and $\lambda = 1$, shown in Figure 4.4, are based on the following commands:

If no value of the smoothing parameter is specified, the the optimal value is calculated using generalized cross-validation as discussed in the next chapter.



(a) Weak smoothing

(b) Strong smoothing

Figure 4.4: Simulated data superimposed with fitted smoothing splines.

4.6 Exercies

4.1 Recall, from MATH3823 Chapter 3, the tropical cyclone data recording the number of cyclones in each of 13 seasons:

Table 4.1: Numbers of tropical cyclones in $n = 13$ successive seasons[1]

Season	1	2	3	4	5	6	7	8	9	10	11	12	13
No of cyclones	6	5	4	6	6	3	12	7	4	2	6	7	4

It is reasonable to allow the average number of cyclones to vary with season, rather than assuming a constant rate.

Fit a cubic smoothing spline to these data, using a range of values for the smoothing parameter λ to estimate the time-varying cyclone rate. By eye, suggest a suitable value for λ and justify your choice. Compare the smoothed estimate of rate with the value obtain by assuming a constant rate.

Click here to see hints.

Follow the R code used in Section 4.5 but don't worry about finding an exact value for the smoothing parameter, only the order of magnitude.

4.2 Consider the *Old Faithful* data set on geyser eruptions available in R. Use the following commands to learn more about and visualize the data:

```
data(faithful)
help(faithful)
plot(faithful)
```

[1] Dobson and Barnett, 3rd edn, Table 1.2

Fit a cubic smoothing spline to these data, using a range of values for the smoothing parameter λ. By eye, suggest a suitable value for λ.

Click here to see hints.

Taking waiting as the explanatory and eruption as the response, follow the R code used in Section 4.5 but don't worry about finding an exact value for the smoothing parameter, only the order of magnitude.

> **Note**
>
> Exercise 4 Solutions can be found here.

5 Choosing the Smoothing Parameter

Here is a short video [4 mins] to introduce the chapter.

5.1 Overview

Suppose we are given data $D = \{(t_i, y_i),\ i = 1, \dots, n\}$ and that our model is:

$$y_i = f(t_i) + \epsilon_i, \qquad \epsilon_i \sim \mathrm{N}(0, \sigma^2) \tag{5.1}$$

where the ϵ_i are i.i.d. $\sim \mathrm{N}(0, \sigma^2)$ and $f(t)$ is assumed to be smooth. Given knot positions $\{t_i,\ i = 1, \dots, n\}$, we can estimate $f(t)$ with a smoothing spline $\hat{f}_\lambda(t)$.

How then should we choose the value of the smoothing parameter λ? By setting $\lambda \to 0$, we obtain exactly the interpolating spline $\hat{f}_0(t)$ and a perfect fit to the data. However, this tends to *overfit* the data: applying it to to a new sample of data where model Equation 5.1 still applies would produce a poor fit. Conversely, by setting $\lambda \to \infty$, we get:

$$f_\infty(t) = \begin{cases} \hat{a}_0, & \nu = 1,\ p = 1 \\ \hat{a}_0 + \hat{a}_1 t, & \nu = 2,\ p = 3. \end{cases}$$

Here, $\hat{a}_0 = \bar{y}$, for the $\nu = 1, p = 1$ case, and $\{\hat{a}_0,\ \hat{a}_1\}$, for the $v = 2, p = 3$ case, are the OLS linear regression parameters.
If the true $f(t)$ was constant or linear, this solution would be reasonable, but often we are interested in less regular functions.

5.2 Training/test approach

One way to approach estimation of λ is to partition the set of indices $I = \{1, \dots, n\}$ into two subsets I_1 and I_2, where $I_1 \cup I_2 = I$ and $I_1 \cap I_2 = \phi$. Thus we obtain two datasets:

- Training dataset: $D_1 = \{(t_i, y_i),\ i \in I_1\}$,
- Test dataset: $D_2 = \{(t_i, y_i),\ i \in I_2\}$.

