Unimodal Spline Regression and Its Use in Various Applications with Single or Multiple Modes

Dissertation

by

Claudia Köllmann

Submitted to the Faculty of Statistics of the TU Dortmund University

in Partial Fulfilment of the Requirements for the Degree of

Doktor der Naturwissenschaften

Dortmund, June 2016

Referees:

Prof. Dr. Katja Ickstadt

Prof. Dr. Roland Fried

Date of Oral Examination: September 9, 2016

Abstract

Research in the field of non-parametric shape constrained regression has been extensive and there is need for such methods in various application areas, since shape constraints can reflect prior knowledge about the underlying relationship. This thesis develops semi-parametric spline regression approaches to unimodal regression.
However, the prior knowledge in different applications is also of increasing complexity and data shapes may vary from few to plenty of modes and from piecewise unimodal to accumulations of identically or diversely shaped unimodal functions. Thus, we also go beyond unimodal regression in this thesis and propose to capture multimodality by employing piecewise unimodal regression or deconvolution models based on unimodal peak shapes.
More explicitly, this thesis proposes unimodal spline regression methods that make use of Bernstein-Schoenberg-splines and their shape preservation property. To achieve unimodal and smooth solutions we use penalized splines, and extend the penalized spline approach towards penalizing against general parametric functions, instead of using just difference penalties. For tuning parameter selection under a unimodality constraint a restricted maximum likelihood and an alternative Bayesian approach for unimodal regression are developed. We compare the proposed methodologies to other common approaches in a simulation study and apply it to a dose-response data set. All results suggest that the unimodality constraint or the combination of unimodality and a penalty can substantially improve estimation of the functional relationship.
A common feature of the approaches to multimodal regression is that the response variable is modelled using several unimodal spline regressions. This thesis examines mixture models of unimodal regressions, piecewise unimodal regression and deconvolution models with identical or diverse unimodal peak shapes. The usefulness of these extensions of unimodal regression is demonstrated by applying them to data sets from three different application areas: marine biology, astroparticle physics and breath gas analysis.
The proposed methodologies are implemented in the statistical software environment R and the implementations and their usage are explained in this thesis as well.

Contents

List of Figures

Notations and abbreviations

Notations

$\mathbb{1}$: indicator function, i.e., $\mathbb{1}_{\mathcal{A}}(x) = \begin{cases} 1, & x \in \mathcal{A} \\ 0, & x \notin \mathcal{A} \end{cases}$

$\mathbf{0}_L$: zero-vector of length L

$\mathbf{1}_L$: one-vector of length L

$\boldsymbol{I}_L$: identity matrix of dimension $L \times L$

Δ^q: differencing operator of order q

$\mathcal{V}$: Bernstein-Schoenberg operator

η_k: space of spline functions of degree k

$\mathcal{C}^q$: space of q-times continuously differentiable functions

$\mathcal{P}_k$: space of polynomial functions of degree smaller or equal to k

$[a, b]^n$: n-fold Cartesian product of the interval $[a, b]$

$|\mathcal{M}|$: Cardinality of the set $\mathcal{M}$

$\mathcal{N}_{\mathcal{M}}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$: Multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ truncated to the set $\mathcal{M} \subset \mathbb{R}^d$

In general, lower-case letters (e.g., y) represent real numbers, that is, constants, observations or indices, while their bold counterparts (e.g., $\boldsymbol{y}$) stand for vectors and bold capital letters (e.g., $\boldsymbol{Y}$) for matrices of such numbers. Random variables are indicated by capital letters (e.g., Y) and random vectors by bold, calligraphic letters (e.g., $\boldsymbol{\mathcal{Y}}$). When taking the Bayesian perspective, especially, when parameters or parameter vectors are considered random, we will not distinguish between the observation and its random

counterpart for convenience. In addition, a simplified notation for probability densities, which is commonly used in Bayesian statistics, will be employed. For example, the notation $p(\boldsymbol{\beta}|\boldsymbol{y})$ stands for the conditional density $p(\boldsymbol{\beta} = \boldsymbol{\beta}|\boldsymbol{\mathcal{Y}} = \boldsymbol{y})$, where $\boldsymbol{\beta}$ is the random counterpart of $\boldsymbol{\beta}$.

Abbreviations

ADF Average daily feed consumption

ADG Average daily gain of weight

AIC Akaike information criterion

ASE Average squared error

BIC Bayesian information criterion

B-S Bernstein-Schoenberg

ed Effective Dimension

FACT First G-APD Cherenkov telescope

G-APD Geiger-mode avalanche photodiods

G/F Gain-to-feed ratio

IMS Ion mobility spectrometry

IRLS Iteratively re-weighted least squares

MCC Multi capillary columns

MCMC Markov Chain Monte Carlo

MCR Multivariate curve resolution

MRL Mean relative loss

PST Porcine somatotropine

REML Restricted maximum likelihood

RSS Residual sum of squares

sigE Sigmoid E_{max}

TDR Time-depth-recorder

TPF Truncated power function

1 Introduction

1.1 Motivation

In statistical modelling, many approaches aim at describing the way in which some aspect in life changes depending on one or more variables in its environment. For example, the diving depth of a marine animal during a dive is first monotone increasing and then monotone decreasing over time, see Figure 1.1A. In explicit, there is a unimodal dependence on time.

Establishing a relationship between a dependent variable (response) and one or several independent variables (predictors) is called regression analysis. Usually, these variables are observed on the units of a certain population and the interest is in the mean of the response Y given the values $\boldsymbol{x}$ of the predictors $\boldsymbol{\mathcal{X}}$. In regression analysis the mean is modelled by a function f of the predictor values:

$$E(Y|\boldsymbol{\mathcal{X}} = \boldsymbol{x}) = f(\boldsymbol{x}),$$

where f is usually depending on a vector $\boldsymbol{\theta}$ of parameters, $f(\boldsymbol{x}) = f(\boldsymbol{x}|\boldsymbol{\theta})$, to be estimated.

How many and which predictors to choose for a multiple regression model, is studied under the subject of variable selection. In this thesis however, the focus is on univariate regression.

For chosen variables X and Y there are, of course, still infinitely many possibilities to specify f: first of all, one can assume a simple linear relationship as well as polynomial functions, both of which can be estimated from data using linear model theory. If the function is non-linear in the parameters, non-linear optimization algorithms can be used to estimate $\boldsymbol{\theta}$ (see, e.g., the book by Seber and Wild, 2003, on non-linear regression techniques). In these cases, there are usually few parameters, since the gain from using higher polynomial degrees is small and non-linear models with many parameters are computationally hard to estimate. To sum up, parametric approaches are quite restric-

tive regarding the shapes of the function f that can be modelled. In other words, they pose so many assumptions about the underlying relationship that, if the true relationship deviates from them, a lot of bias is introduced. On the other hand, if the assumptions are justified, the variance of the estimated relationship between different samples is quite low. In pharmaceutical dose-response trials, for example, the functional form f used for the analysis has to be pre-specified in the study protocol (before data collection). This is very difficult and practical methods often rely on specification of a candidate set of parametric dose-response models (see, e.g., Bretz et al., 2005) and on model selection or model averaging.

Competitors to these procedures are regression approaches which do not specify the functional form of f by a set of parameters and are hence called non-parametric. They often rely on local estimation approaches, where no closed-form representation of f exists, for example, kernel smoothers such as nearest neighbour estimators or local polynomial regression (see, e.g., Chapter 6 of Hastie et al., 2009, for an introduction). These methods are – contradictory to the term "non-parametric" – also said to have infinitely many parameters. They are very flexible and usually have a low estimation bias, but the estimated relationships may vary strongly between different samples. Additionally, the estimated functions f are not necessarily differentiable or even discontinuous.

Semi-parametric approaches can be thought of as a compromise and are often characterized by many parameters. In addition, they usually describe the form of f as a linear combination of basis functions, for example, radial or B-spline basis functions (see also Figure 1.1B and C for an illustration of the latter). The coefficients of the linear combination are the parameters to be estimated.

The B-spline basis induces the popular class of polynomial spline functions, which are smooth piecewise polynomials. Using a large basis (many parameters) yields very flexible functions, which are continuous and differentiable up to a known degree. The function itself and its derivatives have a closed-form representation. Nevertheless, f can be estimated in terms of a linear model due to the linear combination of basis functions. Again, the flexibility reduces bias, but comes at the price of higher variability across samples from the same population.

The task of reducing the variance of the estimators without increasing the estimation bias has been tackled in two popular ways:

1. introduction of a smoothness penalty, dating back to the smoothing spline approaches by Reinsch (1967, 1971),

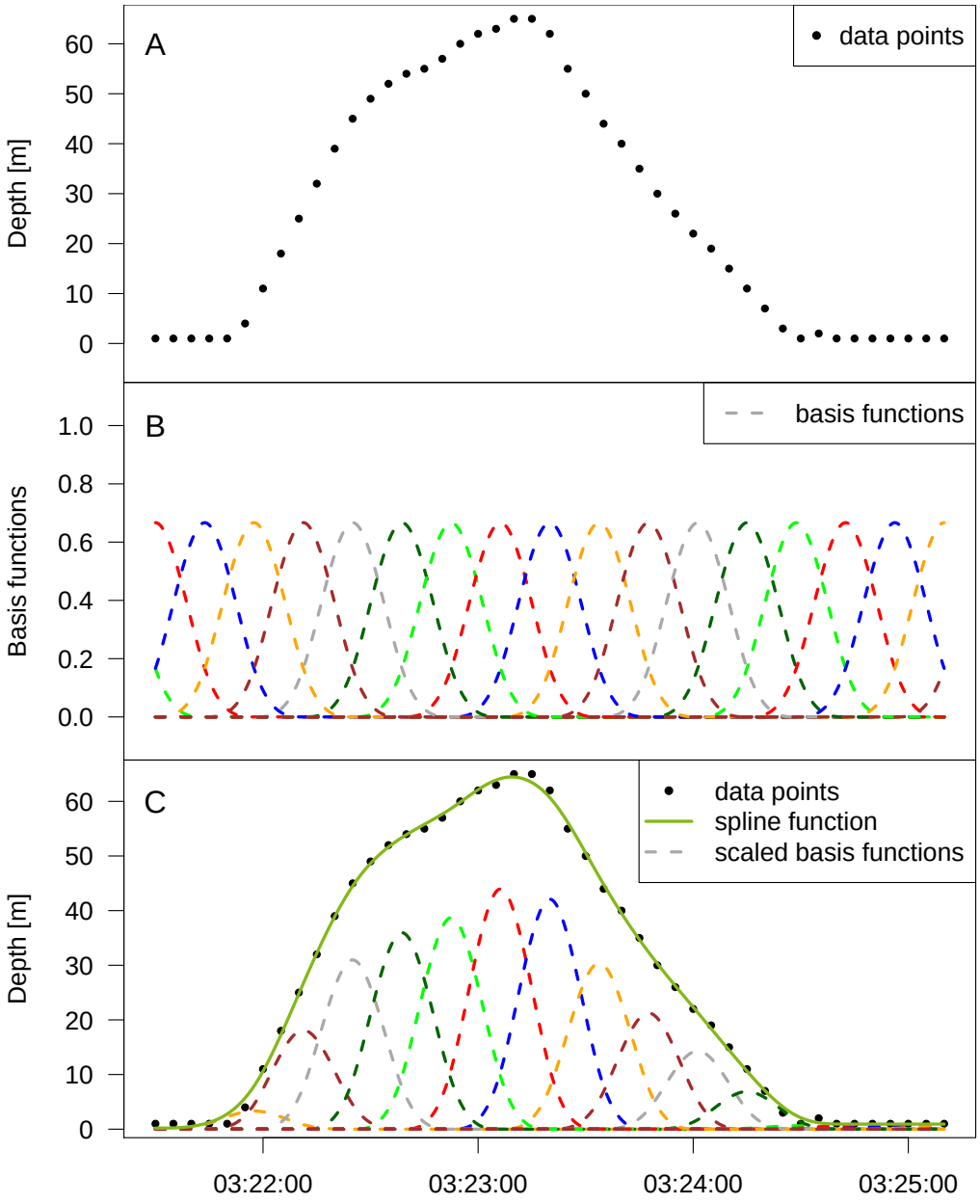


Figure 1.1: **Dive of a marine animal with spline basis and fitted spline.** (A) Scatterplot of the diving depth [in m] of a marine animal versus time (between 03:21:30 a.m. and 03:25:10 a.m. on January 6th 2002). It is an excerpt from the data set `divesTDR` from R package `diveMove` (version 1.4.1, cf. Luque, 2007). (B) B-spline basis functions. (C) Data from (A) and a fitted spline function, which is the sum of the corresponding scaled B-spline basis functions. In explicit, the spline is a linear combination of the B-spline basis functions.

2. introduction of a shape constraint, dating back to constrained estimation approaches from the 1950s (Brunk, 1955; Hildreth, 1954) and to the "pool adjacent violators algorithm" by Barlow et al. (1972) for pointwise monotone regression.

The first approach penalizes functions that are too variable and aims at finding a compromise between over- and underfitting, between small bias and small variance. The second approach reduces the function space from which f is taken, which also decreases the variability. Shape constraints such as monotonicity or unimodality do not restrict the function space as severely as polynomial or non-linear parametric models and, therefore, the chance of violated assumptions and thus increased bias is smaller in those situations. In addition, there are many applications in which the assumption of a certain shape constraint, such as positivity, monotonicity or unimodality, is very plausible and the incorporation of this prior knowledge into the model can only be advantageous. Regarding the diving depth of a marine animal, for example, the assumption of a unimodal shape with respect to time seems likely, while something like a quadratic relationship is less easily justified.

1.2 Aims and outline

This thesis will consider the use of smoothness penalties and shape constraints as well as their combination in univariate regression. The focus will be on the shape constraint of unimodality, which has received less attention in the literature so far.
Unimodal regression – as a type of non-parametric shape-constrained regression – is a suitable choice in regression problems when the prior information about the underlying relationship between predictor and response is vague, but when it is (almost) certain that the response variable first increases with higher values of the predictor variable up to a maximum (or mode) and then decreases again.
While there exist a variety of parametric approaches to estimate a unimodal relationship, this thesis introduces a flexible semi-parametric method for estimating a smooth unimodal function based on spline functions. For this purpose, spline functions will be shown to be particularly well-suited for shape-constrained function estimation.
A prominent application, which will serve as an example several times during this thesis, is dose-response analysis. Here, the (beneficial) effect of a substance increases with increasing dose up to a saturation point, after which the effect starts to decrease again, as the substance might cause, for example, interfering toxic effects.
However, the prior knowledge in different applications has various degrees of complexity

since data shapes may vary from (piecewise) unimodal relationships to accumulations of identically or even diversely shaped unimodal functions. This thesis argues that unimodal regression is also useful in situations where the relationship between two variables is not unimodal, but multimodal, in explicit, the function f has several modes (local maxima). Therefore, this thesis also goes beyond unimodal regression and proposes to model multimodality using several unimodal functions.

The outline of this thesis is as follows: In Chapter 2 data sets from different areas are introduced to further motivate the need for unimodal regression and its multimodal extensions in real applications.The examples stem from dose-response analysis, marine biology, astroparticle physics and breath gas analysis.
Chapter 3 introduces regression splines and their smoothness-penalized pendants and highlights the benefits for regression purposes, especially in the presence of a shape constraint such as unimodality. A frequentist as well as a Bayesian approach to unimodal spline regression are developed. The aim of Chapter 4 is to present methodology, which enables the handling of a broad spectrum of applications with multimodal data. Thus, several approaches, extending the methodology of Chapter 3, are proposed and recommendations on the method of choice in different situations are given. Both Chapters 3 and 4 provide a literature overview within the respective branch of research.
Chapter 5 introduces an R package which implements the frequentist unimodal regression approach, and also describes implementational details of the other methods for unimodal and multimodal regression.
The performance of the proposed unimodal spline regression approaches in comparison to competing methods is assessed in Chapter 6 with an extensive simulation study in the dose-response analysis context. The question, if a combination of shape constraint and penalization is beneficial or if one of them suffices, is addressed there, too.
The usefulness of the methods in practice is demonstrated in Chapter 7 by applying the methodology to the real data examples from Chapter 2. The applications are increasing in complexity as they vary from unimodal or piecewise unimodal relationships to convolutions of identically or even diversely shaped unimodal functions.
Chapter 8 summarises the thesis and provides an outlook on further extensibility of the presented methodology and future research objectives.

2 Fields of application and data material

This chapter describes application areas where unimodal regression can be useful and provides details on the respective data sets that are analysed throughout this thesis. The fields of application are very diverse as the data sets stem from dose-response trials, marine biology, astroparticle physics and breath gas analysis.

2.1 Growth hormone dose response data

The first field of application is dose-response analysis, which was also discussed in Köllmann et al. (2014).
Characterization of the dose-response relationship for desirable and undesirable effects of a pharmaceutical compound is the central problem of its clinical development. The pre-specification of one dose-response model for analysis in the study protocol (before data collection) is difficult, which is why practical methods often rely on specification of a candidate set of parametric dose-response models (see, e.g., Bretz et al., 2005) and on model selection or model averaging. Unimodal regression is a non-parametric competitor to these techniques.
A typical assumption in parametric as well as non-parametric dose-response analyses is monotonicity. However, in a variety of cases this assumption can be challenged as the interference of potential saturation or toxicity effects cannot be excluded. When considering a clinical utility index that combines efficacy and safety measures (see e.g. Khan et al., 2009) one explicitly expects a unimodal relationship and a monotonically increasing curve would be surprising to observe. But even when considering efficacy alone unimodality can occur as in the example to follow. A unimodal shape constraint relaxes the assumption of monotonicity and is adequate whenever an umbrella dose-response curve cannot be excluded a priori.
The example data set originates from animal science, where the growth of pigs is eval-

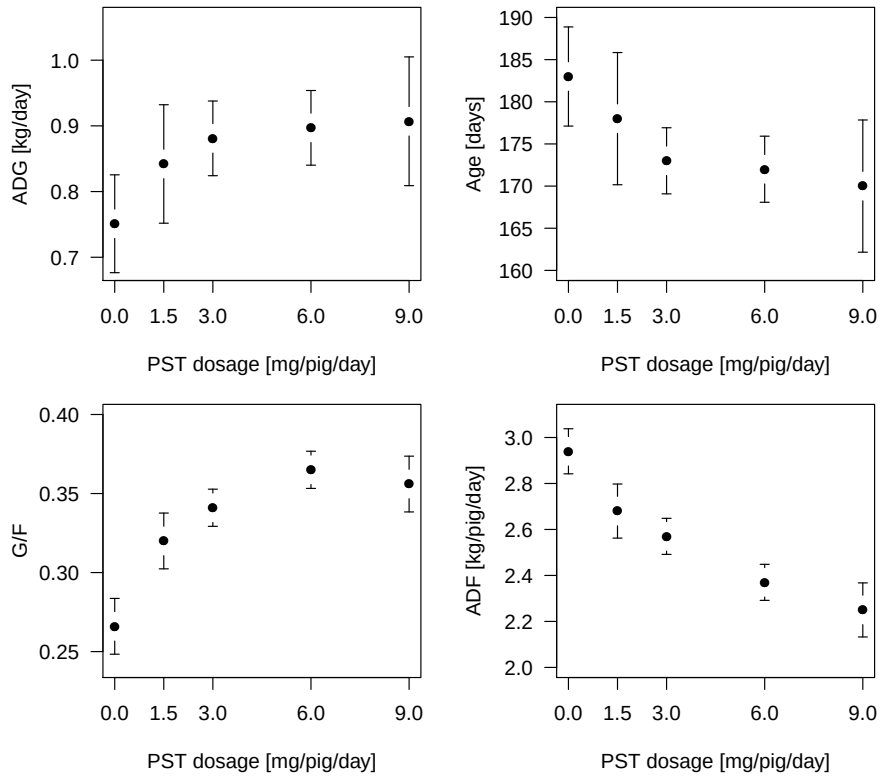


Figure 2.1: **Scatterplots of PST dosage vs. means of the response variables.** The standard errors at each dose are indicated with bars. Data source: McLaren et al. (1990).

uated in dependence of an increasing dose of a growth hormone. McLaren et al. (1990) investigated the relationship between administration of porcine somatotropin and several growth variables in 195 pigs. Details on the experimental procedure and data preprocessing can be found in their article. The (aggregated) data used here are the porcine somatotropin dosage levels [mg/pig/day] (PST) and the least squares means and standard deviations of four response variables: Average daily gain of weight [kg/day] (ADG), age at 103.5 kg [days] (Age), gain-to-feed ratio (G/F) and average daily feed consumption [kg/pig/day] (ADF). The five dosage levels are 0, 1.5, 3, 6, 9 mg/pig/day and the means and standard errors at the respective levels correspond to 29, 29, 57, 58, and 22 pigs. The data are plotted in Figure 2.1 and the actual data values can be found in Table 2.1.

While the modes of the means of ADG, Age and ADF are at extreme doses (suggesting monotone relationships), the means of G/F have their mode in the interior at dose 6. Since monotonicity is a special case of unimodality, it seems reasonable to relax the

monotonicity assumption and apply unimodal regression to all four variables (inverse unimodal for the variables Age and ADF). The results are presented in Section 7.1.

Table 2.1: **Porcine somatotropin (PST) dosages [mg/pig/day] and least squares means and standard errors of the four performance variables.** ADG = average daily gain of weight [kg/d], Age = age at 103.5 kg [d], G/F = gain-to-feed ratio, ADF = average daily feed consumption [kg/pig/d]. Data source: McLaren et al. (1990).

	PST dosage				
	0	1.5	3	6	9
ADG	0.751 (0.038)	0.842 (0.046)	0.881 (0.029)	0.897 (0.029)	0.907 (0.050)
Age	183 (3)	178 (4)	173 (2)	172 (2)	170 (4)
G/F	0.266 (0.009)	0.320 (0.009)	0.341 (0.006)	0.365 (0.006)	0.356 (0.009)
ADF	2.940 (0.050)	2.680 (0.060)	2.570 (0.040)	2.370 (0.040)	2.250 (0.060)

2.2 Analysis of dive phases of marine animals

The second application example is the analysis of diving behaviour of marine animals (see also Köllmann et al., 2016). Time-depth-recorders (TDRs) measure the diving depth of marine animals such as seals or whales. The resulting data sets may contain measurements over several days at regular sampling frequencies. In the case of marine mammals, the animals repeatedly perform dives from the water surface down to various depths to find food and for other activities. An excerpt from such a TDR data set, taken from the R package `diveMove` (version 1.4.1, cf. Luque, 2007; Luque and Fried, 2011), is shown in Figure 2.2A.

Marine biologists are interested, among other things, in the detection of phases within a dive which correspond to different behaviours (see e.g. Halsey et al., 2007). As claimed by Halsey et al. (2007) there is need for objective and automated categorisation of the diving behaviour and they develop a Matlab program that classifies diving depth data using a set of pre-specified criteria. This approach does not consider measurement error, which can be accounted for by modelling the dives statistically. This is, for example, realized in version 1.4.1 of the R package `diveMove` by fitting multiple smoothing splines to the data. Afterwards, the derivative of the fitted splines is used to divide the dives into different phases like descent and ascent. In Section 7.2.1 we show that using piecewise

unimodal regression splines is advantageous for this purpose. Since the animal definitely needs to come back to the surface to draw breath, in explicit, since the dives do not overlap, the time series can be modelled by piecewise unimodality.

2.3 Astroparticle physics data analysis

The third field of application is astroparticle physics, which was also presented in Köllmann et al. (2016). The First G-APD Cherenkov Telescope (FACT; see Anderhub et al., 2013; Biland et al., 2014) is used by astroparticle physicists to detect cosmic rays. These cosmic rays induce light flashes in the earth's atmosphere, which can be used to calculate the primary particle's properties. The camera of the telescope has several pixels and each pixel collects a signal, that is, a time series of measured voltages. See Figure 2.2B for an example with 250 observations.
Each photon hitting a camera pixel causes a change in the signal, which can be described by a unimodal loading curve with an amplitude of approximately 10 mV (Anderhub et al., 2013). The aim is to detect the arrival times and numbers of photons to draw conclusions about the type of the triggering particle (gamma or hadron). A good overall fit is of interest, too, since the integral over the signal is used in subsequent analyses. The shape of the signal is similar to that of a loading and unloading condenser and thus, physicists have suggested a parametric wave form for the change in the voltage due to the arrival of one or more photons. When n_p photons arrive at time t_0 this wave form is given by

$$U(t) = \gamma + n_p \cdot U_0 \cdot \left(1 - e^{-\frac{t-t_0}{\xi_1}}\right) e^{-\frac{t-t_0}{\xi_2}} \mathbb{1}_{[t_0,\infty)}(t), \tag{2.1}$$

where γ is the baseline voltage shortly before the photons' arrival (cf. Buß, 2013, formula 6.11). The remaining parameters U_0, ξ_1, ξ_2 specifying the wave form do not have a concrete physical interpretation.
Since the telescope has been constructed quite recently, standard methods for the evaluation of measured signals are mostly heuristic and only applied on segments of a signal. Parameter estimates for waves of the form (2.1) from well-distinguished signals of *single* photons ($n_p = 1$) were derived in Buß (2013). As photons can arrive anytime, the measured voltage is a convolution of several loading curves. This suggests using a deconvolution model with accumulated unimodal parametric waves for the analysis of a whole time series of one pixel.

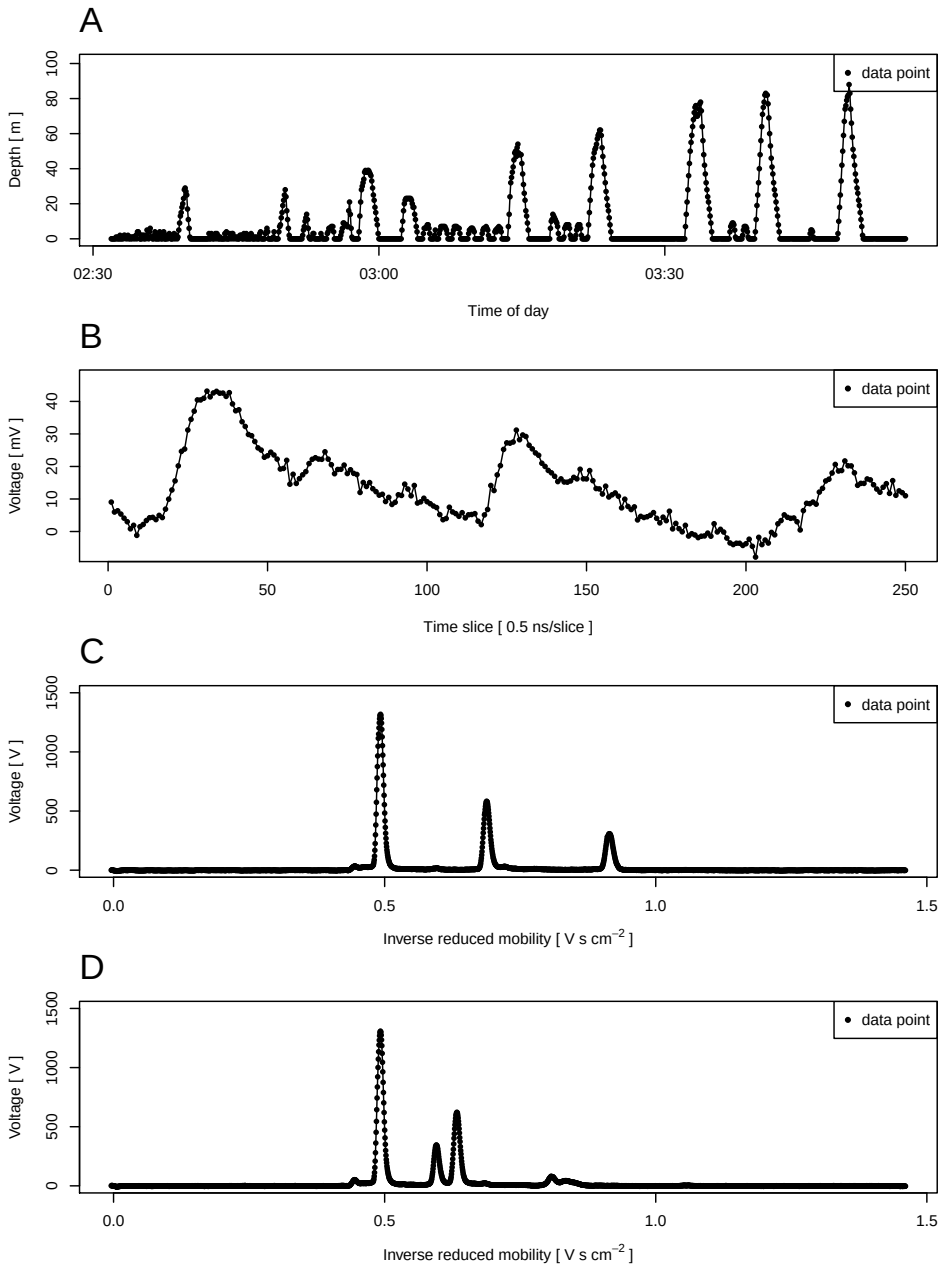


Figure 2.2: **Multimodal example data sets used throughout the thesis.** (A) shows an excerpt from data set `divesTDR` (R package `diveMove`, version 1.4.1, Luque, 2007). It displays the diving depth [in m] of a marine animal, which was recorded every five seconds between 02:31:55 a.m. and 03:55:15 a.m. on January 6th 2002. (B) is an example of a FACT time series of length 250. The x-axis is the number of the time slice or sample, where the slices are about 0.5 ns wide. The y-axis gives the voltages measured in millivolt [mV]. (C) and (D) show spectrum A and B of the IMS example data set. The x-axis is inverse reduced mobility [V s cm^{-2}], a transformation of drift time. The y-axis gives the measured voltages in volt [V]. Both spectra have 2499 observations.

2.4 Breath gas analysis with ion mobility spectrometry

The last application area presented in this thesis is breath gas analysis (see also Köllmann et al., 2016), where ion mobility spectrometry (IMS) coupled with multicapillary columns (MCCs) is used to measure the amount of certain molecules in the air or in exhaled breath. Knowledge about the presence of such molecules and their concentrations can be used for medical purposes, for example, to diagnose lung cancer (cf. Westhoff et al., 2009). An IMS-MCC data set is a matrix of measured intensities. Looking only at one row or one column at a time, the intensities are time series along drift time (rows) or along retention time (columns). In this thesis we focus only on the observed intensities along the drift time, which are called spectrum. Typically, the intensities in a spectrum fluctuate around zero and exhibit few peaks, see also Figures 2.2C and 2.2D. At least one peak is always present at about 0.5 and does not carry information about the analyte: the so-called reaction ion peak. The other peak locations and their amplitudes provide information about the presence of different molecule types. Since the IMS technology is getting more and more miniaturized and feasible for mobile use, there is also need for suitable analysis methods, which a) work automatically, in explicit, without an expert, and b) process the data online (e.g. analysing one spectrum during measurement of the next).

One of the main steps in IMS analysis is peak extraction, that is, data reduction in a way that every peak is described by a number of parameters, which reproduce the characteristics of a spectrum as closely as possible, for example, location and amplitude of the peaks. Currently, this is done manually by experts using interactive visualization software (Kopczynski et al., 2012; D'Addario et al., 2014). Subsequent analyses use the parameters of all spectra to identify different molecules in the analyte or to classify samples into subgroups like healthy or not (see also Hauschild et al., 2013). In this thesis we only focus on the peak extraction step and aim at modelling a single spectrum at a fixed retention time statistically.

For whole IMS-MCC data matrices Kopczynski et al. (2012) used a 2-dimensional mixture model with a background component and several peak components, where each of the latter ones is based on the product of two shifted inverse Gaussian distributions, one for drift time and one for retention time. In a more recent approach Kopczynski and Rahmann (2014) again use shifted inverse Gaussian distributions to model each peak in one spectrum (row) of IMS-MCC data sets. This means that a peak is modelled with

only a small number of parameters in both cases. Other approaches for modelling IMS or similar data (see, e.g. Vogtland and Baumbach, 2009; Bödeker and Baumbach, 2009; Rossoni and Feng, 2006) use different distributions, but also with very few parameters. Although one of the aims is data reduction and the distributions allow, for example, for skewness, this representation of the data might be too restrictive regarding the shape of a peak. Since it is known that each peak is a unimodal function of the drift time, we propose instead to describe a spectrum with multiple unimodal spline functions. Thus we will consider piecewise unimodal regression and deconvolution models with unimodal peak shapes for this application.

3 Unimodal spline regression

This chapter is one of the two main methodological chapters of this thesis and presents frequentist and Bayesian spline regression approaches to unimodal regression. Section 3.1 provides an overview of existing approaches to shape-constrained regression, especially unimodal regression. The methodological part starts in Section 3.2 with an introduction to spline functions and discusses their approximation power as well as their use in regression settings. Section 3.3 then describes penalized estimation with different smoothness penalties. Regarding the form of the penalty, which can also be viewed as prior information from the Bayesian perspective, we first introduce the finite difference penalties by Eilers and Marx (1996). In a second step, we propose a novel approach to incorporate prior information by penalizing against a parametric model, while still allowing for departures from it, when the data suggest so. In Section 3.4 the shape-preservation properties of splines are discussed using the characteristics of Bernstein-Schoenberg splines and it is shown how to incorporate shape constraints such as unimodality. Sections 3.5 and 3.6 present the main parts of the methodology for unimodal regression: a frequentist and a Bayesian approach towards estimation of a penalized unimodal spline regression, in explicit, the combination of penalized and shape-constrained spline regression. Section 3.7 introduces weighted spline regression and its use in iterative algorithms for robust estimation of a unimodal spline regression.
The article Köllmann et al. (2014) is based on parts of the material in this chapter.

3.1 Overview

Most common shape constraints used in the context of splines (and polynomials) are monotonicity, convexity or concavity and log-concavity, because finite dimensional constraints on the spline coefficients ensure the desired shape constraint. See, for example, Ramsay (1988), Kelly and Rice (1990), Wood (1994), Hazelton and Turlach (2011), Meyer et al. (2011) and Wang and Ghosh (2012) for different approaches. Essentially,

these shape constraints induce non-negativity constraints on a derivative which can be ensured using constrained optimization in non-Bayesian approaches (see e.g. Ramsay (1988) or Wang and Ghosh (2012)) and by prior specification in Bayesian approaches (see e.g. Hazelton and Turlach (2011) and Meyer et al. (2011)).
Another alternative to constrained optimization is the use of an asymmetric penalty on the spline coefficients introduced by Eilers (2005). It penalizes those coefficients that violate the shape restriction and is iteratively recalculated. The shape constraint of log-concavity (which also guarantees unimodality), for example, can be achieved by imposing concavity on the logarithm of the response with an asymmetric penalty. Yet, log-concavity is a stronger requirement than unimodality and caution is required if the underlying shape is unimodal, but not log-concave, for example, if the shape is "heavy-tailed".
However, the unimodality constraint does not reduce to a single positivity constraint on a derivative of the modelled function and has received less attention in the spline literature so far. In an unpublished manuscript, Woodworth (1999) uses B-splines with a certain hierarchical prior on the coefficients that guarantees unimodality. Unfortunately, no general suggestions on the choice of the prior parameters and the knot sequence are given. The incorporation of a smoothness penalty, which could be used to address the problem of knot placement, is also not straightforward in this model.
Even beyond splines, the literature on non-parametric estimation of a smooth unimodal function is relatively sparse. The first approaches to unimodal regression date back to the 1980s (see Frisén and Goteborg (1980), Frisén (1986), Hildenbrand and Hildenbrand (1986)), where the proposed methods produced *pointwise* least squares estimates by successively splitting the data into two possible subsets, applying an isotonic and an antitonic regression (see e.g. Barlow et al. (1972)) and choosing the split with minimal sum of squares. Later, several authors proposed algorithms that solve the unimodal least squares regression problem faster (see e.g. Turner and Wollan (1997), Bro and Sidiropoulos (1998), Stout (2008)) or that give unimodal regression estimates minimizing other objective functions such as L_1- or L_∞-metrics (see e.g. Boyarshinov and Magdon-Ismail (2006), Stout (2008)). Since all of these algorithms give pointwise estimates, these have to be interpolated or smoothed in a second step. Similar drawbacks arise in methods for unimodal density estimation. The approaches of Wegmann (1972), Bickel and Fan (1996) and Birge (1997) produce unimodal step-functions by combining two monotone density estimates of Grenander (1956). But even joining two *smooth* monotone regression or density estimates will not help, because it yields estimates which are not smooth

or even discontinuous at the mode. The problem is that the shape constraint is imposed using two local monotonicity constraints instead of one global constraint.
A different, but very general approach to incorporate shape constraints is data sharpening, where the data points are shifted as little as possible so that the unconstrained estimate satisfies the constraint. See, for example, Braun and Hall (2001) for an approach to unimodal density estimation.
In this chapter we present a semi-parametric spline regression approach to unimodal regression. The method is based on the fact that using the B-spline basis, a spline can be restricted to be unimodal by choosing a unimodal sequence of B-spline coefficients with a fixed mode, which in contrast is a global constraint. The use of spline functions guarantees the continuity of the fit. Smoothness can be achieved by using a penalization approach.

3.2 Spline functions and curve fitting

The following introduction to spline functions and their use for regression purposes is based on Dierckx (1993) unless noted otherwise.

3.2.1 The vector space of spline functions and suitable bases

Before discussing the use of spline functions for (shape-constrained) curve fitting, we review their definition and some characteristics.

Definition 1. *A spline function of degree $k > 0$ with knots $\tau_0 < \tau_1 < \ldots < \tau_{g+1}$ is a function $s(x)$, defined on a finite interval $[a, b]$, where $\tau_0 = a, \tau_{g+1} = b$, that satisfies*

(i) $s(x)$ is on each knot interval $[\tau_j, \tau_{j+1}]$ a polynomial of degree $\leq k$, that is

$$s_{|[\tau_j, \tau_{j+1}]} \in \mathcal{P}_k, j = 1, \ldots, g.$$

(ii) $s(x)$ and its $(k-1)$ first derivatives are continuous on $[a, b]$, that is

$$s(x) \in \mathcal{C}^{k-1}[a, b].$$

If the knots are not strictly increasing but there exist coincident knots ($\tau_{j-1} < \tau_j = \ldots = \tau_{j+l} = c < \tau_{j+l+1}$), Definition 1 can be extended to this case by requiring in condition 2.

that only the $(k-1-l)$ first derivatives are continuous at point c.
If the k-th order derivative of a spline is discontinuous at an interior knot $\tau_j,\ j \in \{1,\dots,g\}$, then this knot is called *active*. If a spline $s(x)$ satisfies $s^{(l)}(a) = s^{(l)}(b),\ l = 0,1,\dots,k-1$, then it is called *periodic*. Furthermore, a *natural* spline function is a spline of odd degree $k = 2l-1$ $(l \geq 2)$, which satisfies $s^{(l+j)}(a) = s^{(l+j)}(b),\ j = 0,\dots,l-2$.
A consequence of Definition 1 is that every polynomial on $[a,b]$ of degree $\leq k$ is also a spline function of degree k on $[a,b]$. The other way around, a spline function $s(x)$ can be written with the help of polynomials as

$$s(x) := p_{k,j}(x) = \sum_{i=0}^{k} a_{i,j}(x-\tau_j)^i \text{ if } \tau_j \leq x \leq \tau_{j+1},\ j = 0,\dots,g,$$

where the restrictions $p^{(l)}_{k,j-1}(\tau_j) = p^{(l)}_{k,j}(\tau_j),\ j = 1,\dots,g,\ l = 0,\dots,k-1$ apply for the coefficients $a_{i,j}$ because of the continuity conditions.
Let $\eta_k(\tau_0,\tau_1,\dots,\tau_{g+1})$ denote the vector space of all spline functions of degree k as determined by Definition 1. It follows from the $g \cdot k$ conditions on the coefficients $a_{i,j}$ that the space's cardinality is given by

$$d := |\eta_k(\tau_0,\tau_1,\dots,\tau_{g+1})| = (g+1)(k+1) - gk = g + k + 1.$$

We are now interested in a basis for this vector space, that is, a set of d basis spline functions, so that each element of η_k can uniquely be written as their linear combination.

Definition 2. *A function $f : \mathbb{R} \mapsto \mathbb{R}$ that is given by*

$$f(x) = (x-\phi)_+^k = \begin{cases} (x-\phi)^k, & x \geq \phi \\ 0, & x < \phi, \end{cases}$$

where $\phi \in \mathbb{R}$ is an arbitrary constant, is called truncated power function (TPF).

It is easily verified that a truncated power function is a spline with an active knot at the point ϕ and it can be proven that each member of η_k can uniquely be written in the form

$$s(x) = \sum_{i=0}^{k} b_i x^i + \sum_{i=1}^{g} c_i (x-\tau_i)_+^k,$$

where $b_i \in \mathbb{R},\ i = 0,\dots,k$ and $c_i \in \mathbb{R},\ i = 1,\dots,g$.

For the polynomials in each knot interval it follows that

$$p_{k,j}(x) = \sum_{i=0}^{k} b_i x^i + \sum_{i=1}^{g} c_i (x-\tau_i)_+^k = \sum_{i=0}^{k} b_i x^i + \sum_{i=1}^{j} c_i (x-\tau_i)^k,\ x \in [\tau_j, \tau_{j+1}],\ j = 0, \ldots, g.$$

The $d = g+k+1$ functions $1, x, \ldots, x^k, (x-\tau_1)_+^k, \ldots, (x-\tau_g)_+^k$ form a basis of the spline space η_k. However, this *TPF basis* is numerically ill-conditioned. Thus, the so-called B-spline basis is introduced in the following.

Definition 3. *The k-th divided difference of a function f at (distinct) points $\tau_0, \ldots, \tau_k$ is the leading coefficient (coefficient of x^k) of the unique polynomial $p_k(x)$ which satisfies $p_k(\tau_j) = f(\tau_j),\ j = 0, \ldots, k$. It is denoted by $[\tau_0, \ldots, \tau_k]f$ or $\Delta_x^k(\tau_0, \ldots, \tau_k)f(x)$.*

It directly follows that if f itself is a polynomial of degree $\leq k-1$ ($f \in \mathcal{P}_{k-1}$) through the points $f(\tau_j)$, then the coefficient of x^k is 0 and thus $[\tau_0, \ldots, \tau_k]f = 0$. Other interesting properties of the divided difference are the following:

Symmetry The order of the numbers $\tau_0, \ldots, \tau_k$ does not matter, so they are assumed to be in increasing order.

Linearity If $f(x) = \alpha g(x) + \beta h(x)$ then, $[\tau_0, \ldots, \tau_k]f = \alpha[\tau_0, \ldots, \tau_k]g + \beta[\tau_0, \ldots, \tau_k]h$.