We fit a smoothing spline $\hat{f}_{\lambda,I_1}(t)$ to the training dataset, and judge the quality of the fit using the test dataset:

$$Q_{I_1:I_2}(\lambda) = \sum_{i \in I_2} \left(y_i - \hat{f}_{\lambda,I_1}(t_i)\right)^2. \tag{5.2}$$

We choose λ to minimize $Q_{I_1:I_2}(\lambda)$. Many algorithms exist for such minimization, for example through evaluation on a fine grid of λ values, although many more computationally efficient algorithms exist.

5.3 Cross-validation or *leave-one-out*

This is an extreme form of the above principle. The test dataset D_2 comprises a single observation, (t_j, y_j), for a given value of j. The training set D_1 is then $D_{-j} = \{(t_i, y_i),\ i \in I_{-j}\}$, where I_{-j} denotes the full set I excluding j. Then in a slightly amended notation we can write

$$Q_{-j:j}(\lambda) = \left(y_j - \hat{f}_{\lambda,-j}(t_j)\right)^2$$

to assess the quality of fit. Of course, j is arbitrary, so we repeat this process for each $j \in \{1, \ldots, n\}$ then average the assessments to form the *ordinary cross-validation criterion*:

$$Q_{OCV}(\lambda) = \frac{1}{n} \sum_{j=1}^{n} \left(y_j - \hat{f}_{\lambda,-j}(t_j)\right)^2. \tag{5.3}$$

We then choose the value $\hat{\lambda}$ which minimizes $Q_{OCV}(\lambda)$. Hopefully, a plot of $Q_{OCV}(\lambda)$ will appear as in Figure 5.1, but there is no theoretical guarantee that this curve will have a unique turning point, making it difficult to locate the minimum.

At first sight, evaluation of $Q_{OCV}(\lambda)$ for a given λ appears computationally intensive: we must compute n different smoothing solutions, each corresponding to one of the *left-out* data points. Fortunately, there is a computational trick which enables us to compute $Q_{OCV}(\lambda)$ directly from the smoothing spline solution constructed from the whole dataset

5.4 The smoothing matrix

Here we show, for a given value of the smoothing parameter λ and the index $\nu \geq 1$, that the fitted value $\hat{f}_\lambda(t_k)$ at each knot t_k may be written as a linear combination of the observations, $y_1, \ldots, y_n$.

Recall from Proposition 4.3 that the smoothing spline $\hat{f}_\lambda(t)$, which minimizes the penalized sum of squares criterion Equation 4.2 for a given value of the smoothing parameter λ, has coefficients $\hat{\mathbf{a}}, \hat{\mathbf{b}}$ where

$$\begin{bmatrix} \hat{\mathbf{a}} \\ \hat{\mathbf{b}} \end{bmatrix} = M_\lambda^{-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{y} \end{bmatrix} \quad \text{where} \quad M_\lambda = \begin{bmatrix} 0 & L_\nu^T \\ L_\nu & K_\nu + \lambda^* I_n \end{bmatrix}$$