Recursion The following recursion relation holds for the divided differences:

$$[\tau_j, \ldots, \tau_{j+l}]f = \frac{[\tau_{j+1}, \ldots, \tau_{j+l}]f - [\tau_j, \ldots, \tau_{j+l-1}]f}{\tau_{j+l} - \tau_j} \text{ and } [\tau_j]f = f(\tau_j).$$

Figure 3.1 gives a corresponding scheme useful for calculating divided differences.

Newton form The polynomial $p_k(x)$ which satisfies $p_k(\tau_j) = f(\tau_j),\ j = 0, \ldots, k$, can be written with the help of the diagonal elements from Figure 3.1 in the following form:

$$p_k(x) = [\tau_0]f + \sum_{j=1}^{k} (x-\tau_0) \cdots (x-\tau_{j-1})[\tau_0, \ldots, \tau_j]f.$$

Explicit expression The divided differences can also be calculated using the explicit expression $[\tau_0, \ldots, \tau_k]f = \sum_{j=0}^{k} \frac{f(\tau_j)}{\prod_{l=0, l\neq j}^{k} (\tau_j - \tau_l)}$.

Definition 4. *The (normalized) B-spline $N_{j,k+1}$ of degree k with knots $\tau_j, \ldots, \tau_{j+k+1}$ is defined as*

$$N_{j,k+1}(x) = (\tau_{j+k+1} - \tau_j)\Delta_t^{k+1}(\tau_j, \ldots, \tau_{j+k+1})(t-x)_+^k.$$

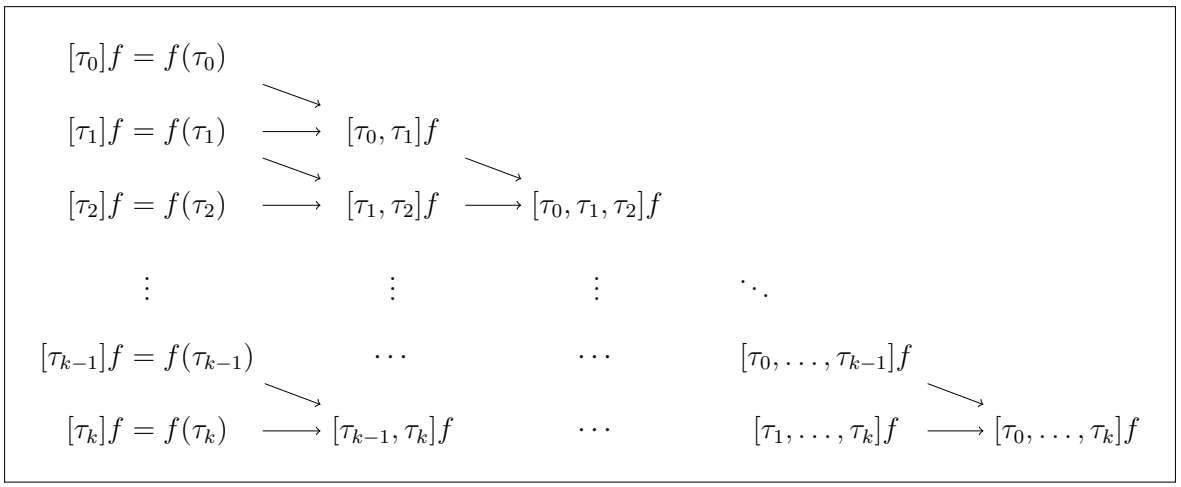


Figure 3.1: **Triangular scheme for the calculation of divided differences.**

With the help of the explicit expression of divided differences this can be written as

$$N_{j,k+1}(x) = (\tau_{j+k+1} - \tau_j) \sum_{i=0}^{k+1} \frac{(\tau_{j+i} - x)_+^k}{\prod_{l=0, l\neq i}^{k+1} (\tau_{j+i} - \tau_{j+l})}.$$

Some resultant properties of the B-splines are:

Positivity $N_{j,k+1}(x) \geq 0 \ \forall x$.

Local support $N_{j,k+1}(x) = 0 \ \forall x \notin [\tau_j, \tau_{j+k+1}]$.

Boundary values $N_{j,k+1}^{(l)}(\tau_j) = N_{j,k+1}^{(l)}(\tau_{j+k+1}) = 0, \ l = 0, \ldots, k-1$.

Recursion $N_{j,l+1}(x) = \frac{x-\tau_j}{\tau_{j+l}-\tau_j} N_{j,l}(x) + \frac{\tau_{j+l+1}-x}{\tau_{j+l+1}-\tau_{j+1}} N_{j+1,l}(x)$, $N_{j,1}(x) = \begin{cases} 1, & x \in [\tau_j, \tau_{j+1}) \\ 0, & x \notin [\tau_j, \tau_{j+1}) \end{cases}$.

Derivative $N'_{j,k+1}(x) = k \left(\frac{N_{j,k}(x)}{\tau_{j+k}-\tau_j} - \frac{N_{j+1,k}(x)}{\tau_{j+k+1}-\tau_{j+1}} \right)$.

Suppose a spline $s(x) \in \eta_k(\tau_0, \ldots, \tau_{g+1})$ should be characterized by a linear combination of B-splines. There exist $g - k + 1$ linearly independent B-splines of degree k for the knots $a = \tau_0, \ldots, \tau_{g+1} = b$, namely $N_{0,k+1}, N_{1,k+1}, \ldots, N_{g-k,k+1}$. Another $2k$ linearly independent B-splines are needed for a basis of the $d = g + k + 1$-dimensional vector space η_k. If $2k$ additional knots $\tau_{-k} \leq \tau_{-k+1} \leq \ldots \leq \tau_{-1} \leq a$ and $b \leq \tau_{g+2} \leq \ldots \leq \tau_{g+k} \leq \tau_{g+k+1}$ are arbitrarily chosen, the corresponding B-splines $N_{-k,k+1}, \ldots, N_{-1,k+1}$ and $N_{g-k+1,k+1}, \ldots, N_{g,k+1}$ make the B-spline basis complete. See Figure 3.2B for an example of a B-spline basis. The full sequence of knots is denoted by $\mathcal{T} = (\tau_j)_{-k}^{g+k+1}$ in

the following. The knots $\tau_1, \ldots, \tau_g$ are called *inner knots*, while the remaining ones to the left and to the right are called *boundary knots*.
Each spline function $s(x)$ has a unique representation as linear combination of the basis functions of the form

$$s(x) = \sum_{j=-k}^{g} \beta_j N_{j,k+1}(x).$$

The constants β_j are called *B-spline coefficients* of $s(x)$. For $x \in [\tau_l, \tau_{l+1})$ it follows from the local support property of B-splines that $s(x) = \sum_{j=l-k}^{l} \beta_j N_{j,k+1}(x)$.
The basis $\{N_{-k,k+1}, \ldots, N_{g,k+1}\}$ forms a *partition of unity* on $[a, b]$, that is

$$\sum_{j=-k}^{g} N_{j,k+1}(x) = 1 \ \forall x \in [a, b]$$

while $N_{j,k+1} \geq 0 \ \forall x \in [a, b], \ \forall j$. This is why the B-splines are also called normalized (de Boor, 1978).
Another interesting characteristic of splines with respect to the B-spline basis is the *variation diminishing property*, which says that a spline $s(x) = \sum_{j=-k}^{g} \beta_j N_{j,k+1}(x)$ changes its sign at most as often as the corresponding sequence of B-spline coefficients $\beta_{-k}, \ldots, \beta_g$ does (de Boor, 1978). This property allows to easily impose shape constraints. For example, the usage of cubic splines under convexity constraints leads to a quadratic programming problem. In Section 3.4 we exploit the variation diminishing property to show that splines are well-suited for unimodal regression.
There exist several algorithms for the calculation of spline function values or values of derivatives. The algorithms are based on the recursion formulae for B-splines (see above) or the following recursion relation based on the B-spline coefficients β_j:
For the spline itself, it is true that

$$s(x) = \beta_j^{[k]}(x), \text{ with } \beta_j^{[i]}(x) = \begin{cases} \beta_j, & i = 0 \\ \frac{(x-\tau_j)\beta_j^{[i-1]}(x) + (\tau_{j+k+1-i} - x)\beta_{j-1}^{[i-1]}(x)}{\tau_{j+k+1-i} - \tau_j}, & i > 0 \end{cases}$$

and for a spline's derivatives it holds that

$$s^{(\nu)}(x) = \prod_{i=1}^{\nu}(k+1-i) \sum_{j=-(k-\nu)}^{g} \beta_j^{(\nu)} N_{j,k+1-\nu}(x) \text{ with } \beta_j^{(\nu)} = \begin{cases} \beta_j, & \nu = 0 \\ \frac{\beta_j^{(\nu-1)} - \beta_{j-1}^{(\nu-1)}}{\tau_{j+k+1-\nu} - \tau_j}, & \nu > 0 \end{cases}. \tag{3.1}$$

If $x \in [\tau_l, \tau_{l+1})$, some summands in the derivative are known to be zero and the reduced formula is $s^{(\nu)}(x) = \prod_{i=1}^{\nu}(k+1-i)\sum_{j=l-k+\nu}^{l}\beta_j^{(\nu)}N_{j,k+1-\nu}(x)$.
The above recursive computations are numerically stable. In addition, the local support and partition of unity properties induce sparsity and boundedness and thus, the B-spline basis is better suited for numerical computations than the TPF basis.

3.2.2 Suitability of B-splines for curve fitting

Besides its numerical stability, the B-spline basis is also sufficiently flexible and therefore suitable for curve fitting. To see this, we review some basic properties of Bernstein-Schoenberg splines (see, for example, Goodman, 1995), which are a particular choice of B-splines. Let $[a, b] = [0, 1]$ and let the sequence of knots be restricted to $\tau_{-k} = \ldots = \tau_0 = 0$, $\tau_{g+1} = \ldots = \tau_{g+k+1} = 1$ (so-called coincident boundary knots).
The *Bernstein-Schoenberg spline* (B-S spline) or *Bernstein-Schoenberg operator* of a function f has specific B-spline coefficients and is defined for $x \in [0, 1]$ as

$$\mathcal{V}_k^{\mathcal{T}} f(x) = \sum_{j=-k}^{g} f(\tau_j^*)N_{j,k+1}(x),$$

where $\tau_j^* = \frac{1}{k}\sum_{i=1}^{k}\tau_{j+i}$, $j = -k, \ldots, g$, are the so-called knot averages.
For the B-S spline it holds that, if $f \in \mathcal{C}^{\omega}[0, 1]$, then $\lim_{k\to\infty}(\mathcal{V}_k^{\mathcal{T}} f)^{(\omega)} = f^{(\omega)}$ uniformly on $[0, 1]$ for $\omega = 0, 1$.
For $g = 0$ the B-spline basis functions $N_{j,k+1}(x)$ $(j = -k, \ldots, 0)$ reduce to the binomial probabilities $\binom{k}{\ell}x^{\ell}(1-x)^{k-\ell}$ $(\ell = 0, \ldots, k)$ (see also Figure 3.2A) and the B-S splines reduce to the Bernstein polynomials, $\mathcal{B}_k f(x) = \sum_{\ell=0}^{k} f\left(\frac{\ell}{k}\right)\binom{k}{\ell}x^{\ell}(1-x)^{k-\ell}$, which have been increasingly employed for shape-constrained regression, see, for example, Chang et al. (2005) or more recently Wang and Ghosh (2012). Here, we prefer B-S splines since they are a straightforward, more flexible generalization of Bernstein polynomials and converge faster than the latter for suitably chosen knot sequences (cf. Marsden, 1970).

In summary, a function $f \in \mathcal{C}^{\omega}[0, 1]$ $(\omega = 0, 1)$ can be approximated by a spline $s(x) = \sum_{j=-k}^{g}\beta_j N_{j,k+1}(x)$ with $\beta_j = f(\tau_j^*)$ with uniform convergence properties. The same is true for functions on arbitrary intervals $[a, b]$ since their support can always be transformed to $[0, 1]$. B-S splines are a tool in approximation theory problems, but will be used here for regression purposes. Since the functional relationship f between predictor and response is unknown, we cannot choose $\beta_j = f(\tau_j^*)$, but have to estimate

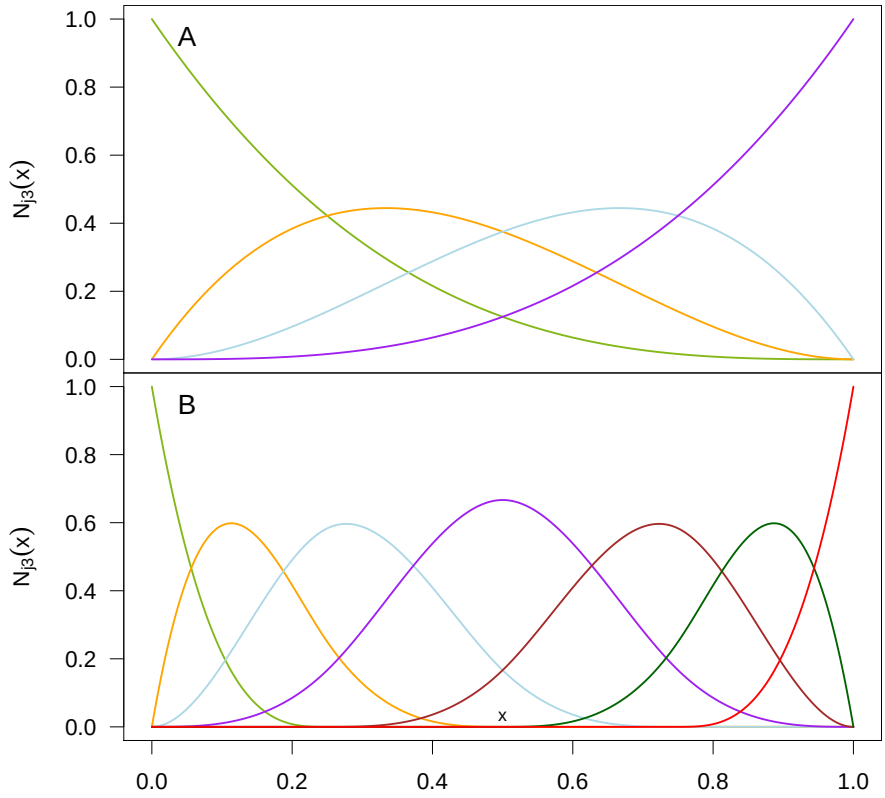


Figure 3.2: **Comparison of Bernstein and B-spline basis.**
(A) Bernstein basis / binomial probabilities with $k = 3$ (identical with cubic B-splines on the interval $[0, 1]$ without inner knots ($g = 0$) and with coincident outer knots on 0 and 1). (B) Cubic B-spline basis functions with $g = 3$ equidistant inner knots in the interval $[0, 1]$ and coincident outer knots on 0 and 1.

the B-spline coefficients $\boldsymbol{\beta} = (\beta_{-k}, \ldots, \beta_g) \in \mathbb{R}^d$ from data. However, the properties of B-S splines give an idea of the flexibility of the B-splines in the context of curve fitting.

3.2.3 Criteria for curve fitting with spline functions

Curve fitting with splines is based on the following approximation problem:
given n pairs of observations $(x_1, y_1), \ldots, (x_n, y_n)$ (sorted according to the x-values) of an independent variable X and a dependent variable Y it is desired to fit a function $f(x) = f(x|\boldsymbol{\theta})$ to the y-values which satisfies some kind of approximation criterion so that $f(x_i) \approx y_i$.
The parameters in $\boldsymbol{\theta}$ that have to be estimated (or fixed) when splines are used for curve fitting, that is, when $f(x|\boldsymbol{\theta}) = s(x) = \sum_{j=-k}^{g} \beta_j N_{j,k+1}(x)$, are

(i) the degree k of the spline,

(ii) the number and the positions of the knots τ_j,

(iii) the B-spline coefficients β_j.

Curve fitting may have several objectives. One of those is data smoothing: as the measurement of the values y_i is subjected with errors, one may hope to find a function f so that $f(x_i)$ is more accurate than y_i. Such a function would be supposed to look smooth. Another objective of curve fitting is to obtain a functional representation of the underlying relationship for the data. It enables to predict y-values at any point x in the observed range or to calculate derivatives, integral values etc. A third objective is data reduction. The parameters θ_i will usually require much less storage space than the data.
It is common to use splines of odd degree, explicitly cubic splines ($k = 3$). Recommendations for the determination of the other parameters are linked to particular approximation criteria, two of which will be discussed here: the least squares criterion and the criterion of the natural smoothing spline.

Applying the **least squares criterion** one is looking for the spline function $s(x)$ that minimizes the sum of squared deviations $\sum_{i=1}^{n}(y_i - s(x_i))^2$.
With fixed spline degree and fixed number and positions of the knots, the above least squares problem is linear in the B-spline coefficients β_i and can therefore be solved as an overdetermined and sparse (local support property) linear system. This can be seen

as follows.
Let

$$\boldsymbol{B} := \begin{pmatrix} N_{-k,k+1}(x_1) & \cdots & N_{g,k+1}(x_1) \\ \vdots & & \vdots \\ N_{-k,k+1}(x_n) & \cdots & N_{g,k+1}(x_n) \end{pmatrix} \in \mathbb{R}^{n \times d}$$

be the matrix of B-spline basis functions evaluated at the observation points $x_1, \ldots, x_n$. $\boldsymbol{y} = (y_1, \ldots, y_n)'$ is the vector of observed responses and $\boldsymbol{\beta} := (\beta_{-k}, \ldots, \beta_g)' \in \mathbb{R}^d$ the vector of B-Spline coefficients, so that

$$\boldsymbol{B\beta} = \begin{pmatrix} \sum_{j=-k}^{g} \beta_j N_{j,k+1}(x_1) \\ \vdots \\ \sum_{j=-k}^{g} \beta_j N_{j,k+1}(x_n) \end{pmatrix} = \begin{pmatrix} s(x_1) \\ \vdots \\ s(x_n) \end{pmatrix}.$$

We can rewrite the minimization problem as $\arg\min_{\boldsymbol{\beta}} \|\boldsymbol{y} - \boldsymbol{B\beta}\|_2^2$. If the number of distinct x-values is greater than or equal to the number of B-spline coefficients d, the matrix $\boldsymbol{B'B}$ is invertible and the unique solution is simply $\hat{\boldsymbol{\beta}} = (\boldsymbol{B'B})^{-1}\boldsymbol{B'y}$.
When assuming homoscedastic normally distributed errors, that is, when the data stem from the model

$$Y = s(x) + \mathcal{E} \text{ with } \mathcal{E} \sim \mathcal{N}(0, \sigma^2),$$

minimization of the least squares criterion is equivalent to maximization of the likelihood. Since the log-likelihood is proportional to $-\frac{1}{\sigma^2}||\boldsymbol{y} - \boldsymbol{B\beta}||_2^2$, the maximum likelihood approach leads to the same estimate $\hat{\boldsymbol{\beta}}$ as above. Thus, we will use this weighted form of the least squares criterion in the following, which will also help to unify upcoming approaches.
Unfortunately, choosing the number and positions of the knots properly is not trivial. Too many knots may lead to a very flexible and thus overfitted spline, while too few knots might cause underfitting. However, treating these parameters as variable and optimizing over, e.g., the knot positions is not trivial either since this requires solving a non-linear and constrained least squares problem.
The **criterion of the natural smoothing spline** gives a compromise between an approximation with a smooth behaviour (no overfitting) and an approximation with a close fit to the data values (no underfitting). This is achieved by minimizing the term $\int_{x_1}^{x_n} (s^{(l)}(x))^2 dx$ subject to the condition $\sum_{i=1}^{n} (y_i - s(x_i))^2 \leq h$, where h is a non-negative value called smoothing factor, which controls the compromise between smoothness and goodness of fit. The solution of this conditional minimization problem is a natural

spline of degree $k = 2l - 1$ with knots at the observation points, that is $\tau_i = x_{i+1},\ i = 0, 1, \ldots, g + 1 = n - 1$. There exist several procedures and algorithms for the exact determination of this spline. The parameter l is often chosen as two, resulting in a cubic smoothing spline.
The conditional minimization problem can be rewritten (here only for $k = 3$) as a penalized least squares problem of the form

$$\arg\min_s \sum_{i=1}^{n} (y_i - s(x_i))^2 + \lambda \int_{x_1}^{x_n} (s''(x))^2 dx,$$

where now the choice of the parameter λ gives the compromise between closeness to the data and smoothness in terms of curvature of the function f (cf. Hastie et al., 2009, ch. 5.4). So in contrast to the least squares criterion, the number and positions of the knots can be fixed, but the solution will be both reasonably smooth and reasonably close to the data. According to Eilers and Marx (1996) this advantage comes again at the cost of high complexity of the optimization problem and they propose to replace the smoothness penalty $\int_{x_1}^{x_n} (s''(x))^2 dx$ by a penalty based on finite differences of the B-spline coefficients. Such penalties enable the same compromise as the smoothness penalty, but are much easier to handle computationally. The concept of penalized spline regression is described in more detail in the following section.

3.3 Penalized spline regression

As already mentioned, the combination of the least squares criterion with a suitable penalty term can be used to solve the knot placing problem in the spline regression context. This section will discuss different penalization strategies.

3.3.1 Penalized least squares estimation

Eilers and Marx (1996) propose to use a relatively large number of knot positions (compared to the number of the predictor values and the variation in the data) and a penalty term based on finite differences of the B-spline coefficients to find a compromise between over- and underfitting. In their approach the term

$$\lambda \sum_{j=-k+q}^{g} (\Delta^q \beta_j)^2$$

is added to the objective function, where $\Delta\beta_j = \beta_j - \beta_{j-1}$, $\Delta^2\beta_j = \Delta(\Delta\beta_j) = \beta_j - 2\beta_{j-1} + \beta_{j-2}$ and so on.
The parameter $\lambda > 0$ enables the tuning of the penalization. In matrix notation the objective function is thus given by $\frac{1}{\sigma^2}\left\|\boldsymbol{y} - \boldsymbol{B}\boldsymbol{\beta}\right\|_2^2 + \lambda\left\|\boldsymbol{D}_q\boldsymbol{\beta}\right\|_2^2$, where $\boldsymbol{D}_q \in \mathbb{R}^{(d-q)\times d}$ is the matrix representation of the finite differences of order q, for example,

$$\boldsymbol{D}_2 = \begin{pmatrix} 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & 0 & & & & \\ & & & 1 & -2 & 1 \end{pmatrix} \in \mathbb{R}^{(d-2)\times d}.$$

Eilers and Marx (1996) propose to use second order differences ($q = 2$) and a knot sequence that is equidistant (also the boundary knots), that is, $\tau_j = a + j\frac{b-a}{g+1} \quad \forall\ j = -k, \ldots, g+k+1$. In this case the penalty term is zero for linear in- or decreasing coefficient sequences and thus, one penalizes against linear functions.
To enable inclusion of other penalties (see also next paragraph) we will use a generalized penalty term in the following, i.e., the objective function is the penalized residual sum of squares

$$\frac{1}{\sigma^2}\left\|\boldsymbol{y} - \boldsymbol{B}\boldsymbol{\beta}\right\|_2^2 + \lambda\left\|\boldsymbol{\Omega}^{\frac{1}{2}}(\boldsymbol{\beta} - \boldsymbol{\beta}_0)\right\|_2^2.$$

In the case of the difference penalty we thus have $\boldsymbol{\beta}_0 = \boldsymbol{0}$ and $\boldsymbol{\Omega} = \boldsymbol{D}_q'\boldsymbol{D}_q$.
For fixed λ the penalized objective function is minimized by

$$\hat{\boldsymbol{\beta}} = \left(\frac{1}{\sigma^2}\boldsymbol{B}'\boldsymbol{B} + \lambda\boldsymbol{\Omega}\right)^{-1}\left(\frac{1}{\sigma^2}\boldsymbol{B}'\boldsymbol{y} + \lambda\boldsymbol{\Omega}\boldsymbol{\beta}_0\right). \tag{3.2}$$

The question of how to determine a suitable value for the tuning parameter λ will be discussed in Section 3.5 after the introduction to shape-constrained splines in Section 3.4, so that it is possible to derive a universal approach for spline regression both with and without shape constraint.

For the penalized weighted residual sum of squares criterion holds a similar statement as for the least squares approach. Suppose that λ is fix and the coefficient vector has a (possibly improper) prior distribution $\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\beta}_0, \lambda^{-1}\boldsymbol{\Omega}^-)$, where $\boldsymbol{\Omega}^- \in \mathbb{R}^{d\times d}$ is the pseudo-inverse of $\boldsymbol{\Omega}$ if $r := \operatorname{rank}(\boldsymbol{\Omega}) < d$, and the regular inverse otherwise. Then, the

posterior is given by

$$p(\boldsymbol{\beta}|\boldsymbol{y}) \propto p(\boldsymbol{\beta}, \boldsymbol{y}) = p(\boldsymbol{y}|\boldsymbol{\beta})p(\boldsymbol{\beta}) \propto \exp\left\{-\frac{1}{2\sigma^2}\left\|\boldsymbol{y} - \boldsymbol{B}\boldsymbol{\beta}\right\|_2^2\right\} \exp\left\{-\frac{1}{2}\lambda\left\|\boldsymbol{\Omega}^{\frac{1}{2}}(\boldsymbol{\beta} - \boldsymbol{\beta}_0)\right\|_2^2\right\}$$

and the negative log-posterior of $\boldsymbol{\beta}$ is proportional to $\frac{1}{\sigma^2}\left\|\boldsymbol{y} - \boldsymbol{B}\boldsymbol{\beta}\right\|_2^2 + \lambda\left\|\boldsymbol{\Omega}^{\frac{1}{2}}(\boldsymbol{\beta} - \boldsymbol{\beta}_0)\right\|_2^2$. Thus, the penalized least squares estimation results in the same estimator of $\boldsymbol{\beta}$ as the maximum a posteriori approach.

3.3.2 Penalization against parametric functions

There are situations in which prior information about the functional relationship exists from preceding experiments, as, for example, in dose-response trials. Thus, one might have a particular parametric function in mind to estimate the relationship, but one would like to safeguard against mis-specification of this function (as discussed in Yuan and Yin (2011), for example). We now present how to integrate this information in form of a penalty.

Suppose we want to penalize against a function h. The linear interpolant of the points $(\tau^*_{-k}, \beta_{-k}), \ldots, (\tau^*_g, \beta_g)$, which Dierckx (1993) calls the "control polygon" of a spline, mimics the form of a spline. Thus, it seems natural to penalize against this form, in explicit to penalize the differences in the B-spline coefficients against the differences in the values of the fitted function $\hat{h}$ at the knot averages. Explicitly, the penalty term is

$$\left\|\boldsymbol{\Omega}^{\frac{1}{2}}(\boldsymbol{\beta} - \boldsymbol{\beta}_0)\right\|_2^2 = \sum_{j=-k}^{g}\left\{\Delta^q\beta_j - \Delta^q\hat{h}(\tau^*_j)\right\}^2 = \sum_{j=-k}^{g}\left\{\Delta^q(\beta_j - \hat{h}(\tau^*_j))\right\}^2, \tag{3.3}$$

that is, $\boldsymbol{\Omega}$ and $\boldsymbol{\beta}_0$ are given by $\boldsymbol{D}_q'\boldsymbol{D}_q$ and $\left(\hat{h}(\tau^*_{-k}), \ldots, \hat{h}(\tau^*_g)\right)'$. The fit $\hat{h}$ can stem from previous analyses on similar (historical) data or on the data at hand. Since we have already seen that a penalty term can be formulated as prior distribution of the B-spline coefficients, estimation of the penalty from the same data resembles an empirical Bayes approach.

In dose-response applications a possible choice of the parametric model is the sigmoid E_{max} model (implemented, e.g., in the R package `DoseFinding`, Bornkamp et al. (2016)). One characteristic of this model is that it allows for a steep increase in the response. Thus, if prior information suggests such a steep increase, it is possible to incorporate this knowledge by penalizing the spline against the fitted sigmoid E_{max} function. Explicitly,

the function $h_{\text{sigE}}(x) = E_0 + E_1 \frac{x^\zeta}{ED_{50}^\zeta + x^\zeta}$ with parameters E_0, E_1, ED_{50} and ζ and $\boldsymbol{\Omega} = \boldsymbol{I}_d$ ("zero order differences", $q = 0$) are employed in Equation (3.3). Another idea would be to penalize against the increase in a sigmoid-E_{max} fit, that is, using the first order ($q = 1$) differences matrix $\boldsymbol{\Omega} = \boldsymbol{D}_1'\boldsymbol{D}_1$. An overview of penalization possibilities will be given in the next section.

3.3.3 Possible penalties

As already seen, the choice of $\boldsymbol{\Omega}$ and $\boldsymbol{\beta}_0$ determines the nature of the penalty. Apart from second order differences of the B-spline coefficients as proposed in Eilers and Marx (1996) or penalization against parametric functions, which are summarised in Table 3.1, plenty of other possibilities are imaginable.

Table 3.1: **Possibilities to define the penalty term.**

penalty	$\boldsymbol{\beta}_0$	$\boldsymbol{\Omega}$
difference penalty of order $q = 0$ (ridge)	$\mathbf{0}$	$\boldsymbol{I}_d$
difference penalty of order $q = 1$	$\mathbf{0}$	$\boldsymbol{D}_1'\boldsymbol{D}_1$
difference penalty of order $q = 2$	$\mathbf{0}$	$\boldsymbol{D}_2'\boldsymbol{D}_2$
difference penalty of general order q	$\mathbf{0}$	$\boldsymbol{D}_q'\boldsymbol{D}_q$
sigmoid E_{max} penalty	$\left(\hat{h}_{\text{sigE}}(\tau_{-k}^*), \ldots, \hat{h}_{\text{sigE}}(\tau_g^*)\right)'$	$\boldsymbol{I}_d$
sigmoid E_{max} increase penalty	$\left(\hat{h}_{\text{sigE}}(\tau_{-k}^*), \ldots, \hat{h}_{\text{sigE}}(\tau_g^*)\right)'$	$\boldsymbol{D}_1'\boldsymbol{D}_1$
general functional penalty	$\left(\hat{h}(\tau_{-k}^*), \ldots, \hat{h}(\tau_g^*)\right)'$	$\boldsymbol{I}_d$

A very common penalty, also used for variable selection, is the so-called ridge penalty which has the sum of squared B-spline coefficients as penalty term. The first row of Table 3.1, a zero order difference penalty, corresponds to it. Thus it can be interpreted in the spline regression context as a penalty against a function, which is constantly zero. It makes sense for the IMS data sets, for example, where the intensities are very small over most of the x-axis. This penalty is obtained when $\boldsymbol{\Omega}$ is the $d \times d$ identity matrix and $\boldsymbol{\beta}_0 = \mathbf{0}$.

3.4 Shape-constrained splines

In this section we revert to the variation diminishing property of B-splines from Section 3.2 to illustrate that they are well-suited for shape-constrained modelling, as a

simple constraint on the B-spline coefficients ensures unimodality of a spline.
The characteristics of the control polygon (cf. Section 3.3.2) already indicate that the shape of the B-spline coefficients influences the shape of the spline function. A more formal argument is the variation diminishing property introduced by Schoenberg (1967). It says that the number of sign changes of a spline $\sum_{j=-k}^{g} \beta_j N_{j,k+1}(x)$ is not larger than the number of sign changes of the B-spline coefficient sequence $(\beta_{-k}, \ldots, \beta_g)$. Carnicer and Pena (1994) even found that the B-spline basis is *optimally* shape preserving for the space of spline functions. This property makes B-splines very interesting for shape-constrained regression. Concerning unimodality we derive the following lemma and proof on the basis of Lemma 1 in Köllmann et al. (2014):

Lemma 1. *Let $s(x) = \sum_{j=-k}^{g} \beta_j N_{j,k+1}(x)$ be a spline function on $[a, b]$ with knot sequence $\mathcal{T} = (\tau_j)_{-k}^{g+k+1}$. If $\exists\ m \in \{-k, \ldots, g\} : \beta_{-k} \leq \ldots \leq \beta_{m-1} \leq \beta_m \geq \beta_{m+1} \geq \ldots \geq \beta_g$ then s is a unimodal spline function.*

Proof of Lemma 1. Let $s(x) = \sum_{j=-k}^{g} \beta_j N_{j,k+1}(x)$ be a spline function on $[a, b]$ with knot sequence $\mathcal{T} = (\tau_j)_{-k}^{g+k+1}$ and $\exists\ m \in \{-k, \ldots, g\} : \beta_{-k} \leq \ldots \leq \beta_{m-1} \leq \beta_m \geq \beta_{m+1} \geq \ldots \geq \beta_g$. We have to prove that s is a unimodal function, or equivalently, that the derivative s' has at most one sign change.
According to Equation (3.1), the first derivative of s is given by

$$\begin{aligned} s^{(1)}(x) &= \prod_{i=1}^{1}(k+1-i) \sum_{j=-(k-1)}^{g} \beta_j^{(1)} N_{j,k+1-1}(x) \\ &= k \cdot \sum_{-k+1}^{g} \frac{\beta_j - \beta_{j-1}}{\tau_{j+k+1-1} - \tau_j} N_{j,k}(x) \\ &= \sum_{j=-k+1}^{g} \underbrace{\frac{k(\beta_j - \beta_{j-1})}{\tau_{j+k} - \tau_j}}_{:=\alpha_j} N_{j,k}(x), \end{aligned}$$

which is a spline function of degree $k-1$ with B-spline coefficients α_j, $j = -k+1, \ldots, g$. From Definition 1 we know that the inner knots of a spline all have to be distinct, so that $\tau_{j+k} - \tau_j > 0$ for $j = -k+1, \ldots, g$. Thus, the sign of α_j depends only on the sign of $\beta_j - \beta_{j-1}$.

We have that $\exists\, m \in \{-k, \ldots, g\}$: $\beta_{-k} \leq \ldots \leq \beta_{m-1} \leq \beta_m \geq \beta_{m+1} \geq \ldots \geq \beta_g$,

$$\Rightarrow \begin{cases} \alpha_j \leq 0 \;\forall\; j, & m = -k \\ \alpha_j \geq 0 \;\forall\; j \leq m \;\wedge\; \alpha_j \leq 0 \;\forall\; j \geq m+1, & m \notin \{-k, g\} \\ \alpha_j \geq 0 \;\forall\; j, & m = g \end{cases}.$$

Explicitly, this means that the coefficient sequence of the derivative has at most one sign change from negative to positive. Hence, according to the variation diminishing property, the derivative itself has at most one sign change and s is unimodal (including the special cases monotone increasing/decreasing or constant). □

The reverse implication does not necessarily hold, as can be seen from the following counterexample: Let $s_1 \in \eta_4(\tau_0 = 0, \tau_1 = 1)$ $(g = 0,\ k = 4)$ be the spline function with functional equation $s_1(x) = -8(x - 0.5)^4 + 0.5$. For coincident boundary knots $\tau_{-4} = \ldots = \tau_0 = 0, \tau_1 = \ldots = \tau_4 = 1$ the unique representation as linear combination of B-splines is given by the coefficients $\boldsymbol{\beta}_1 = (0, 1, 0, 1, 0)'$ (see Figure 3.3 for a graphical display of the spline and the B-spline basis). This can be seen as follows:
As mentioned in 3.2.2, the B-spline basis functions reduce to binomial probabilities for $g = 0$. Thus, we have that

$$\begin{aligned} s(x) &= \sum_{j=-4}^{0} \beta_j N_{j,5}(x) \\ &= N_{-3,5}(x) + N_{-1,5}(x) \\ &= \binom{4}{1} x^1 (1-x)^{4-1} + \binom{4}{3} x^3 (1-x)^{4-3} \\ &= 4x(1-x)^3 + 4x^3(1-x) \\ &= -8\left[x^4 - 2x^3 + \frac{3}{2}x^2 - \frac{1}{2}x\right] \\ &= -8\left[x^4 - 2x^3 + \frac{3}{2}x^2 - \frac{1}{2}x + \frac{1}{16} - \frac{1}{16}\right] \\ &= -8\left[\left(x - \frac{1}{2}\right)^4 - \frac{1}{16}\right] = -8(x-0.5)^4 + 0.5 \end{aligned}$$

Thus, s is a unimodal function, as it has only a single maximum at 0.5, but the coefficient sequence is clearly not unimodal.
However, due to the flexibility of spline functions this lack of equivalence is judged as unproblematic. Figure 3.3 shows that a spline function s_2 from the same spline space

but with unimodal coefficient sequence $\boldsymbol{\beta}_2 = (0.08, 0.65, 0.58, 0.58, 0.11)'$ is quite similar to spline s_1 (top) and that a spline function s_3 from the spline space with only one knot more ($g = 1$ inner knot $\tau_1 = 0.5$) and with unimodal coefficient sequence $\boldsymbol{\beta}_3 = (0, 0.5, 0.5, 0.5, 0.5, 0)'$ is already indistinguishable from spline s_1 (bottom). In fact, spline functions s_1 and s_3 are identical.
Thus, when choosing a large number of knots (and placing a penalty on the coefficients for smoothness as described in Section 3.3), we can suppose that an arbitrary spline, whether it has a unimodal coefficient sequence or not, can be described by a spline function with unimodal coefficients. This combination of penalization and shape constraint will be discussed in Section 3.5. Prior to this, we address the question of how to fit a unimodal, or more generally, a shape-constrained spline to data.

As follows from Lemma 1, it is sufficient to restrict the B-spline coefficients to form a unimodal sequence to ensure unimodality of the fitted spline. In addition, the proof of Lemma 1 also revealed two special cases of unimodality: Firstly, if $m = -k$, i.e., the mode of the coefficient sequence is its first element, then the spline has a non-positive derivative and is thus monotone decreasing. Secondly, if $m = g$, i.e., the mode of the coefficient sequence is its last element, then the spline has a non-negative derivative and is thus monotone increasing. It is also trivial to note that a B-spline coefficient sequence that is first decreasing and then increasing leads to an inverse unimodal (or also called "U-shaped") spline function.
More precisely, all those shape constraints can be written as the linear inequality $\boldsymbol{C}\boldsymbol{\beta} \geq \boldsymbol{0}$ with suitably chosen matrix $\boldsymbol{C} \in \mathbb{R}^{(d-1)\times d}$. For unimodality with fixed mode m of the B-spline coefficients this constraint matrix is given by

$$\boldsymbol{C} = \boldsymbol{C}_m = \begin{pmatrix} -1 & 1 & & & & & \\ & \ddots & \ddots & & & \boldsymbol{0} & \\ & & -1 & 1 & & & \\ & & & 1 & -1 & & \\ & \boldsymbol{0} & & & \ddots & \ddots & \\ & & & & & 1 & -1 \end{pmatrix} \leftarrow m\text{-th row.} \tag{3.4}$$

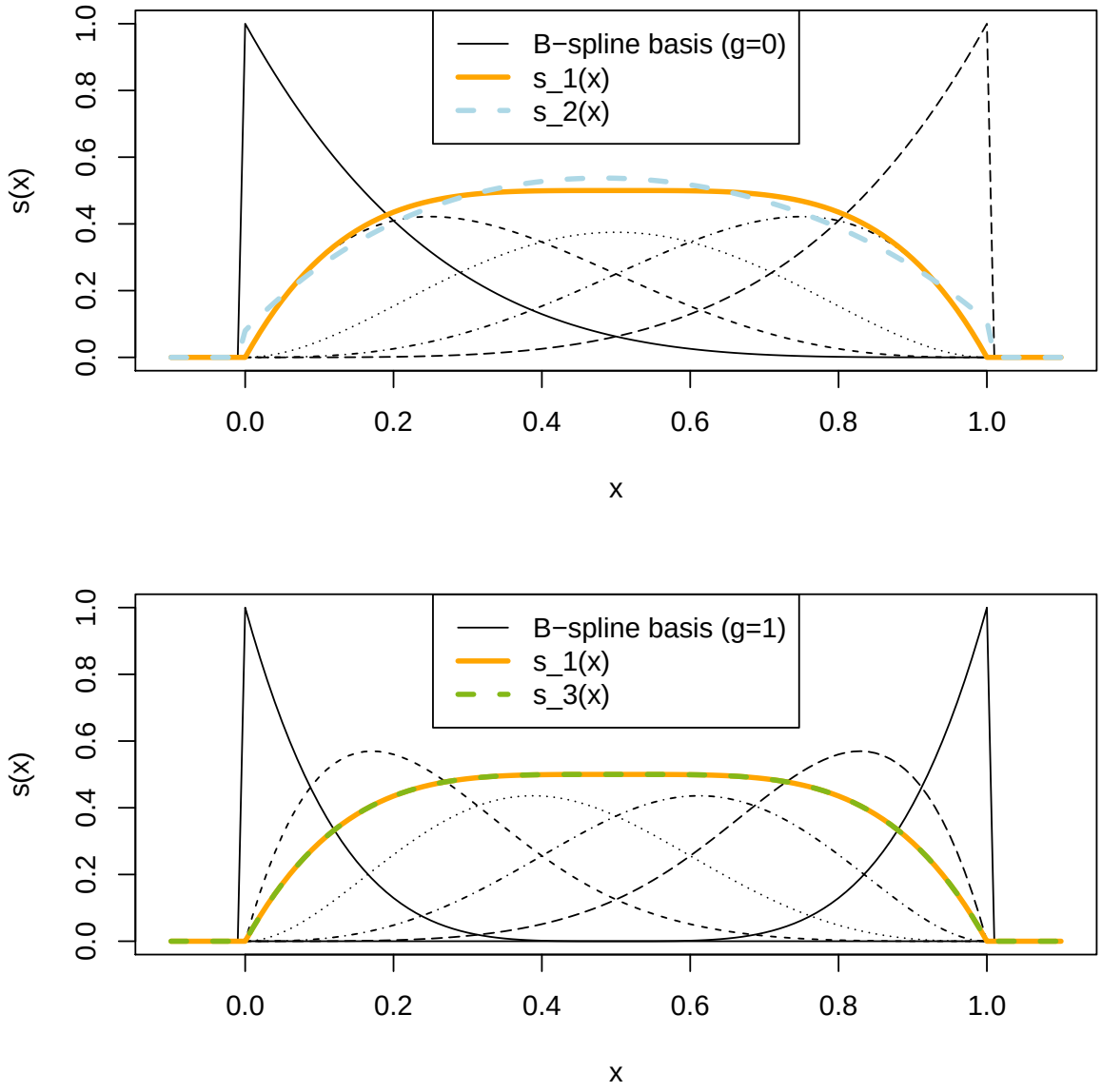


Figure 3.3: **Example of a unimodal spline function with non-unimodal coefficient sequence.**
Top: Spline functions s_1 and s_2 with coefficients $\boldsymbol{\beta}_1 = (0, 1, 0, 1, 0)'$ and $\boldsymbol{\beta}_2 = (0.08, 0.65, 0.58, 0.58, 0.11)'$ corresponding to the depicted B-spline basis with $g = 0$ inner knots. Bottom: Spline functions s_1 (as before) and s_3 with coefficient sequence $\boldsymbol{\beta}_3 = (0, 0.5, 0.5, 0.5, 0.5, 0)'$ corresponding to the depicted B-spline basis with $g = 1$ inner knot.