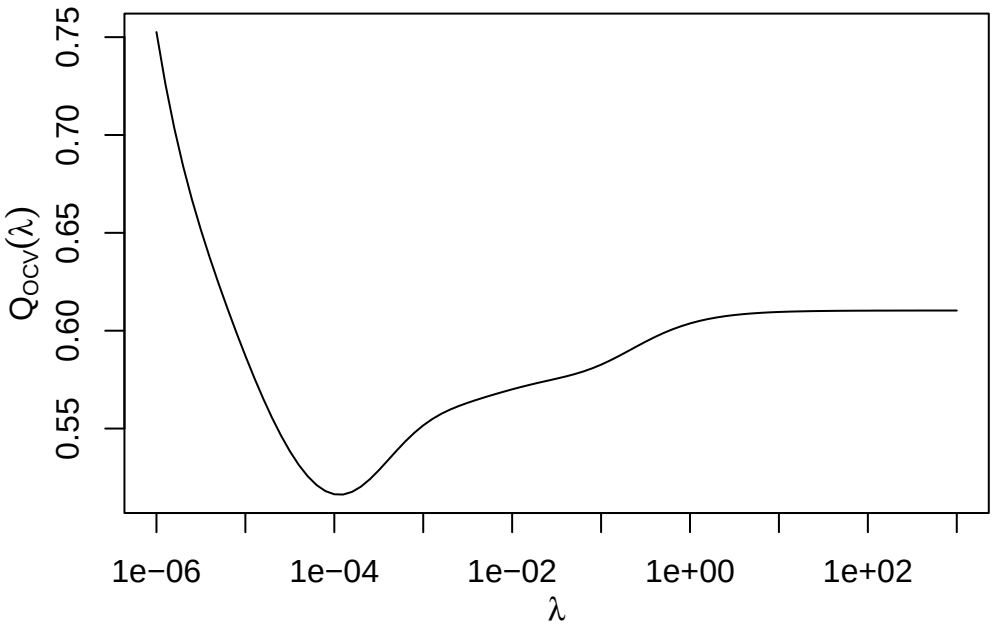


Figure 5.1: Ordinary cross validation plot

The fitted values of the smoothing spline at the knots can be represented in matrix form as

$$\hat{\mathbf{f}} = \begin{bmatrix} \hat{f}(t_1) \\ \vdots \\ \hat{f}(t_n) \end{bmatrix} = K\,\hat{\mathbf{b}} + L\,\hat{\mathbf{a}} = \begin{bmatrix} K & L \end{bmatrix} \begin{bmatrix} M_\lambda^{12} \\ M_\lambda^{22} \end{bmatrix} \mathbf{y},$$

where M_λ^{-1} has been partitioned in the form

$$M_\lambda^{-1} = \begin{bmatrix} M_\lambda^{11} & M_\lambda^{12} \\ M_\lambda^{21} & M_\lambda^{22} \end{bmatrix},$$

where M_λ^{12} is $\nu \times \nu$ and M_λ^{22} is $n \times n$.

It can be shown that the matrix

$$S_\lambda = \begin{bmatrix} K & L \end{bmatrix} \begin{bmatrix} M_\lambda^{12} \\ M_\lambda^{22} \end{bmatrix}$$

is a symmetric positive definite matrix for $\lambda > 0$ called the *smoothing* matrix. Then, S_λ connects the data $\mathbf{y}$ to the fitted values through

$$\hat{\mathbf{f}} = S_\lambda\,\mathbf{y}. \tag{5.4}$$

That is, the fitted values are simple linear function of the data.

5.5 Effective degrees of freedom

How many degrees of freedom are there in the smoothing spline? There are altogether $n+\nu$ parameters, ν in $\mathbf{a}$ and n in $\mathbf{b}$, but these are not completely free.

We showed earlier that as the smoothing parameter $\lambda \to \infty$, the smoothing spline $\hat{f}(t)$ becomes the least-squares regression solution for model formula $y \sim 1$ when $\nu = 1$, or model formula $y \sim 1 + t$ when $\nu = 2$. Thus, when $\lambda = \infty$, the degrees of freedom in the spline is ν. We also showed that when $\lambda = 0$, the smoothing spline $\hat{f}(t)$ becomes the interpolating spline, for which the degrees of freedom is the number of observations, n.

Thus, intuitively, for values of λ between the extremes of 0 and ∞, the spline degrees of freedom should lie somewhere between ν and n; the greater the smoothing, the fewer the *effective* degrees of freedom. How can we capture this notion precisely?

A clue comes from Ordinary Least Squares (OLS) regression, in which the fitted values are given by

$$\hat{\mathbf{y}} = X(X^T X)^{-1} X^T \mathbf{y} = H\mathbf{y},$$

where X is the $n \times p$ design matrix, where p is the number of model parameters. Here,

$$H = X(X^T X)^{-1} X^T$$

is called the **hat** matrix, which linearly maps the data $\mathbf{y}$ onto the fitted values $\hat{\mathbf{y}}$. Using the property that $\text{trace}(QR) = \text{trace}(RQ)$ for matrices Q, R of conformable dimensions, the trace of the hat matrix is:

$$\begin{aligned} \text{trace}(X(X^T X)^{-1} X^T) &= \text{trace}((X^T X)^{-1} X^T X) \\ &= \text{trace}(I_p) \\ &= p, \end{aligned}$$

where I_p is the $p \times p$ identity matrix. Thus, for OLS regression, we see that the trace of the hat matrix equals the number of model parameters.