For the special case of increasing monotonicity we have

$$\boldsymbol{C} = \boldsymbol{C}_g = \begin{pmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & \mathbf{0} & & & \\ & & & -1 & 1 \end{pmatrix}.$$

For decreasing monotonicity and inverse unimodality with "mode" m the negative forms of the aforementioned matrices can be used: $\boldsymbol{C} = \boldsymbol{C}_{-k} = -\boldsymbol{C}_g$ and $\boldsymbol{C} = -\boldsymbol{C}_m$, respectively.
In summary, a shape-constrained spline can be fitted by minimizing the (weighted) least squares criterion

$$\frac{1}{\sigma^2} \|\boldsymbol{y} - \boldsymbol{B}\boldsymbol{\beta}\|_2^2 \text{ subject the constraints } \boldsymbol{C}\boldsymbol{\beta} \geq \mathbf{0}.$$

As already mentioned in Section 3.2, such shape constraints result in a quadratic programming problem (defined by a quadratic objective function with linear constraint), which do not have an explicit solution, but can be solved efficiently and reliably, see for example the R package `quadprog` (Turlach and Weingessel, 2013).
Usually the mode of the coefficient vector will be unknown and must be learned from data. We propose to fit a unimodal (or an inverse unimodal) regression for each possible choice of $m \in \{-k, \ldots, g\}$ and then select the regression coefficient $\hat{\boldsymbol{\beta}}_m$ with lowest residual sum of squares $\left\|\boldsymbol{y} - \boldsymbol{B}\hat{\boldsymbol{\beta}}_m\right\|_2^2$. This idea is similar to the one in Frisén (1986), but a global shape constraint is used instead of two local ones. On the one hand, this selection mechanism increases computation time, but on the other hand, the possibility to fix the mode at a certain value can also be advantageous (cf. the deconvolution model with varying peak shapes in Section 4.3.4). An alternative is to take a (Bayesian) average over different choices of the mode, which is done in Section 3.6. But prior to this we intend to combine shape-constrained and smoothness-penalized splines in a frequentist setting.

3.5 Frequentist penalized unimodal spline regression

This section aims at bringing the penalized spline regression from Section 3.3 and the shape-constrained spline regression from Section 3.4 together. We will first discuss how to estimate a regression spline for fixed λ within this combined framework before tackling

the problem of how to choose the tuning parameter.

3.5.1 Combining shape constraint and penalty

In this subsection the underlying model is

$$\boldsymbol{\mathcal{Y}}|\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{B}\boldsymbol{\beta}, \sigma^2 \boldsymbol{I}_n),$$

and it is desired to find the best fitting unimodal spline that is also appropriately smooth. In explicit, the aim is a combination of unimodal spline regression and penalized spline regression. For the time being suppose that the standard deviation σ is known. This assumption will be relaxed in the further course.
The combination of unimodal and penalized spline regression is quite straightforward. In explicit, for a fixed tuning parameter λ the penalized objective function

$$\frac{1}{\sigma^2} \|\boldsymbol{y} - \boldsymbol{B}\boldsymbol{\beta}\|_2^2 + \lambda \left\| \boldsymbol{\Omega}^{\frac{1}{2}} (\boldsymbol{\beta} - \boldsymbol{\beta}_0) \right\|_2^2$$

is now minimized over the set $\mathcal{S}_m$ of all unimodal coefficient vectors $\boldsymbol{\beta}$ with mode m, that is, $\mathcal{S}_m := \{\boldsymbol{\beta} \in \mathbb{R}^d : \boldsymbol{C}_m \boldsymbol{\beta} \geq \boldsymbol{0}\}$.
Again, the constrained problem cannot be solved explicitly, but for each mode m the estimate $\hat{\boldsymbol{\beta}}_m$ can be found using quadratic programming algorithms and the best fitting mode is found by minimizing the residual sum of squares criterion over all possible choices from $-k$ to g.

3.5.2 REML estimation of the tuning parameter

Before estimating $\boldsymbol{\beta}$ in the above described manner, a value for the tuning parameter λ has to be chosen, which gives a compromise between overfitting and underfitting. In simple regression problems the method of choice is often leave-one-out cross-validation, where the hat matrix $\boldsymbol{H}$ for which $\hat{\boldsymbol{y}} = \boldsymbol{H}\boldsymbol{y}$ can be used as a shortcut for the estimation of the tuning parameter. But when estimating $\boldsymbol{\beta}$ (and calculating $\hat{\boldsymbol{y}}$) under the unimodality constraint, such a matrix and thus a similar simple way of calculating the cross-validated tuning parameter is not available. In what follows we will describe a restricted maximum-likelihood (REML) procedure for estimating λ, as this can relatively straightforwardly be extended to the constrained case. We follow an approach, for example, taken by Wood (2011) in other penalized likelihood problems, where no shape constraint is present, and assume that $\boldsymbol{\beta}$ and λ are random. The restricted likelihood of λ is the marginal posterior

density $p(\lambda|\boldsymbol{y})$ which can be found by integrating $\boldsymbol{\beta}$ out of the joint posterior $p(\boldsymbol{\beta}, \lambda|\boldsymbol{y})$, in explicit, $p(\lambda|\boldsymbol{y}) = \int p(\boldsymbol{\beta}, \lambda|\boldsymbol{y}) d\boldsymbol{\beta}$. An automatic choice of the tuning parameter λ based on REML estimation is then $\hat{\lambda} = \arg\max_{\lambda} p(\lambda|\boldsymbol{y})$. The following Lemma 2 specifies the form of the marginal likelihood:

Lemma 2. *Assuming the model*

$$\begin{aligned} \boldsymbol{\mathcal{Y}}|\boldsymbol{\beta} &\sim \mathcal{N}(\boldsymbol{B}\boldsymbol{\beta}, \sigma^2 \boldsymbol{I}_n), \quad \sigma^2 > 0 \\ \boldsymbol{\beta}|\lambda &\sim \mathcal{N}_{\mathcal{M}}(\boldsymbol{\beta}_0, \lambda^{-1}\boldsymbol{\Omega}^{-1}), \quad \lambda > 0, \boldsymbol{\Omega} \text{ } pd, \mathcal{M} \subseteq \mathbb{R}^d \\ \lambda &\sim p(\lambda), \text{ } \textit{a prior density on } (0, \infty), \end{aligned}$$

the marginal posterior density of λ is given by

$$p(\lambda|\boldsymbol{y}) \propto p(\lambda) \; |\lambda\boldsymbol{\Omega}|^{\frac{1}{2}} \; |\mathbf{E}_\lambda|^{\frac{1}{2}} \frac{c_\lambda^{post}}{c_\lambda^{prior}} \; \exp\left(\frac{1}{2}\mathbf{e}_\lambda' \mathbf{E}_\lambda^{-1} \mathbf{e}_\lambda - \frac{1}{2}\lambda\boldsymbol{\beta}_0'\boldsymbol{\Omega}\boldsymbol{\beta}_0\right),$$

where $\mathbf{E}_\lambda := \left(\frac{1}{\sigma^2}\boldsymbol{B}'\boldsymbol{B} + \lambda\boldsymbol{\Omega}\right)^{-1}$, $\mathbf{e}_\lambda' := \left(\frac{1}{\sigma^2}\boldsymbol{y}'\boldsymbol{B} + \lambda\boldsymbol{\beta}_0'\boldsymbol{\Omega}\right)\mathbf{E}_\lambda$ *and* c_λ^{prior} *is the probability of the truncation set* $\mathcal{M}$ *under* $\mathcal{N}(\boldsymbol{\beta}_0, \lambda^{-1}\boldsymbol{\Omega}^{-1})$ *and* c_λ^{post} *is its probability under* $\mathcal{N}(\mathbf{e}_\lambda, \mathbf{E}_\lambda)$.

A proof is given in Appendix B.

The notation $\mathcal{N}_{\mathcal{M}}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ in Lemma 2 stands for a multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ truncated to the set $\mathcal{M} \subseteq \mathbb{R}^d$. For $\mathcal{M} = \mathcal{S}_m$ we have the unimodality constraint with mode m of the B-spline coefficients and for $\mathcal{M} = \mathbb{R}^d$ there is no shape constraint. In the unconstrained case both normalizing constants, c_λ^{prior} and c_λ^{post}, are the probability of $\mathbb{R}^d$ under multivariate normal distributions and thus, equal to one. Under the unimodality shape constraint the normalizing constants are the respective probabilities of $\mathcal{S}_m$, which can be approximated numerically (see, for example, the R package `mvtnorm` by Genz et al. (2012)).

If $\boldsymbol{\Omega}$ is positive definite as required in Lemma 2 the factor $|\lambda\boldsymbol{\Omega}|^{\frac{1}{2}}$ in the marginal posterior can be simplified to $\lambda^{\frac{d}{2}}|\boldsymbol{\Omega}|^{\frac{1}{2}} \propto \lambda^{\frac{d}{2}}$. If one is interested in a model, where $\boldsymbol{\Omega}$ is not positive definite (in explicit, the prior is improper), as, for example, when using a difference penalty, Lemma 2 does not hold, but we propose to use the specified formula nevertheless, though with two variations: First, we have that $|\lambda\boldsymbol{\Omega}|^{\frac{1}{2}} = \lambda^{\frac{r}{2}}|\boldsymbol{\Omega}|_+^{\frac{1}{2}} \propto \lambda^{\frac{r}{2}}$, where $r = \text{rank}(\boldsymbol{\Omega})$ and $|\cdot|_+$ denotes the pseudo-determinant, which is the product of all non-zero eigenvalues of a square matrix (cp. Wood, 2011). Second, the normalizing

constant of the prior does not exist in these cases and we propose to calculate c_λ^{prior} as the normalizing constant of a slightly modified proper prior: If $\boldsymbol{\Omega}$ is determined by a general difference matrix $\boldsymbol{D}_q$, the modified covariance matrix of the prior is given by $\tilde{\boldsymbol{\Omega}}_\lambda = \frac{1}{\sigma_v^2}\boldsymbol{I}_d + \lambda \boldsymbol{D}_q'\boldsymbol{D}_q$. This can be interpreted as a combination of ridge and difference penalty, where we propose to keep the "tuning parameter" σ_v^2 of the ridge penalty fixed, so that the influence of the difference penalty still increases with higher λ.

To sum up, we propose to choose the tuning parameter as $\hat{\lambda} = \arg\max_\lambda w_1(\lambda)$ with

$$w_1(\lambda) = p(\lambda)\ \lambda^{\frac{r}{2}}\ |\mathbf{E}_\lambda|^{\frac{1}{2}} \frac{c_\lambda^{post}}{c_\lambda^{prior}}\ \exp\left(\frac{1}{2}\mathbf{e}_\lambda'\mathbf{E}_\lambda^{-1}\mathbf{e}_\lambda - \frac{1}{2}\lambda\boldsymbol{\beta}_0'\boldsymbol{\Omega}\boldsymbol{\beta}_0\right), \tag{3.5}$$

where $r = \text{rank}(\boldsymbol{\Omega})$ and c_λ^{prior} and c_λ^{post} as specified above. The function w_1 is proportional to the marginal posterior (if $\text{rank}(\boldsymbol{\Omega}) < d$ only approximately) and therefore has the same maximum. The estimate of λ can be found using optimization algorithms and is then used in the estimation procedure for $\boldsymbol{\beta}$ described in Section 3.5.1.
Since we also consider models for multimodal data in Chapter 4, where the unimodal regression has to be applied repeatedly, we propose to choose λ by approximate REML to reduce the computational burden in those approaches. This means that the unimodality constraint is not accounted for during tuning parameter optimization and the normalizing constants in w_1 are set to one, resulting in the same tuning parameter as for the respective unconstrained model.
The described procedure is not invariant to scaling of the data, thus, we propose to scale the observations $\boldsymbol{y}$ into $[-1, 1]$ and transform the fitted values back. For the simulation study in Chapter 6 and the applications in Chapter 7 we choose $\sigma_v^2 = 5$, which can be thought of as uninformative since the B-spline coefficients β_i, which mimic the functional relationship, approximately also lie in $[-1, 1]$.
In real applications the standard deviation σ is usually unknown and has to be estimated as well. If it is possible to attain an accurate estimate from preceding trials or from the data itself, as, for example, in dose-response trials with repeated measurements, then the above process works fine. In other cases, it is still possible to iterate between estimation of σ given an interim estimate of $\boldsymbol{\beta}$ and estimation of $\boldsymbol{\beta}$ given an interim estimate of σ until some defined convergence.
When replacing the maximization in the above procedure by averaging, we arrive at the Bayesian approach that will be discussed in the next section.

3.6 Bayesian unimodal spline regression

In contrast to performing model selection over a grid of possible values for the mode and maximizing the restricted likelihood over λ, one can also think of model averaging based on a Bayesian approach. Then, besides $\boldsymbol{\beta}$ and λ, the mode is random, too. In explicit, the model is given by

$$\begin{aligned} \boldsymbol{\mathcal{Y}}|\boldsymbol{\beta} &\sim \mathcal{N}(\boldsymbol{B}\boldsymbol{\beta}, \sigma^2 \boldsymbol{I}_n), \quad \sigma^2 > 0, \\ \boldsymbol{\beta}|\lambda, m &\sim \mathcal{N}_{\mathcal{S}_m}(\boldsymbol{\beta}_0, \tilde{\boldsymbol{\Omega}}_\lambda^{-1}), \quad \tilde{\boldsymbol{\Omega}}_\lambda \text{ with full rank}, \\ (\lambda, m) &\sim p(\lambda, m), \end{aligned} \tag{3.6}$$

where $p(\lambda, m)$ is a prior density on $(0, \infty) \times \{-k, \ldots, g\}$. A simple choice of this joint prior for tuning parameter and mode will be discussed at the end of this section.
For the prior of $\boldsymbol{\beta}$ we directly use the full-rank precision matrix $\tilde{\boldsymbol{\Omega}}_\lambda = \frac{1}{\sigma_v^2}\boldsymbol{I}_d + \lambda \boldsymbol{D}_q' \boldsymbol{D}_q$. The variance σ^2 of the responses and the penalty components $\boldsymbol{\beta}_0$ and $\tilde{\boldsymbol{\Omega}}_\lambda$ are assumed to be fixed, since they are either known from preceding experiments or can be estimated from the data leading to an empirical Bayes approach (cf. Section 3.3.2).

For model (3.6) the joint posterior distribution factorizes as

$$p(\boldsymbol{\beta}, \lambda, m|\boldsymbol{y}) = p(\boldsymbol{\beta}|\lambda, m, \boldsymbol{y}) p(\lambda|m, \boldsymbol{y}) p(m|\boldsymbol{y}). \tag{3.7}$$

Thus, we can generate a Monte Carlo random sample from the posterior distribution by sampling successively from $p(m|\boldsymbol{y})$, $p(\lambda|m, \boldsymbol{y})$ and $p(\boldsymbol{\beta}|\lambda, m, \boldsymbol{y})$. Estimators for $\boldsymbol{\beta}$, λ and m are obtained by posterior averages, in explicit, means or medians of the posterior samples.
The following Lemma 3 states two marginal posterior densities which are essential for the proposed random sampling scheme.

Lemma 3. *Assume model (3.6), then*

(i) the marginal posterior density of $\boldsymbol{\beta}$ is given by

$$p(\boldsymbol{\beta}|\lambda, m, \boldsymbol{y}) \propto |\mathbf{E}_\lambda|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{\beta} - \mathbf{e}_\lambda)' \mathbf{E}_\lambda^{-1} (\boldsymbol{\beta} - \mathbf{e}_\lambda)\right\} \mathbb{1}_{\mathcal{S}_m}(\boldsymbol{\beta}),$$

that is

$$\boldsymbol{\beta}|\lambda, m, \boldsymbol{y} \sim \mathcal{N}_{\mathcal{S}_m}(\mathbf{e}_\lambda, \mathbf{E}_\lambda),$$

(ii) the marginal posterior density of $(\lambda, m)'$ *is given by*

$$p(\lambda, m|\boldsymbol{y}) \propto p(\lambda, m)\ |\tilde{\boldsymbol{\Omega}}_\lambda|^{\frac{1}{2}}\ |\mathbf{E}_\lambda|^{\frac{1}{2}}\ \frac{c_\lambda^{post}}{c_\lambda^{prior}} \exp\left(\frac{1}{2}\mathbf{e}_\lambda'\mathbf{E}_\lambda^{-1}\mathbf{e}_\lambda - \frac{1}{2}\boldsymbol{\beta}_0'\tilde{\boldsymbol{\Omega}}_\lambda\boldsymbol{\beta}_0\right),$$

where $\mathbf{E}_\lambda := \left(\frac{1}{\sigma^2}\boldsymbol{B}'\boldsymbol{B} + \tilde{\boldsymbol{\Omega}}_\lambda\right)^{-1}$, $\mathbf{e}_\lambda' := \left(\frac{1}{\sigma^2}\boldsymbol{y}'\boldsymbol{B} + \boldsymbol{\beta}_0'\tilde{\boldsymbol{\Omega}}_\lambda\right)\mathbf{E}_\lambda$ *and* c_λ^{prior} *is the probability of the truncation set* $\mathcal{S}_m$ *under* $\mathcal{N}(\boldsymbol{\beta}_0, \tilde{\boldsymbol{\Omega}}_\lambda^{-1})$ *and* c_λ^{post} *is its probability under* $\mathcal{N}(\mathbf{e}_\lambda, \mathbf{E}_\lambda)$.

The proof can be found in Appendix B.

Define $w_2(\lambda, m)$ as the right hand side of (ii) in Lemma 3. The posterior density of the mode is given by $p(m|\boldsymbol{y}) = \int p(\lambda, m|\boldsymbol{y})d\lambda \propto \int w_2(\lambda, m)d\lambda$. A regular distribution is obtained by the following normalization step:

$$p(m = m^*|\boldsymbol{y}) = \frac{\int w_2(\lambda, m^*)d\lambda}{\sum_{j=-k}^{g}\int w_2(\lambda, j)d\lambda}, \tag{3.8}$$

where the integration of $w_2(\lambda, m)$ can be done numerically as will be explained in Section 5.2.

Furthermore, the marginal posterior density of the tuning parameter, $p(\lambda|m, \boldsymbol{y}) = \frac{p(\lambda, m|\boldsymbol{y})}{p(m|\boldsymbol{y})}$, is a univariate continuous density proportional to $w_2(\lambda, m)$.

We propose the following random sampling scheme:

1. Draw a random number m from $\{-k, \ldots, g\}$ according to the discrete distribution $p(m|\boldsymbol{y})$ in (3.8).

2. Given m, draw a random number λ from $p(\lambda|m, \boldsymbol{y}) \propto w_2(\lambda, m)$ using, for example, the slice sampler introduced by Neal (2003).

3. Given m and λ, draw a random vector $\boldsymbol{\beta}$ from $\mathcal{N}_{\mathcal{S}_m}(\mathbf{e}_\lambda, \mathbf{E}_\lambda)$, for example, using the inverse Bayes formulae sampler proposed by Yu and Tian (2011) (cf. Appendix C).

Because of factorization (3.7) this successive sampling scheme yields a Monte Carlo random sample with uncorrelated draws from the joint posterior. Markov Chain Monte Carlo (MCMC) methods (inverse Bayes formulae and slice sampler) are only necessary for generating a single draw from $p(\boldsymbol{\beta}|\lambda, m, \boldsymbol{y})$ and $p(\lambda|m, \boldsymbol{y})$ for each joint posterior sample.

In the same way as for the REML approach in Section 3.5.2, we recommend to transform

the observations $\boldsymbol{y}$ onto $[-1, 1]$ before fitting the model, to achieve a scale invariant procedure. For the simulations and applications in Chapters 6 and 7 we use independent priors for the tuning parameter and the mode, that is $p(\lambda, m) = p(\lambda)p(m)$, where $p(\lambda) \propto \frac{1}{\lambda}\mathbb{1}_{[e^{-3},e^{10}]}(\lambda)$, since λ can also be interpreted as a scale parameter, and $p(m) = \frac{1}{d}\ \forall\ m \in \{-k, \ldots, g\}$. Both priors are uninformative when considered individually, but their joint distribution might be informative for the model at hand. Thus, a joint Jeffreys prior will be derived in the next section.

3.7 Robust unimodal spline regression

Up to now all observations contributed equally to the estimation of the spline regression functions, but since the spline regression model can be written in the form of a standard linear model, weights can easily by incorporated. Strictly speaking, we already used weighted fitting criteria, but all observations received the same weight, namely the inverse of the standard deviation. It is often desirable to down-weight certain observations compared to others, for example in case of heteroscedasticity or in the presence of outliers. In this section we first address weighted spline regression. Afterwards, robust estimation procedures are presented which use iterative re-weighting schemes.

3.7.1 Weighted spline regression

Suppose our data stem from the model

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{B}\boldsymbol{\beta} + \boldsymbol{\mathcal{E}}, \quad \boldsymbol{\mathcal{E}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}),$$

where $\boldsymbol{\Sigma}$ is usually a diagonal matrix with entries $\frac{\sigma^2}{w_i},\ i = 1, \ldots, n$, though the case of correlated errors can also be expressed by using a suitable covariance matrix.
In this scenario, the penalized objective function for (constrained or unconstrained) spline regression is given by

$$\underbrace{(\boldsymbol{y} - \boldsymbol{B}\boldsymbol{\beta})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{y} - \boldsymbol{B}\boldsymbol{\beta})}_{=\frac{1}{\sigma^2}\sum_{i=1}^{n} w_i(y_i - s(x_i))^2 \text{for uncorrelated errors}} + \lambda(\boldsymbol{\beta} - \boldsymbol{\beta}_0)'\boldsymbol{\Omega}(\boldsymbol{\beta} - \boldsymbol{\beta}_0).$$

For a fixed value λ the minimizer of this objective function is given by

$$\hat{\boldsymbol{\beta}} = \left(\frac{1}{\sigma^2}\boldsymbol{B}'\boldsymbol{\Sigma}^{-1}\boldsymbol{B} + \lambda\boldsymbol{\Omega}\right)^{-1}\left(\frac{1}{\sigma^2}\boldsymbol{B}'\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{y} + \lambda\boldsymbol{\Omega}\boldsymbol{\beta}_0\right)$$

in the unconstrained case or can be found with quadratic programming subject to the desired constraint. When estimating the tuning parameter λ via REML in this setting, the matrix $\mathbf{E}_\lambda$ and vector $\mathbf{e}_\lambda$ in the restricted likelihood of λ have to be changed accordingly: $\mathbf{E}_\lambda := \left(\frac{1}{\sigma^2}\boldsymbol{B}'\boldsymbol{\Sigma}^{-1}\boldsymbol{B} + \lambda\boldsymbol{\Omega}\right)^{-1}$ and $\mathbf{e}'_\lambda := \left(\frac{1}{\sigma^2}\boldsymbol{y}'\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{B} + \lambda\boldsymbol{\beta}'_0\boldsymbol{\Omega}\right)\mathbf{E}_\lambda$.

3.7.2 Robust estimation: iteratively re-weighted least squares

Many robust linear regression estimators can be obtained by repeatedly using weighted least squares regression. The estimation algorithm iterates between updating the regression coefficients and updating the weights and thus, this procedure is also called iteratively re-weighted least squares (IRLS).

Again, since the spline regression model can be written in the form of a standard linear model, it suggests itself to adapt this algorithm for (shape-constrained) penalized spline regression. For unconstrained cases, robust penalized spline regression approaches were proposed in the literature. For example, Lee and Oh (2007) introduce M-estimation for penalized regression splines and Tharmaratnam et al. (2010) provide an S-type estimation procedure. The Master's thesis of Vanessa Baumann (Baumann, 2014) addresses possibilities to robustify the penalized unimodal spline regression proposed in Section 3.5. This Master's thesis was co-supervised by the author of this Ph.D. thesis, who also proposed the statistical methodology used therein. The basic ideas are explained in the following.

In general the objective function for robust penalized spline regression is given by

$$\sum_{i=1}^{n} \rho\left(\frac{y_i - \boldsymbol{b}_i'\boldsymbol{\beta}}{\sigma}\right) + \lambda(\boldsymbol{\beta} - \boldsymbol{\beta}_0)'\boldsymbol{\Omega}(\boldsymbol{\beta} - \boldsymbol{\beta}_0),$$

where $\boldsymbol{b}_i$ is the i-th row of the B-spline basis matrix $\boldsymbol{B}$, σ is the previously estimated scale and $\rho : \mathbb{R} \mapsto \mathbb{R}_0^+$ is a loss function. When defining the weight function $w(r) = \frac{\rho'(r)}{r}$ and given a start estimate for the coefficient vector, the minimizer of the objective function is found by iteratively calculating

$$\begin{aligned} \boldsymbol{W} &= diag(w_1, \ldots, w_n) \text{ with } w_i = w\left(\frac{y_i - \boldsymbol{b}_i'\hat{\boldsymbol{\beta}}}{\sigma}\right) \text{ and} \\ \hat{\boldsymbol{\beta}} &= \left(\frac{1}{\sigma^2}\boldsymbol{B}'\boldsymbol{W}^{-1}\boldsymbol{B} + \lambda\boldsymbol{\Omega}\right)^{-1}\left(\frac{1}{\sigma^2}\boldsymbol{B}'\boldsymbol{W}^{-\frac{1}{2}}\boldsymbol{y} + \lambda\boldsymbol{\Omega}\boldsymbol{\beta}_0\right), \end{aligned}$$

compare also Holland and Welsch (1977).
For shape-constrained estimation, it seems obvious to replace the weighted least squares estimator by its shape-constrained counterpart, which is estimated by optimizing the weighted objective function subject to the desired constraint. In each step, the tuning parameter can be chosen via REML.
Typical choices for the loss function are, for example,

$$\rho(r) = |r|$$

for median regression or

$$\rho_c(r) = \begin{cases} r^2 & |r| \leq c \\ 2c|r| - c & |r| > c \end{cases}$$

for Huber-M-regression.
For the first choice, the optimal $\boldsymbol{\beta}$ can be found with linear programming algorithms more efficiently, though. For example, median regression models, as a special case of quantile regression, can be fitted in R with function `rq` from package `quantreg` (Koenker, 2016). The optimization can also be done subject to linear constraints on $\boldsymbol{\beta}$ when choosing `method="fnc"` and thus, unpenalized median unimodal spline regression can easily be done using this package. This approach can be used to find a start estimate $\hat{\boldsymbol{\beta}}_0$ and an estimate of the scale, $\hat{\sigma}^2 = median(|r_1|, \ldots, |r_n|)/0.675$, where $r_i = y_i - \boldsymbol{b}_i\hat{\boldsymbol{\beta}}_0$, $i = 1, \ldots, n$.
The second choice, Huber's ρ-function, was also used in Lee and Oh (2007), though their fitting algorithm is not based on IRLS, but uses iteratively re-computed pseudo-data. Baumann (2014) uses a version of their algorithm adapted for shape-constrained estimation as well as the above shape-constrained IRLS algorithm in a simulation study. Further estimation methods included in the comparison are least squares, the unconstrained original estimation procedures of Lee and Oh (2007) as well as Tharmaratnam et al. (2010) and the non-robust unimodal regression from Section 3.5. The results will be shortly summarised in Section 7.3.

4 Multimodal regression

4.1 Overview

The different approaches to unimodal spline regression presented in Section 3 cannot only be seen as methodology on its own, but also as a building block for more complex models. This chapter aims at presenting methodology which enables the handling of a broad spectrum of applications with multimodal data. Such multimodality can have different reasons and thus, one can take different approaches when modelling multimodal data.

First of all, the observed multiple modes can arise due to an inhomogeneous population. In explicit, two or more subpopulations are present to which the observed entities belong with certain probabilities. Thus, Section 4.2 deals with the task of modelling data observed from several subpopulations, where the predictor-response-relationship exhibits different shapes, especially unimodal shapes with different modes, in the subpopulations.

Another source of multimodality are series of non-overlapping or overlapping unimodal pieces in the predictor-response-relationship of a homogeneous population. The focus of this chapter is on the latter type of populations and several modelling approaches are presented in Section 4.3.

A common feature of the presented methods is that the response variable is modelled using several unimodal functions, which will also be referred to as "peaks". There are applications where knowledge about the shape of those individual peaks exists. The shape might either be known exactly or can be described by a parametric function. If this is not the case, more sophisticated estimation methods for the peak shape are required, for example the unimodal spline regression approaches proposed in Chapter 3. The article Köllmann et al. (2016) is based in parts on the following material.

4.2 Inhomogeneous population

In this section, we briefly tackle the problem of an inhomogeneous population in which predictor and response variable have been observed. To be more precise, we assume the presence of two or more subpopulations to whom the observed units belong with certain probabilities and where the response variable follows different unimodal relationships in each of the subpopulations. In the whole population the relationship between predictor and response appears multimodal. An example of unimodal functions in three subpopulations and multimodal data simulated from it are depicted in the third row of Figure 4.1.

A modelling approach for such data is a mixture model of regressions, where each component of the mixture is a unimodal regression and each observation stems from the different components with a certain probability. In this model the random variable Y has the conditional density

$$p(Y = y|x, \boldsymbol{\theta}) = \sum_{\ell=1}^{L} \pi_\ell \, \varphi\left(y|s_\ell(x), \sigma_\ell^2\right), \tag{4.1}$$

where $\varphi\left(y|s_\ell(x), \sigma_\ell^2\right)$ is the density of a normal distribution with variance σ_ℓ^2 and the mean $s_\ell(x)$ is given by a unimodal spline function $s_\ell(x) = \sum_{j=-k}^{g} \beta_{\ell,j} N_{j,k+1}(x)$ with coefficients $\beta_{\ell,-k} \leq \ldots \leq \beta_{\ell,m_\ell-1} \leq \beta_{\ell,m_\ell} \geq \beta_{\ell,m_\ell+1} \geq \ldots \geq \beta_{\ell,g}$. That is, the parameter vector $\boldsymbol{\theta}$ is given by $\boldsymbol{\theta} = (\pi_1, \ldots, \pi_L, \beta_{1,-k}, \ldots, \beta_{L,g}, \sigma_1, \ldots, \sigma_L)$. This model does not only yield regression functions for all components, but can also classify each data point into the subgroups specified by the components. Mixture regression models can be fitted with the Expectation-Maximization algorithm (see, e.g., Dempster et al., 1977; Grün and Leisch, 2008) and are, for example, implemented in the R package `flexmix` (see Grün and Leisch, 2008). The number of components, L, can be determined with the help of a model selection criterion, for example, Akaike information criterion (see also Section 4.3.5).

A special case of this model has been investigated by Hannah Bürger in her Bachelor's thesis (Bürger, 2012), which comprises a simulation study to assess the performance of this modelling approach. The statistical methodology was proposed by the author of this Ph.D. thesis, who co-supervised the Bachelor's thesis. The model is designed for situations where the population consist of two subpopulations, one with a constant

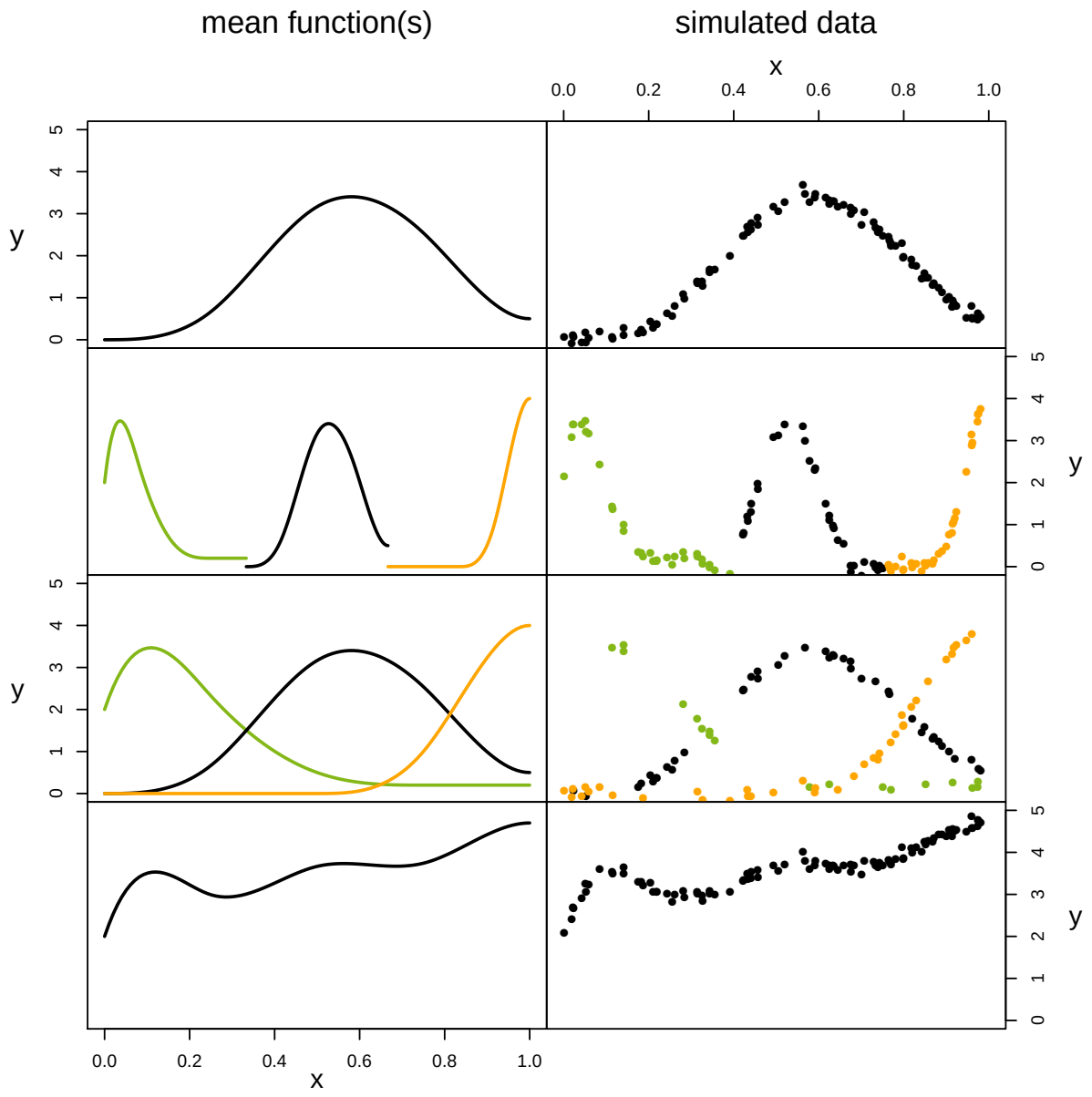


Figure 4.1: **Simulated examples.**
Mean function(s) (left column) and simulated data (right column) of the following models: unimodal spline, piecewise unimodal splines, mixture model with unimodal splines and convolution model with unimodal splines. We use three different coefficient sequences: $\boldsymbol{\beta}_{1\cdot} = (2, 4.5, 2, 0.2, 0.2, 0.2, 0.2)'$ (green), $\boldsymbol{\beta}_{2\cdot} = (0, 0, 0, 4, 3, 0.5, 0.5)'$ (black), $\boldsymbol{\beta}_{3\cdot} = (0, 0, 0, 0, 0, 4, 4)'$ (orange). The first row corresponds to a spline with coefficient sequence $\boldsymbol{\beta}_{2\cdot}$. For the mixture model in row three we employed mixing weights $\pi_1 = 0.2$, $\pi_2 = 0.4$, $\pi_3 = 0.4$ (cf. Equation (4.1)) and the parameter α of the additive model (cf. Equation (4.5)) in row four was set to 0. For each model 100 data points were simulated from normal distributions with the respective mean functions and standard deviation $\sigma = 0.1$.

predictor-response-relationship and one exhibiting a unimodal relationship. Such a scenario might occur, if a population is exposed to a sudden event and some entities react to it with an increase in the variable of interest and some do not. When the effect of the event wears off, the response falls back to a normal level.
Thus, the mixture model has two components and its conditional density is given by

$$p(Y = y|x, \boldsymbol{\theta}) = \pi_1 \ \varphi\left(y|s_1(x), \sigma_1^2\right) + \pi_2 \ \varphi\left(y|s_2(x), \sigma_2^2\right), \tag{4.2}$$

where the first regression function, $s_1(x) \equiv \beta_{1,0}$, comprises only an intercept (namely, a spline of degree 0 with 0 inner knots) and the second one is a unimodal spline $s_2(x) = \sum_{j=-k}^{g} \beta_{2,j} N_{j,k+1}(x)$, that is, $\boldsymbol{\theta} = (\pi_1, \beta_{1,0}, \beta_{2,-k}, \ldots, \beta_{2,g}, \sigma_1, \sigma_2)$. The findings of the Bachelor's thesis will be briefly summarised in Section 7.3.

4.3 Homogeneous population

For the case of a homogeneous study population there are several approaches to model multimodality. In general, the situation can be described as follows: Let $(x_i, y_i) \in [a, b] \times \mathbb{R}$ be pairs of observations that may be modelled by

$$Y = f(x) + \mathcal{E}. \tag{4.3}$$

In contrast to the above model for inhomogeneous populations, the conditional density of the random variable Y in this model is not a mixture of normal densities with unimodal mean functions but a single normal density with a multimodal mean function f, that is, $p(Y = y|x, \boldsymbol{\theta}) = \varphi(f(x|\boldsymbol{\theta}), \sigma^2)$. The mean function f will be specified in more detail by the different methods in the following subsections, which present piecewise unimodal regression and several deconvolution models. Recommendations on the method of choice are given in Section 4.3.6.

While the application of unimodal regression to several pieces of a data set is quite straightforward and will be briefly described in Section 4.3.1, representatives of the class of deconvolution models are very diverse and numerous. This statement still remains valid, if we make the assumption of linear convolution, that is, confining to cases where the observed global response is a linear combination of an unknown number of unobserved input processes (henceforth referred to as "peaks"). In explicit, the observed

response vector $\boldsymbol{y}$ can be written as

$$\boldsymbol{y} = \boldsymbol{S}\boldsymbol{a} + \boldsymbol{\epsilon}, \tag{4.4}$$

where $\boldsymbol{\epsilon}$ are the measurement errors, $\boldsymbol{a}$ are the coefficients of the linear combination and the columns of $\boldsymbol{S}$ describe the shapes of the peaks. The task is then to deconvolve the observed multimodal signal into the unobserved single peaks, that is, to estimate both the matrix $\boldsymbol{S}$ and the vector $\boldsymbol{a}$.

This deconvolution problem mathematically belongs to the class of inverse problems. Since numerous different combinations of $\boldsymbol{S}$ and $\boldsymbol{a}$ can explain the output equally well, the problem is ill-conditioned and the respective models are under-determined. To arrive at reasonable solutions nevertheless, various deconvolution algorithms have been proposed in the literature, which mostly use iterative schemes such as the Expectation-Maximization (EM) algorithm (see de Rooi and Eilers, 2011, for an overview).

In signal processing or chemometrics such approaches are often called blind source separation techniques. A widely used one is MCR, multivariate curve resolution, where "multivariate" refers to the fact that usually several contiguous output vectors are obtained and the algorithms are designed for direct application to the composite data matrix with model equation $\boldsymbol{Y} = \boldsymbol{S}\boldsymbol{A} + \boldsymbol{E}$. Different MCR methods are, for example, given in Tauler (1995), Pomareda et al. (2010) and Oller-Moreno et al. (2015). A drawback of those approaches is that the number of peaks is not estimated within the model, but by other methods previous to the actual analysis.

Bad conditioning and under-determination can generally be addressed by regularization and the use of constraints. Using positivity constraints in MCR, for example, leads to an approach called non-negative matrix factorization (cf. Pomareda et al., 2010). Other examples of constraints are unimodality or sparsity, the latter one being imposed by regularization in form of a LASSO/L_1-penalty (see, for example MCR-LASSO by Pomareda et al., 2010) or by an L_0-penalty (see de Rooi and Eilers, 2011). We follow de Rooi and Eilers (2011) in using the L_0-penalty, which allows for estimation of the number of peaks simultaneous to the other model parameters. Since their original model is restricted to the pointwise estimation of identically shaped peaks, we make some enhancements, for example, enabling the estimation of a functional description of identically shaped peaks. The most interesting development, however, is our combination of the deconvolution model using L_0-penalty with the "additive unimodal regression model" (see Equation (4.5)). The combined approach enables sparse deconvolution of

multimodal data using differently shaped unimodal spline components.

Apart from using penalties for regularization, the unimodality constraint is of great importance here. We will impose unimodality on spline functions throughout the next sections by using the penalized unimodal spline regression from Section 3.5. In many situations log-concave spline smoothing by Eilers (2005) can be used as well, but regarding the deconvolution model with varying peak shapes in Section 4.3.4 our approach is advantageous.

For the remainder of this chapter we suppose that an adequate estimate of the error variance $\sigma^2 > 0$ is available prior to model fitting.

4.3.1 Piecewise unimodal regression

A simple approach for modelling multimodal data is piecewise unimodal regression, that is, dividing the x-axis (heuristically) between each pair of modes and fitting separate unimodal splines. The function f from model (4.3) can be written as

$$f(x) = \sum_{\ell=1}^{L} s_\ell(x) \mathbb{1}_{I_\ell}(x),$$

where $I_1, \ldots, I_L \subset [a, b]$ are L intervals corresponding to the x-axis' pieces and s_ℓ are unimodal spline functions on the respective intervals. Depending on the application, there are different ways to determine these intervals, for example, using a threshold.
This model implies the assumption that the underlying process generating the observations is also divisible in some respects. We will refer to this modelling approach as pUniReg.
An example of a mean model function with three unimodal pieces and data simulated from this model is given in the second row of Figure 4.1.