Now in Equation 5.4 we see that the smoothing matrix S_λ takes the role of a hat matrix, since it linearly maps the data onto the fitted values. This suggests that, for the smoothing spline, we can calculate an effective number of degrees of freedom as:

$$\text{edf}_\lambda = \text{trace}\, S_\lambda. \tag{5.5}$$

It can be shown from the limiting behaviour of the smoothing splines as $\lambda \to \infty$ and $\lambda \to 0$ that $\text{edf}_\infty = \nu$ (the number of parameters in the OLS solution) and $\text{edf}_0 = n$ (the number of parameters in the interpolating spline).

5.6 Generalized Cross Validation

The cross-validation criterion $Q_{OCV}(\lambda)$, defined in Equation 5.2, is used to set the smoothing parameter, λ. However, as noted in Section 5.3, using Equation 5.2 to compute $Q_{OCV}(\lambda)$ would be impractical for large n, since it would require fitting a new smoothing spline $\hat{f}_{\lambda,-j}$ for each *leave-one-out* dataset I_{-j}.

Fortunately, it is possible to compute $Q_{OCV}(\lambda)$ directly from the spline $\hat{f}_\lambda$ fitted to the full dataset. It can be shown that Equation 5.2 can be rewritten:

$$Q_{OCV}(\lambda) = \frac{1}{n}\sum_{j=1}^{n}\left(\frac{y_j - \hat{f}_\lambda(t_j)}{1 - s_{jj}}\right)^2, \tag{5.6}$$

where $\hat{f}_\lambda(t_j)$ is the full-data fitted spline value at t_j given by Equation 5.4, and s_{jj} is the jth diagonal element of the smoothing (hat) matrix S_λ.

Prior to the discovery of algorithm to compute Equation 5.6 quickly, a computationally efficient approximation to the cross-validation criterion Equation 5.2 was proposed, called the **Generalized Cross-Validation** criterion (GCV):

$$Q_{GCV}(\lambda) = \frac{\frac{1}{n}\sum_{j=1}^{n}\left(y_j - \hat{f}_\lambda(t_j)\right)^2}{\left(1 - \frac{1}{n}\text{trace}(S_\lambda)\right)^2}. \tag{5.7}$$

Thus Equation 5.7 replaces s_{jj} in Equation 5.6 with the average of the diagonal elements of S_λ, which equals $\frac{1}{n}\text{edf}_\lambda$. Thus a low value of edf_λ will deflate $Q_{GCV}(\lambda)$, making that value of λ more favourable.

In principle, OCV supplants GCV, but GCV is still used as it is numerically more stable. In particular, GCV is used in the `mgcv` package – this will be discussed in the next chapter.

6 General Additive Models

Here is a short video [4 mins] to introduce the chapter.

6.1 Overview

So far, we have considered the modelling of a response variable y in terms of a single response variable x. In particular, we have assumed that the form of this relationship is unknown but can be written as

$$y_i = f(t_i) + \epsilon_i, \qquad \epsilon_i \sim \mathrm{N}(0, \sigma^2)$$

where the ϵ_i are i.i.d. $\sim \mathrm{N}(0, \sigma^2)$ and $f(t)$ is assumed to be smooth. Given knot positions $\{t_i,\ i = 1, \ldots, n\}$, we can estimate $f(t)$ with a smoothing spline $\hat{f}_\lambda(t)$ for given smoothing parameter λ and, further, we can estimate λ using ordinary or generalized cross validation. This approach is in contrast to simple linear regression where y is expressed as a linear function of the explanatory variable $y_i = \alpha + \beta x_i + \epsilon_i$ which enforces a very inflexible relationship.

In this chapter, we generalize the modelling to describe the dependence of the response variable y on a set of explanatory variables $\mathbf{x} = (x_1, x_2, \ldots, x_p)$ where, conditionally on $\mathbf{x}$, observation y has a distribution which is not necessarily normal.

Just as with a generalized linear model, the *general additive model* relates a continuous or discrete response variable Y to a set of explanatory variables $\mathbf{x} = (x_1, x_2, \ldots, x_p)$ and, again, the model contains three parts:

Random part: The probability (mass or density) function of Y is assumed to belong to the *two-parameter exponential family* of distributions with parameters θ and ϕ.