4.3.2 Deconvolution with identical peak shapes using the L_0-penalty

Other multimodal regression approaches, so-called deconvolution models, describe the observations as a convolution of peaks. As opposed to the former approach, they allow modelling of overlapping peaks which accumulate at the overlap to the observed values,

but can also be applied, when no overlap is present. In this thesis, we examine only linear convolution. See row 4 of Figure 4.1 for an example of overlapping, unimodal functions and data simulated from their linear convolution.
Let us first assume that all peaks have the same basic shape and each peak is a scaled version of this shape. This can be expressed by writing the function f from model (4.3) as

$$f(x_i) = \sum_{k=1}^{n_s} s_k a_{i-k}, \ i = 1, \ldots, n,$$

where $\boldsymbol{s} = (s_1, \ldots, s_{n_s})'$ is the vector describing the known (pointwise) peak shape and $\boldsymbol{a} = (a_{-n_s+1}, \ldots, a_{n-1})' \in \mathbb{R}^{n+n_s-1}$ is the vector of the so-called input pulses, which describe the number of peaks (given by the number of values $a_j \neq 0$), their locations (given by index j) and heights (given by the actual value of a_j). The number n_s of x-points, for which the peak shape is given, is usually smaller than n and thus, the individual peaks do not span over the whole range of x-values. This model was introduced by de Rooi and Eilers (2011) and it was shown that it can also be reformulated as a typical linear regression model:

$$\boldsymbol{y} = \boldsymbol{S}\boldsymbol{a} + \boldsymbol{\epsilon},$$

where the convolution matrix $\boldsymbol{S} \in \mathbb{R}^{n \times (n+n_s-1)}$ holds shifted copies of the same peak shape $\boldsymbol{s}$ in its columns (compare also to the general linear deconvolution model in Equation (4.4)).
If the peak shape $\boldsymbol{s}$ is known, the least squares estimate of $\boldsymbol{a}$ is given by $\hat{\boldsymbol{a}} = (\boldsymbol{S}'\boldsymbol{S})^{-1}\boldsymbol{S}\boldsymbol{y}$, but the columns of $\boldsymbol{S}$ are highly correlated, which leads to ill-posedness. This problem was already described in de Rooi and Eilers (2011) and the authors propose to use regularization with an L_0-penalty on $\boldsymbol{a}$, that is, using the objective function

$$\|\boldsymbol{y} - \boldsymbol{S}\boldsymbol{a}\|_2^2 + \kappa \sum_k |a_k|^0.$$

The regularized estimate for $\boldsymbol{a}$ is found by minimizing the objective function using an iterative procedure (see the description of the implementation in Section 5.3.1 for details). Since the penalty factor $\sum_j |a_j|^0$ is essentially the number of peaks, the regularized estimation favours sparse models with only few peaks. The higher the tuning parameter κ, the fewer the peaks. de Rooi and Eilers (2011) propose to choose the tuning param-

eter κ by visual inspection. They also describe an additional, asymmetric penalty on $\boldsymbol{a}$ that renders the estimated input pulses positive. Altogether, the procedure is able to estimate the number of peaks, their locations and heights simultaneously. We will refer to this modelling approach as L_0-deco.
de Rooi and Eilers (2011) additionally present an approach for cases where the peak shape $\boldsymbol{s}$ is unknown, which is called "blind deconvolution". The idea is, starting with an initial pointwise peak shape $\boldsymbol{s}^{(0)}$, to iterate between estimation of $\boldsymbol{a}$ and $\boldsymbol{s}$. Given an (interim) estimate of $\boldsymbol{a}$ a new estimate of $\boldsymbol{s}$ can be found in three different ways. The first one is described in de Rooi and Eilers (2011) and produces a pointwise estimate of $\boldsymbol{s}$. It can be found using the reformulated model

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{\epsilon}.$$

Here, $\boldsymbol{A}$ holds shifted copies of $\hat{\boldsymbol{a}}$ in its columns and the pointwise least squares estimate is given by $\hat{\boldsymbol{s}} = (\boldsymbol{A}'\boldsymbol{A})^{-1}\boldsymbol{A}\boldsymbol{y}$ (cf. de Rooi and Eilers, 2011). It is also possible to use a smoothness penalty (differences penalty or unimodal smoother) on the entries of $\boldsymbol{s}$ (cf. de Rooi et al., 2014). We will refer to this approach as pointwise L_0-deco.
In this thesis we want to estimate not only a smooth pointwise description of each peak, but also a continuous functional peak shape. de Rooi and Eilers (2011) already mentioned the possibility to use spline functions for this purpose. Here, we give details for a slightly more general approach, where the peak shape is given by a function $s(x|\boldsymbol{\beta})$ parameterized by vector $\boldsymbol{\beta}$. Either the peak shape is known to (approximately) follow a unimodal parametric function, where the parameters in $\boldsymbol{\beta}$ are usually few and nicely interpretable. Or the function s can be a semi-parametric unimodal spline function, if there is no prior information about the peak shape. In both cases, function f from model (4.3) can be written as

$$f(x_i) = \sum_{j=1}^{n_s} s(x_j|\boldsymbol{\beta})a_{i-j},\ i = 1, \ldots, n,$$

and the following estimation procedure can be applied:
With an initial parameter vector $\boldsymbol{\beta}^{(0)}$ and the respective initial peak shape $\boldsymbol{S}^{(0)} = (s(x_1|\boldsymbol{\beta}^{(0)}), \ldots, s(x_{n_s}|\boldsymbol{\beta}^{(0)}))'$ it is again possible to iterate between estimation of $\boldsymbol{a}$ and $\boldsymbol{\beta}$. In the k-th iteration, estimation of $\boldsymbol{a}$ is done as described above using the peak shape $\hat{\boldsymbol{s}}^{(k-1)} = (s(x_1|\hat{\boldsymbol{\beta}}^{(k-1)}), \ldots, s(x_{n_s}|\hat{\boldsymbol{\beta}}^{(k-1)}))'$, where $\hat{\boldsymbol{\beta}}^{(k-1)}$ the current estimate of $\boldsymbol{\beta}$. The parameter vector $\boldsymbol{\beta}$ can be estimated using the least squares method, that is, minimiz-

ing $||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{s}||^2 = ||\boldsymbol{y} - \boldsymbol{A}(s(x_1|\boldsymbol{\beta}), \ldots, s(x_{n_s}|\boldsymbol{\beta}))'||^2$ with respect to $\boldsymbol{\beta}$, or with unimodal regression. These two blind deconvolution approaches will be referred to as parametric L_0-deco and unimodal L_0-deco. Except for the initial guess, no information about the basic peak shape is needed for the unimodal L_0-deco approach. All blind deconvolution approaches simultaneously obtain estimates of a basic peak shape, the number, locations and heights of the peaks. The parametric and unimodal L_0-deco have the advantage of a functional shape description in contrast to the pointwise L_0-deco approach.

4.3.3 Deconvolution with diverse peak shapes: additive unimodal regression

While there are several applications in which the assumption of the same shape for all peaks is very plausible (for example, the FACT data), there are also situations where the observed signal is a convolution of peaks with different shapes (for example, in IMS data). Eilers (2005) used sums of log-concave smoothing splines for such a deconvolution task. More generally speaking, an appropriate representative of deconvolution models in this situation is an additive model, that describes the observations as convolution of L different peak shapes. In this model the function f from Equation (4.3) is given by

$$f(x_i) = \alpha + \sum_{\ell=1}^{L} s_\ell(x_i), \tag{4.5}$$

where α is an intercept and each $s_\ell(x)$ is a unimodal function describing one of the peaks and can be evaluated over the whole range of the x-observations. Eilers (2005) used log-concave smoothing splines for each s_ℓ, but of course each of these functions can be described by a parametric model (with different parameter values) or by unimodal spline regression (with different parameters and modes) as well. For the latter choice we have $s_\ell(x) = \sum_{j=-k}^{g} \beta_{\ell,j} N_{j,k+1}(x)$ with coefficients $\beta_{\ell,-k} \leq \ldots \leq \beta_{\ell,m_\ell-1} \leq \beta_{\ell,m_\ell} \geq \beta_{\ell,m_\ell+1} \geq \ldots \geq \beta_{\ell,g}$.
Additive models can be fitted using the so-called backfitting algorithm (cf. Hastie et al., 2009), which is given by

1. Initialize $\hat{\alpha} = \frac{1}{n}\sum_{i=1}^{n} y_i$ and $\hat{s}_\ell(x) \equiv 0 \ \forall \ell$.
2. For $\ell = 1, \ldots, L$: calculate $\hat{s}_\ell$ from data $(x_i, \tilde{y}_i)$ with $\tilde{y}_i = y_i - \hat{\alpha} - \sum_{\mathcal{K} \neq \ell} \hat{s}_\mathcal{K}(x_i)$, $i = 1, \ldots, n$.
3. Centre the function estimates around zero: Set $\hat{s}_\ell := \hat{s}_\ell - \frac{1}{n}\sum_{i=1}^{n} \hat{s}_\ell(x_i)$.

4. Repeat steps 2 and 3 until convergence.

In contrast to commonly applied additive models (see e.g. Hastie et al., 2009) we have only one regressor that is used in all components. Thus, the number L of components is not simply the number of regressors. Sometimes the specific application might enforce a fixed number of components or it can be determined with the help of a model selection criterion, for example, the Akaike information criterion (AIC; cf. Section 4.3.5). We will refer to this modelling approach as addUniReg.
The proposed methodology was also used in the Master's thesis of Laura Lange (Lange, 2015), which was co-supervised by the author of this Ph.D. thesis. The results of the Master's thesis obtained on 119 IMS measurements are shortly summarised in Section 7.3.

4.3.4 Deconvolution with diverse peak shapes: combining additive unimodal regression and L_0-deconvolution

In principle, the addUniReg model is applicable in all (linear) deconvolution tasks since it is the most general model. This flexibility comes, for example, at the cost of higher computation times, because each component is estimated with unimodal regression (involving determination of the mode by trying all possibilities) and the number of components can only be determined by fitting several models and making a choice based on AIC. Thus, the L_0-deconvolution model is preferable, since it simultaneously estimates the number, the locations and heights of peaks. Yet, it cannot cope with different peak shapes.
de Rooi et al. (2014) already discussed the fact that the peak shape might vary over time, or more generally speaking, that the peaks might have different shapes. Two concepts were proposed to tackle such problems with L_0-deconvolution: finding a transformation of the x-axis such that the peak shape is constant or estimating the matrix $\boldsymbol{S}$ (holding the varying peak shapes in its columns) as a smooth two-dimensional surface. For the first proposal, knowledge about the way in which the peak shape changes over time would be required and this is not the case for the applications we have in mind (for example, IMS data). The here presented approach is in line with the second suggestion, although the surface will only be smooth in one of its directions (each column holds a smooth peak shape). It can be seen as a combination of unimodal L_0-deco and the additive unimodal regression.
Suppose we have response values $\boldsymbol{y}$ with a baseline close to zero (more explicit, with minimal value equal to zero) and several peaks with maximum peak height equal to one

(data can easily be transformed to fulfil these criteria). For this model the function f from Equation (4.3) can be written as follows:

$$f(x_i) = \sum_{j=1}^{d} s_j(x_i)a_j,$$

where $s_j(.)$ is a spline function with d B-spline coefficients, which have a fixed mode at j, and a_j is the input pulse corresponding to the $j-th$ peak, $j \in \{1, \ldots, d\}$. In explicit, this is an additive model with d spline components that have different mode locations and are scaled by the input pulses. The matrix of varying peak shapes $\boldsymbol{S}$ then consists of the values $s_{ij} = s_j(x_i), i = 1 \ldots, n, j = 1, \ldots, d$. The input pulses $\boldsymbol{a}$ have a slightly different role in this model compared to the original deconvolution model. Each input pulse corresponds to one of at most d peaks, where d is the number of B-spline coefficients which is usually much smaller than n, but also larger than the number of existent peaks. Thereby the index j does not correspond to observation x_j and the peak locations cannot be derived directly, but the model still provides simultaneous estimation of number and heights of the peaks and additionally estimates the different peak shapes. The ability to estimate spline functions with a fixed mode is very essential here. This could not be achieved as easily with other unimodal regression approaches like, for example, log-concave smoothing by Eilers (2005).
The model can be fitted by iteratively estimating the s_j with steps similar to the backfitting algorithm for the additive model and estimating the input pulses $\boldsymbol{a}$ using the L_0-penalty. The estimation procedure is described in Section 5.3.4. The alternating estimation of matrix $\boldsymbol{S}$ and input pulses $\boldsymbol{a}$ is repeated until the position and the height of the positive input pulses stabilizes. In our applications we found that 10 repetitions typically result in sufficiently converged models. We will refer to this model as varying L_0-deco.

4.3.5 Model selection and effective degrees of freedom

As indicated in 4.2 and 4.3.3, a choice has to be made regarding the number of components in both the mixture and the additive model. This is a model selection problem. There exist a number of different model selection criteria that compare different models on the same data and take, for example, different numbers of parameters into account.

For example, the commonly used Akaike information criterion is given by

$$AIC = n \log\left(\frac{RSS}{n}\right) + 2K,$$

where $RSS = \sum_{i=1}^{n} \hat{\epsilon_i}^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$ is the residual sum of squares and K is the effective dimension of the model (see e.g. Fahrmeir et al., 2013, ch. 3.6). If each component is estimated with a parametric approach, K is just the total number of parameters. If each component is estimated with a penalized spline regression, K is the sum of all individual effective dimensions.
The effective dimension of a penalized spline regression is obtained as follows:

Lemma 4. *Suppose that spline regression is performed according to the penalized least squares objective function*

$$\frac{1}{\sigma^2} \|\boldsymbol{y} - \boldsymbol{B}\boldsymbol{\beta}\|_2^2 + \lambda \left\| \boldsymbol{\Omega}^{\frac{1}{2}} (\boldsymbol{\beta} - \boldsymbol{\beta}_0) \right\|_2^2$$

as in Section 3.3 with fixed values of λ, $\boldsymbol{\beta}_0$ and a positive definite matrix $\boldsymbol{\Omega}$.
Then the effective dimension of the regression model is given by

$$ed = tr\left(\frac{1}{\sigma^2}\boldsymbol{B}'\boldsymbol{B}\left(\frac{1}{\sigma^2}\boldsymbol{B}'\boldsymbol{B} + \lambda\boldsymbol{\Omega}\right)^{-1}\right).$$

Proof of Lemma 4. Following Ye (1998) the effective dimension of a regression model is given by $ed = \sum_{i=1}^{n} \frac{\partial \hat{y}_i}{\partial y_i}$, which is equal to $tr(\boldsymbol{H})$ if $\hat{\boldsymbol{y}} = \boldsymbol{H}\boldsymbol{y}$ (e.g. in linear regression). For unconstrained penalized spline regression with penalty vector $\boldsymbol{\beta}_0 = \boldsymbol{0}$, the hat matrix is given by $\tilde{\boldsymbol{H}} = \frac{1}{\sigma^2}\boldsymbol{B}\left(\frac{1}{\sigma^2}\boldsymbol{B}'\boldsymbol{B} + \lambda\boldsymbol{\Omega}\right)^{-1}\boldsymbol{B}'$ and thus, $ed = tr(\tilde{\boldsymbol{H}})$.
Suppose now we conduct an unconstrained penalized spline regression with penalty vector $\boldsymbol{\beta}_0 \neq \boldsymbol{0}$. Then

$$\begin{aligned}
\hat{\boldsymbol{y}} &= \boldsymbol{B}\hat{\boldsymbol{\beta}} \underset{(3.2)}{=} \boldsymbol{B}\left(\frac{1}{\sigma^2}\boldsymbol{B}'\boldsymbol{B} + \lambda\boldsymbol{\Omega}\right)^{-1}\left(\frac{1}{\sigma^2}\boldsymbol{B}'\boldsymbol{y} + \lambda\boldsymbol{\Omega}\boldsymbol{\beta}_0\right) \\
&= \frac{1}{\sigma^2}\boldsymbol{B}\left(\frac{1}{\sigma^2}\boldsymbol{B}'\boldsymbol{B} + \lambda\boldsymbol{\Omega}\right)^{-1}\boldsymbol{B}'\boldsymbol{y} + \underbrace{\lambda\boldsymbol{B}\left(\frac{1}{\sigma^2}\boldsymbol{B}'\boldsymbol{B} + \lambda\boldsymbol{\Omega}\right)^{-1}\boldsymbol{\Omega}\boldsymbol{\beta}_0}_{\text{constant w.r.t. } \boldsymbol{y}} \\
&= \tilde{\boldsymbol{H}}\boldsymbol{y} + \boldsymbol{c}.
\end{aligned}$$

We can conclude that

$$\begin{aligned} ed = \sum_{i=1}^{n} \frac{\partial \hat{y}_i}{\partial y_i} &= \sum_{i=1}^{n} \frac{\partial (\tilde{\boldsymbol{H}}\boldsymbol{y} + \boldsymbol{c})_i}{\partial y_i} \\ &= \sum_{i=1}^{n} \left(\frac{\partial (\tilde{\boldsymbol{H}}\boldsymbol{y})_i}{\partial y_i} + \frac{\partial c_i}{\partial y_i} \right) \\ &= \sum_{i=1}^{n} \left(\frac{\partial (\tilde{\boldsymbol{H}}\boldsymbol{y})_i}{\partial y_i} + 0 \right) = tr(\tilde{\boldsymbol{H}}), \end{aligned}$$

which is the effective dimension of the corresponding regression problem with $\boldsymbol{\beta}_0 = \mathbf{0}$. From properties of the matrix rank it follows that

$$ed = tr(\tilde{\boldsymbol{H}}) = tr\left(\tfrac{1}{\sigma^2}\boldsymbol{B}(\tfrac{1}{\sigma^2}\boldsymbol{B}'\boldsymbol{B} + \lambda\boldsymbol{\Omega})^{-1}\boldsymbol{B}'\right) = tr\left(\tfrac{1}{\sigma^2}\boldsymbol{B}'\boldsymbol{B}(\tfrac{1}{\sigma^2}\boldsymbol{B}'\boldsymbol{B} + \lambda\boldsymbol{\Omega})^{-1}\right),$$

which is the trace of a matrix of dimension $d \times d$ instead of $n \times n$ and thus faster computable (cf. Fahrmeir et al., 2013, ch. 8.1.8). □

For constrained regression there is no explicit expression for $\hat{\boldsymbol{y}}$ and thus a similarly easy way of calculating the effective dimension is not available. Hence, we propose to approximate the effective dimension of a unimodal spline regression with the effective dimension of the corresponding unconstrained problem given in Lemma 4. We think that for our purposes this approximation does not influence the model choice dramatically since all compared models contain solely unimodal components.
If the penalty matrix $\boldsymbol{\Omega}$ is not positive definite, $\lambda\boldsymbol{\Omega}$ has to be replaced by the alternative penalty matrix $\tilde{\boldsymbol{\Omega}}_\lambda$ as described in Section 3.5.2.

Now let ed_ℓ be the (approximated) effective dimension of the spline of component ℓ. Then the AIC of the mixture or additive model is given by

$$n \log\left(\frac{RSS}{n}\right) + 2\sum_{\ell=1}^{L} ed_\ell$$

and can be used to determine an appropriate number of components.
A similar model selection criterion derived in the Bayesian context is

$$n \log\left(\frac{RSS}{n}\right) + \log(n)K,$$

the Bayesian information criterion (BIC; cf. Fahrmeir et al., 2013, ch. 3.6). Here, K is again the effective model dimension. In comparison to AIC, the BIC usually favours sparser models since high effective dimensions are penalized stronger as soon as $n > 7$.

4.3.6 Applicability of the model types

The following Table 4.1 gives an overview of the different data situations with homogeneous population, where the proposed approaches to multimodal regression are applicable. Depending on the shape of the peaks and their overlap the table states the recommended model. In principle, the addUniReg and varying L_0-deco models are applicable in all mentioned situations since they cope with the most general situation of overlapping peaks with diverse peak shapes. For this flexibility one has to pay the prize of high computation times (each peak is estimated over the whole range of the x-values), especially for the addUniReg model, because the number of components can only be determined by fitting several models and making a choice based on AIC. Thus, both approaches should only be considered, when there are overlapping peaks with diverse shapes. The varying L_0-deco model is to be preferred, if the number of peaks is unknown. If there are differently shaped, but non-overlapping peaks, the simpler pUniReg approach is sufficient, which is in principle also "downwards compatible" to situations with identical peak shapes. Nevertheless, the method of choice when all peaks have the same shape and regardless if there is overlap or not, is the L_0-deconvolution model, since it simultaneously estimates the number, the locations and heights of peaks. When the peak shape is unknown, blind deconvolution can be used and we propose to apply the advanced versions, parametric and unimodal L_0-deco, instead of pointwise L_0-deco to obtain smooth function estimates.
As the derivatives of the fitted peaks are also of interest in some applications (see, for example, the diving depth data analysis in Section 7.2.1), it is important to note that derivatives are easily obtained with all approaches that use (unimodal) splines.

Table 4.1: **Overview of the proposed multimodal regression approaches and in which situation to use them.**
The recommended model is marked by boldface. The abbreviations are as follows: L_0-deco: L_0-deconvolution model with fixed peak shape, pointwise L_0-deco: blind L_0-deconvolution model with pointwise peak shape, parametric L_0-deco: blind L_0-deconvolution model with parametric peak shape, unimodal L_0-deco: blind L_0-deconvolution model with unimodal peak shape, pUniReg: piecewise unimodal regression, addUniReg: additive unimodal regression, varying L_0-deco: blind L_0-deconvolution model with diverse unimodal peak shapes.

		no overlap	overlap
peak shapes	identical, known	**L_0-deco**	**L_0-deco**
	identical, unknown	pointwise L_0-deco *or* parametric L_0-deco *or* **unimodal L_0-deco**	pointwise L_0-deco *or* parametric L_0-deco *or* **unimodal L_0-deco**
	diverse, unknown	**pUniReg**	addUniReg *or* **varying L_0-deco**

5 Implementation

This chapter presents computational aspects for the methods proposed in Chapters 3 and 4. All functions used for the simulation study in Chapter 6 and the applications in Chapter 7 are implemented in the statistical software environment R (R Core Team, 2016). We concentrate here on the main functions, which perform the actual model fitting. The implementational details of auxiliary functions can be found in Appendix D of this thesis. The descriptions are provided in a style similar to that of the R help system, though with more details than usual.

5.1 The R package `uniReg`

The frequentist approaches presented in Sections 3.3 to 3.5 are implemented in the R package `uniReg` (Köllmann, 2016). The package in version 1.1 is freely available under the GPLv3 license on CRAN:

`http://cran.r-project.org/web/packages/uniReg/index.html`.