Systematic part: This is a *non-linear predictor equation*:

$$\eta = \sum_{j=1}^{p} f_j(x_j). \tag{6.1}$$

Link function: This is a one-to-one function providing the link between the predictor equation η and the mean $\mu = \mathrm{E}[Y]$:

$$\eta = g(\mu), \quad \text{and} \quad \mu = g^{-1}(\eta) = h(\eta). \tag{6.2}$$

Here, $g(\mu)$ is called the *link function*, and $h(\eta)$ is called the *inverse link function*.

6.2 Penalized deviance

The spline theory in the previous chapters has assumed Gaussian (normally distributed) data and the identity link function. For non-Gaussian data and/or a non-identity link function, we need to replace the penalized least-squares criterion of Equation 4.2 with a *penalized deviance*:

$$R_\nu(f, \lambda, \beta) = D(\mathbf{y}, f, \beta) + \lambda J_\nu(f), \tag{6.3}$$

where $D(\mathbf{y}, f, \beta)$ is the deviance for the vector $\mathbf{y}$ of observations modeled by a linear predictor that comprises a spline function $f(t)$ of order ν and possibly also covariate main-effects and interactions. The penalized deviance is then minimized with respect to the spline coefficients $\mathbf{a}$, $\mathbf{b}$ and regression parameters β, if any. Note that the fitted values for y are obtained by applying the inverse link function to the linear predictor $f(t)$, for example the logistic function when the link is logit.

When there are several smooth terms of order ν in the model, $\mathbf{f} = \{f_1, \ldots, f_m\}$, each may be assigned its own roughness penalty, λ_h, and Equation 6.3 becomes

$$R_\nu(f_1, \ldots, f_m, \lambda_1, \ldots, \lambda_m, \beta) = D(\mathbf{y}, f_1, \ldots, f_m, \beta) + \sum_{h=1}^{m} \lambda_h J_\nu(f_h) \tag{6.4}$$

or we may choose to write this as

$$R_\nu(\mathbf{f}, \lambda, \beta) = D(\mathbf{y}, \mathbf{f}, \beta) + \sum_{h=1}^{m} \lambda_h J_\nu(f_h).$$

6.3 GAMs in R

Fitting smoothing splines is straightforward in practice using R. At the beginning of each **R** session, the first step is to load the package `mgcv` which makes available a set of routines written by Simon Wood of the University of Bath.

The main command is `gam` which fits a smoothing spline (or, more generally, a general additive model). The `gam` function is an extension to the `glm` command for fitting generalized linear models, to allow nonparametric functions of explanatory variables.

The syntax of the `gam` command is similar to that of `glm`. Suppose `y` is a vector of length `n` containing observations of a dependent variable and `tt` is another vector of length `n` containing the times of those observations. Then each of the commands

fits a cubic smoothing spline to the dependent variable `y`. In the first version above, the user explicitly sets the smoothing parameter λ (here denoted `sp`) to the value 3.5. In the second version, the optimal value of λ is chosen by the routine to minimize the Generalized Cross-Validation criterion, GCV. The notation `s(tt)` means a smooth function (a cubic

smoothing spline in this setting) of the explanatory variable `tt`. This notation can be viewed as an extension of the model notation used by the `glm` function. Setting `fx=FALSE` in function `s` specifies that the dimensionality of the spline should be free (see later notes). Parameter `k` of function `s` specifies the *maximum* dimensionality of the spline, **and should be set according to the problem and data at hand** since the default value will not be appropriate in general. For example, if `tt` contains only 6 distinct values, then it would be appropriate to set `k=6`.

In each of the above examples, output from `gam` is stored in an object called `out`. This object contains several components of interest:

- `out$fitted.values` is a vector of length `n` containing the fitted values of the smoothing spline at the data values.
- `out$gcv.ubre` contains the value of the GCV criterion.
- `out$hat` contains the diagonal values of the smoothing matrix, also called the *hat* matrix. The total degrees of freedom in the model (including the intercept term) is given by `sum(out$hat)`.
- `out$sp` contains the smoothing parameter value.
- `summary.gam(out)` provides a summary of the smoothed model fit.
- `anova.gam(out)` provides an analysis of deviance for the model.

Thus, if `gam` is called without an explicit choice of `sp` (as in the second example above), the output from `gam` gives the optimal smoothing parameter value and the corresponding value of the GCV criterion.

To plot a smoothing spline:

- define a dense set of times spanning the data, at which to plot the smooth curve;
- use the `predict.gam` function to compute the cubic spline at these values.

For example, suppose `y` is a vector of length 20 containing the responses at times $1, \ldots, 20$.

6.4 Coronary heart disease (CHD) in South Africa

The textbook *Elements of Statistical Learning*, by Hastie, Tibshirani and Friedman (2nd Edn, 2011), refers to a case–control study of coronary heart disease (CHD) in South Africa.

Variable	Description
`tobacco`	cumulative tobacco consumption (kg)
`famhist`	family history of heart disease (Present, Absent)
`age`	age at onset of the disease (years)
`chd`	case–control status (1 ⇒ CHD; 0 ⇒ no CHD).

The dependent variable in our models will be `chd`. We can consider the CHD status of each individual to be the result of an *experiment* in which the outcome is either CHD (*success*)

or no CHD (*failure*). Thus we can model these data using the Binomial distribution, where the Binomial index is 1. The `glm` function allows such binary (0/1) dependent variables to be specified directly in the model formula, as in the example below, instead of via the usual two-column matrix of successes and failures.

We can examine CHD in relation to tobacco consumption, age and family history of heart disease, with the following **R** commands:

```
Call:
glm(formula = "chd~tobacco+age+famhist", family = "binomial")

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)    -3.620593   0.444576  -8.144 3.83e-16 ***
tobacco         0.083004   0.025712   3.228  0.00125 **
age             0.048812   0.009452   5.164 2.42e-07 ***
famhistPresent  0.974791   0.220023   4.430 9.41e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 596.11  on 461  degrees of freedom
Residual deviance: 495.39  on 458  degrees of freedom
AIC: 503.39

Number of Fisher Scoring iterations: 4
```

All variables in this model are statistically significant: disease is positively related to the amount of tobacco consumed, age and family history of heart disease. However, this model assumes that the logit of the probability of disease is *linearly* related to both tobacco consumption and age (logit being the default link for Binomial). We can explore more flexible tobacco-consumption and age trends with the following generalized additive model:

```
Loading required package: nlme

This is mgcv 1.9-0. For overview type 'help("mgcv-package")'.


Family: binomial
Link function: logit

Formula:
```

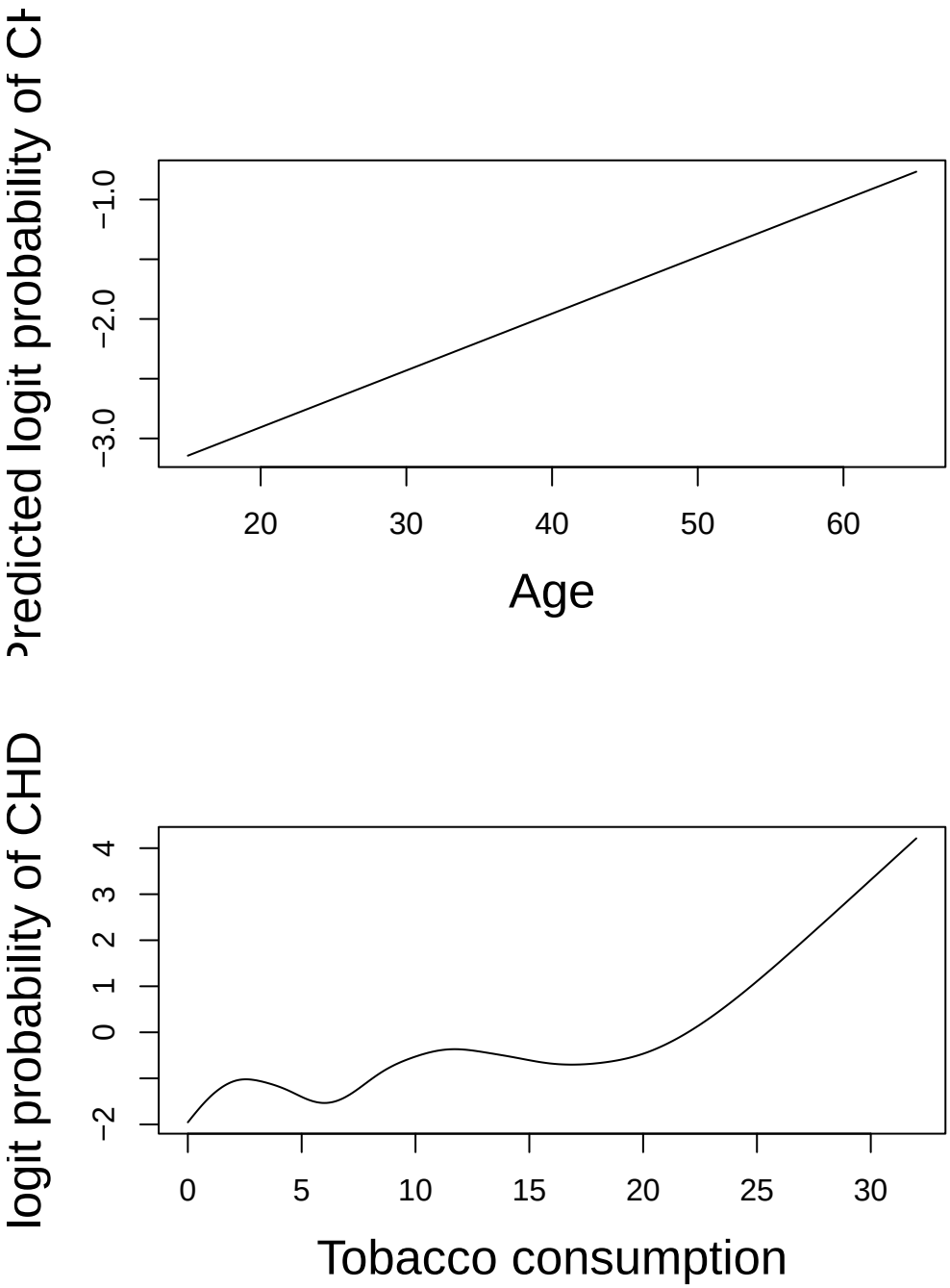
```
chd ~ s(tobacco, k = 20) + s(age, k = 20) + famhist

Parametric coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)     -1.2379     0.1631  -7.592 3.15e-14 ***
famhistPresent   0.9628     0.2233   4.311 1.62e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
             edf Ref.df Chi.sq  p-value
s(tobacco) 6.080  7.573  17.89   0.0179 *
s(age)     1.002  1.003  24.11 9.53e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.212   Deviance explained = 19.1%
UBRE = 0.083268  Scale est. = 1         n = 462
```

Here, the `s` function specifies a smooth (cubic spline) dependence. Parameter `k` of the `s` function specifies the maximum number of degrees of freedom to be allocated to this dependence. Setting `k` too small would restrict the set of basis functions used to construct the splines; setting `k` too large would increase the computational burden unnecessarily.

These results show that the fitted smoothing spline of the dependence of CHD on tobacco consumption has an *effective degrees of freedom* (edf) of 6.080, while the dependence on age has edf of only 1.002, implying an almost linear age-dependence because a linear age term would have exactly 1 degree of freedom. See Section Section 5.5 for further details on the calculation of edf.

The significance of each of these smooth terms is given by the p-value column in the above table. These p-values are computed from the χ^2 statistics in the previous column whose approximate degrees of freedom are given in the column headed `Ref.df`. For example, the p-value of 0.0196 for `s(tobacco)` is computed by referring 17.62 to the χ^2 distribution on 7.573 degrees of freedom. Note that this compares the above model with the model which omits `tobacco` completely.

The following **R** code was used to plot the fitted smooth functions of age and tobacco consumption:



The predicted dependence of the logit probability of CHD on smooth functions of age and tobacco consumption is shown above. We see an almost linear predicted age-dependence, in agreement with its edf. The curious predicted dependence on tobacco consumption might be explained by other factors correlated with tobacco consumption.