The package can be installed and loaded in a running R session by execution of

```
install.packages("uniReg")
library(uniReg)
```

The functions of the package that are directly accessible for users, `unireg`, `equiknots`, `unimat`, `plot.unireg` `points.unireg`, `predict.unireg` and `print.unireg`, are documented in the reference manual of the package available at the above web page and can be retrieved within R's help system via

```
help(package="uniReg")
```

Additionally, the main function `unireg` will be described in the following and the remaining (auxiliary) functions are presented in Appendix D. The functions `negloglikFREQ`,

`unimatind` and `unisplinem` of the R package are internal and no corresponding help pages are included in the reference manual. Their documentation can also be found in Appendix D.

`unireg` *Fitting a unimodal penalized spline regression.*

Description

Function for fitting spline regressions to data. The fit can be constrained to be unimodal, inverse-unimodal, isotonic or antitonic and an arbitrary penalty on the B-spline coefficients can be used.

Usage

```
unireg(x, y, w=NULL, sigmasq=NULL, a=min(x), b=max(x), g=10, k=3,
constr=c("unimodal","none","invuni","isotonic","antitonic"),
penalty=c("diff", "none", "sigEmax", "self", "diag"), Om=NULL,
beta0=NULL, coinc=NULL, tuning=TRUE, abstol=0.01, vari=5, ordpen=2,
m=1:(g+k+1), allfits=FALSE, nCores=1)
```

Arguments

`x`	A numeric vector of x-values, length n. Contains at least $d = g + k + 1 \leq n$ distinct values.
`y`	A numeric vector of observed y-values of length n.
`w`	A positive numeric weight vector of length n. The weights do not have to sum to n, but will be transformed to do so internally. If `sigmasq` is given, `w` should be `NULL` (default).
`sigmasq`	Estimate(s) of the residual variance(s). Can be a positive numeric vector of length n, giving estimates for the variance at each of the x-values. If it is a vector of length 1, equal variances across all x-values are assumed. If `sigmasq=NULL` (default), each x-value has to be appear at least twice and a global variance (same for all x-values) is estimated internally. If `sigmasq` is given, `w` should be `NULL`.

`a`	The left numeric boundary of the interval, on which the spline is defined. If `coinc=TRUE`, the spline is zero to the left of this value. By default `a` is equal to the minimal x-value.
`b`	The numeric right boundary of the interval, on which the spline is defined. If `coinc=TRUE`, the spline is zero to the right of this value. By default `b` is equal to the maximal x-value.
`g`	A non-negative integer giving the number of inner knots of the spline (default: 10).
`k`	A non-negative integer specifying the degree of the spline. By default a cubic spline (`k=3`) is fitted.
`constr`	A character string specifying the shape constraint for the fit. Can be one of `"unimodal"` (default), `"none"`, `"invuni"` (inverse-unimodal), `"isotonic"`, `"antitonic"`.
`penalty`	A character string specifying, which penalty on the B-spline coefficients should be used. Possible choices are `"diff"` (default) for the differences penalty of order `ordpen`, `"none"` for no penalty, `"sigEmax"` for the sigmoid E_{max} penalty, `"self"` for a self-defined penalty and `"diag"` for a ridge penalty (see also Table 3.1). For a self-defined penalty `Om` and `beta0` have to be provided.
`Om`	If a self-defined penalty on the B-spline coefficients is used, `Om` is the penalty matrix of dimension $d \times d$ and full rank d. Otherwise `Om` should be `NULL` (default).
`beta0`	If a self-defined penalty on the B-spline coefficients is used, `beta0` is the penalty vector of length d. Otherwise `beta0` should be `NULL` (default).
`coinc`	Logical indicating, if the outer knots of the knot sequence should be coincident with the boundary knots or not. Default is `NULL` and altering has no effect, if a pre-defined penalty is used. If `penalty="self"`, it has to be specified.

`tuning` Logical indicating, if the tuning parameter lambda should be optimized with (`tuning=TRUE`, default, computationally expensive) or without (`tuning=FALSE`) consideration of the shape constraint. Changing `tuning` has no effect, when `constr="none"` or `penalty="none"`.

`abstol` The iterative estimation of the residual variance σ^2 and the coefficient vector $\boldsymbol{\beta}$ stops after iteration ς, when $|\hat{\sigma}^{(\varsigma)} - \hat{\sigma}^{(\varsigma-1)}|$ is less than a positive numeric value `abstol` (default: 0.01) or when $\varsigma = 10$. If `sigmasq` is not `NULL`, the supplied value is used as starting value in this iteration scheme. There is no iterative estimation, if `abstol` is set to `NULL`.

`vari` Variance parameter $\sigma_v^2 > 0$ (default 5) in the full-rank penalty matrix $\tilde{\boldsymbol{\Omega}}_\lambda$ for cases $\text{rank}(\boldsymbol{\Omega}) < d$.

`ordpen` Order of the difference penalty (integer ≥ 0, default 2). Only effective, if `penalty="diff"`.

`m` An integer vector specifying the modes of the coefficient vector which should be used for fitting, in explicit, a subset of $\{1, \ldots, d\}$. This argument only has an effect if `constr="unimodal"` or `"invuni"`.

`allfits` Logical indicating if the estimated coefficient vectors for all modes in `m` should be returned (`TRUE`) or only the one with minimal residual sum of squares (`FALSE`).

`nCores` The integer number of cores used for parallelization. If `nCores=1`, there is no parallelization (default).

Details

The function `unireg` is the main function of the package and implements the different frequentist spline regression approaches described in Chapter 3. Different shape constraints are possible and an arbitrary penalty on the B-spline coefficients can be used.

The vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ of length n hold the observed x- and y-values, for which the spline regression is to be calculated. A weighted regression is possible by specifying a length n vector $\boldsymbol{w}$ of non-negative weights. If a variance estimate is available prior to model fitting, this estimate can be supplied. If `sigmasq` is provided, $\boldsymbol{w}$ should be `NULL` and the other way round since the variances are internally used

to create a vector of weights. Thus, if weighted regression with inverse variances is desired, `sigmasq` should be supplied instead of $\boldsymbol{w}$.
The procedure starts with sorting $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{w}$ and (where required) `sigmasq` according to $\boldsymbol{x}$. If `sigmasq = NULL`, an initial estimate of the variance is obtained as follows: Let u be the number of distinct x-values, $\tilde{x}_1, \ldots, \tilde{x}_u$, and let $I_t = \{i : x_i = \tilde{x}_t\} \subset \{1, \ldots, n\},\ t = 1, \ldots, u$, be the sets of the corresponding indices. We determine a variance estimate according to the weighted formula,

$$\hat{\sigma}^2 = \frac{1}{\sum_{i=1}^{n} w_i} \sum_{t=1}^{u} WSS_t,$$

where $WSS_t = \sum_{i \in I_t} w_i \left(y_i - \bar{y}_t\right)^2$ is the weighted sum of squared deviations from the weighted mean $\bar{y}_t = \frac{\sum_{i \in I_t} w_i y_i}{\sum_{i \in I_t} w_i}$ in set I_t.
Afterwards, the y-values are transformed to lie in $[-1, 1]$ by $\frac{\boldsymbol{y} - shift}{scale}$, where $scale = 0.5 * (\max(\boldsymbol{y}) - \min(\boldsymbol{y}))$ and $shift = \min(\boldsymbol{y}) + scale$. The (provided or estimated) variance parameter `sigmasq` is multiplied accordingly by $\frac{1}{scale^2}$.
When `penalty` is not `"self"`, the penalty matrix $\boldsymbol{\Omega}$ and vector $\boldsymbol{\beta}_0$ have to be created. The ridge penalty matrix (`"diag"`) is given by $\boldsymbol{\Omega} = \boldsymbol{I}_d$. If `penalty = "diff"` the finite differences matrix $\boldsymbol{D}_q$, where $\boldsymbol{\Omega} = \boldsymbol{D}_q' \boldsymbol{D}_q$, can be calculated from an identity matrix with the help of the function `diff` (package `base` R Core Team, 2016) with respective order $q =$ `ordpen`. For the sigmoid E_{max} increase penalty `ordpen` is set to one. The penalty vector $\boldsymbol{\beta}_0$ is just a vector of zeros for all predefined penalties except the sigmoid E_{max} increase penalty, where it is determined by fitting a sigmoid E_{max} model to the data via the function `fitMod` from package `DoseFinding` (Bornkamp et al., 2016). The model predictions at the knot averages (calculated with function `knotave` from package `SEL` Bornkamp, 2010) are used as entries of $\boldsymbol{\beta}_0$ according to Equation (3.3). Additionally, a $d \times d$ matrix $\tilde{\boldsymbol{D}}$ is created, from which the full-rank matrix $\tilde{\boldsymbol{\Omega}}_\lambda$ can be calculated as $\tilde{\boldsymbol{\Omega}}_\lambda = \tilde{\boldsymbol{D}} + \lambda \boldsymbol{\Omega}$. Thus, $\tilde{\boldsymbol{D}}$ is a zero matrix, if $\boldsymbol{\Omega}$ already has full rank, and $\tilde{\boldsymbol{D}} = \frac{1}{\sigma_v^2} \boldsymbol{I}_d$ otherwise.
If a self-defined penalty is used, the penalty vector $\boldsymbol{\beta}_0$ is transformed in the same way as the observations: $\frac{\boldsymbol{\beta}_0 - shift}{scale}$.
The argument `coinc` is automatically set to `TRUE` except for `penalty = "diff"` and `penalty = "self"`.
The values of `a`, `b`, `g`, `k` and `coinc` are used to create the knot sequence with the function `equiknots` (cf. Appendix D). The matrix $\boldsymbol{B}$ of B-spline basis functions of degree k on the interval $[a, b]$ with g inner knots, evaluated at the observed values

$\boldsymbol{x}$, is determined with the function `splineDesign` from package `splines` (R Core Team, 2016).
The different constraints are realized by the choice of the constraint matrix $\boldsymbol{C}$ during the procedure. A new variable `inverse`, which is set to 1 for isotonic and unimodal fits and -1 for antitonic and inverse unimodal fits is created and the variable `constr` is re-set to `"unimodal"` or `"isotonic"`, respectively. Then, for `constr = "isotonic"` the constraint matrix is given by $\boldsymbol{C} = \texttt{inverse} \cdot \boldsymbol{C}_g$ and for `constr = "unimodal"` we cycle through each possible mode with $\boldsymbol{C} = \texttt{inverse} \cdot \boldsymbol{C}_{-k}, \ldots, \texttt{inverse} \cdot \boldsymbol{C}_g$, where the matrices $\boldsymbol{C}_m$ are determined with the help of the function `unimat` (cf. Appendix D). In explicit, the constraints `"antitonic"` and `"invuni"` are realized by using the negative of the isotonic and unimodal constraint matrices.

Before entering a `repeat`-loop for the iteration between estimation of $\boldsymbol{\beta}$ and σ^2, the matrices and vectors $\boldsymbol{\beta}_0'\boldsymbol{\Omega}\boldsymbol{\beta}_0$, $\boldsymbol{\beta}_0'\boldsymbol{\Omega}$, $\boldsymbol{B}'diag(\boldsymbol{w})\boldsymbol{B}$, $\boldsymbol{y}'diag(\boldsymbol{w})\boldsymbol{B}$ and the rank of $\boldsymbol{\Omega}$ are calculated to avoid their repeated computation. Inside the loop, $\boldsymbol{B}'diag(\boldsymbol{w})\boldsymbol{B}$ and $\boldsymbol{y}'diag(\boldsymbol{w})\boldsymbol{B}$ are then divided by the current variance estimate.
If the constraint is `"unimodal"` either a cluster for parallel calculations is set up (if `nCores > 1`) and `parLapply` from package `parallel` (R Core Team, 2016) is used or the standard `lapply` function is used to cycle through all possibilities of the mode in $\boldsymbol{m} \subset \{1, \ldots, d = g + k + 1\}$. For each mode, the corresponding unimodal regression is conducted with the help of the function `unisplinem` (cf. Appendix D), which estimates the B-spline coefficients with a fixed mode m, and the regression with minimal weighted residual sum of squares (wRSS) is determined.
For the monotone constraints or no shape constraint, the function `unisplinem` is only applied with mode $m = g$.
The estimated coefficients, fitted values and tuning parameter are stored and the effective degrees of freedom are calculated according to Section 4.3.5.
If `abstol = NULL`, the `repeat`-loop is stopped. Otherwise a new estimate of σ^2 is calculated from the residuals $r_i = y_i - \hat{y}_i,\ i = 1, \ldots, n$, using the weighted estimator

$$\hat{\sigma}^2 = \frac{1}{\sum_{i=1}^n w_i} \sum_{i=1}^n w_i \left(r_i - \frac{1}{n} \sum_{i=1}^n w_i r_i \right)^2$$

before a new estimate of $\boldsymbol{\beta}$ is obtained in the above manner. The iterative procedure stops after iteration ς, when $|\hat{\sigma}^{(\varsigma)} - \hat{\sigma}^{(\varsigma-1)}| <$ `abstol` or when $\varsigma = 10$.
If `allfits = TRUE` the estimated coefficient vectors for all modes in $\boldsymbol{m}$ are scaled

back the original scale of the data by multiplying with *scale* and adding *shift* and are stored in a matrix `allcoefs`. Otherwise `allcoefs=NULL`.
The function returns an object of class "unireg", where the estimated coefficients and fitted values have also been scaled back to the original data scale as above.

Value

Returns an object of class "unireg", that is, a list containing the following components:

`x`	The (sorted) vector of x-values.
`y`	The input vector of y-values (sorted according to x).
`w`	The vector of weights used for fitting (sorted according to x).
`a`	The left boundary of the domain $[a, b]$.
`b`	The right boundary of the domain $[a, b]$.
`g`	The number g of inner knots.
`degree`	The degree k of the spline.
`knotsequence`	The sequence of knots (length $g+2k+2$) used for spline fitting.
`constr`	The constraint on the coefficients.
`penalty`	The type of penalty used.
`Om`	The penalty matrix.
`beta0`	The penalty vector.
`coinc`	The input parameter `coinc`.
`tuning`	The input parameter `tuning`.
`abstol`	The input value of `abstol`.
`vari`	The input variance parameter σ_v^2.
`ordpen`	The order of the difference penalty.
`coef`	The vector of estimated B-spline coefficients (corresponding to the mode with minimal RSS).
`fitted.values`	The fitted values at each x-value (corresponding to the mode with minimal RSS).
`lambdaopt`	The optimal tuning parameter found via REML (corresponding to the mode with minimal RSS).
`sigmasq`	The estimated residual variance. If the input for `abstol` was `NULL`, `sigmasq` equals its input value.
`variter`	The number ς of iterations used to estimate the spline coefficients and the variance.

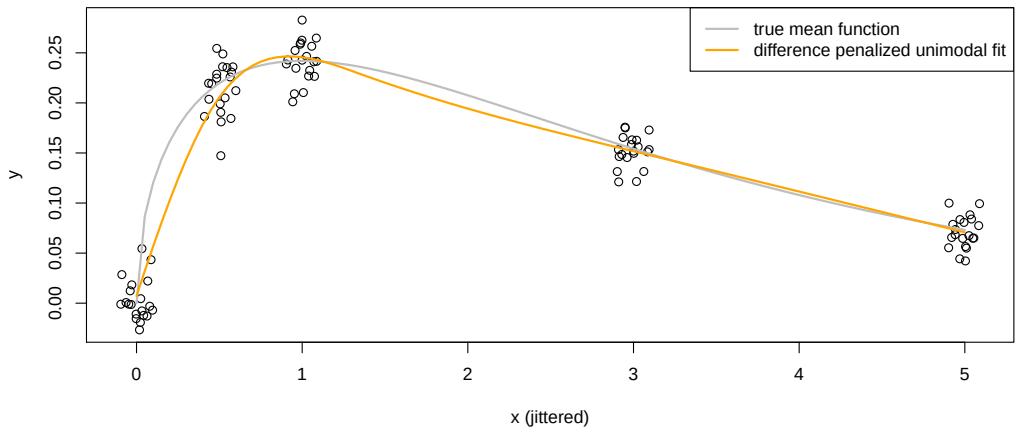
`ed`	The effective degrees of freedom (corresponding to the mode with minimal RSS).
`modes`	The input vector `m` of modes.
`allcoefs`	The object `allcoefs` (either a matrix of coefficient vectors or `NULL`).

Example

```
# generate some data
x <- sort(rep(c(0,0.5,1,3,5),20))
set.seed(41333)
func <- function(mu){rnorm(1,mu,0.02)}
y <- sapply(dchisq(x,3),func)

# fit with default settings
fit <- unireg(x, y)

# plot of true function and fitted spline
plot(jitter(x), y, xlab="x (jittered)")
curve(dchisq(x,3), 0, 5, type="l", col="grey", add=TRUE, lwd=2)
points(fit, lwd=2, col="orange")
legend("topright", legend = c("true mean function", "difference
    penalized unimodal fit"), col=c("grey","orange"), lwd=c(2,2))
```



```
# estimated standard deviation
sqrt(fit$sigmasq) #[1] 0.02025133
```

5.2 Bayesian unimodal spline regression

The Bayesian approach to unimodal regression presented in Section 3.6 is implemented in the following R function `unibayes`.

`unibayes` *Fitting a Bayesian unimodal regression spline.*

Description

Fitting a Bayesian unimodal regression spline. Penalization is realized using a suitable covariance matrix for the prior of the B-spline coefficients and the shape constraint is induced by truncation of this prior. Posterior estimates are obtained using a Monte Carlo random sample from the posterior.

Usage

```
unibayes(x, y, sigmasq=NULL, N=100, a=min(x), b=max(x), g=10, k=3,
penalty=c("diff","sigEmax"), ordpen=2, vari=5, nCores=1)
```

Arguments

`x`	A numeric vector of n x-values. Contains at least $d \leq n$ distinct values.
`y`	A numeric vector of n observed y-values.
`sigmasq`	Positive numeric value of the residual variance. If `sigmasq=NULL` (default), each x-value has to be appear at least twice and `sigmasq` is estimated internally.
`N`	Number of Monte Carlo samples.
`a`	The left numeric boundary of the interval, on which the spline is defined. By default `a` is equal to the minimal x-value.
`b`	The numeric right boundary of the interval, on which the spline is defined. By default `b` is equal to the maximal x-value.

`g` A non-negative integer giving the number of inner knots of the spline (default: 10).

`k` A non-negative integer choosing the degree of the spline. By Default a cubic spline (`k=3`) is fitted.

`penalty` A character specifying, which penalty on the B-spline coefficients should be used. Possible choices are `"diff"` (default) for the differences penalty of order `ordpen` and `"sigEmax"` for the sigmoid E_{max} increase penalty (cf. Table 3.1).

`ordpen` Order of the difference penalty (integer ≥ 0, default 2). Only effective if `penalty="diff"`.

`vari` Variance parameter $\sigma_v^2 > 0$ (default 5) in the full-rank penalty matrix $\tilde{\boldsymbol{\Omega}}_\lambda$.

`nCores` The integer number of cores used for parallelization. If `nCores=1`, there is no parallelization (default).

Details

The function `unibayes` implements the Bayesian approach to unimodal spline regression as described in Section 3.6. The procedure starts with sorting $\boldsymbol{x}$ and $\boldsymbol{y}$ according to the values in $\boldsymbol{x}$. If `sigmasq = NULL`, an initial estimate of the variance is obtained as follows: Let u be the number of distinct x-values, $\tilde{x}_1, \ldots, \tilde{x}_u$, and let $I_t = \{i : x_i = \tilde{x}_t\} \subset \{1, \ldots, n\},\ t = 1, \ldots, u$, be the sets of the corresponding indices. We determine a variance estimate according to the formula,

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{t=1}^{u} \sum_{i \in I_t} (y_i - \bar{y}_t)^2 ,$$

where $\bar{y}_t$ is the mean in set I_t.

Afterwards, the y-values are transformed to lie in $[-1, 1]$ by $\frac{\boldsymbol{y} - shift}{scale}$, where $scale = 0.5 \cdot (\max(\boldsymbol{y}) - \min(\boldsymbol{y}))$ and $shift = \min(\boldsymbol{y}) + scale$. The variance parameter `sigmasq` is multiplied accordingly by $\frac{1}{scale^2}$.

For `penalty = "diff"` the penalty matrix is $\boldsymbol{\Omega} = \boldsymbol{D}_q' \boldsymbol{D}_q$, where $\boldsymbol{D}_q$ is the matrix of differences of order $q =$ `ordpen`, and $\boldsymbol{\beta}_0$ is just a vector of zeros. The variable `coinc` is set to `FALSE`. For the sigmoid E_{max} increase penalty `ordpen` is set to one to create $\boldsymbol{\Omega}$. The penalty vector $\boldsymbol{\beta}_0$ is determined by fitting a sigmoid E_{max} model to the data via the function `fitMod` from package `DoseFinding` (Bornkamp et al., 2016). The model predictions at the knot averages are calculated with function

`knotave` from package `SEL` (Bornkamp, 2010) and are used as entries of $\boldsymbol{\beta}_0$ according to Equation (3.3). The variable `coinc` is set to `TRUE`.
Additionally, a $d \times d$ matrix $\tilde{\boldsymbol{D}} = \frac{1}{\sigma_v^2}\boldsymbol{I}_d$ is created in both cases, from which the full-rank matrix $\tilde{\boldsymbol{\Omega}}_\lambda$ can be calculated as $\tilde{\boldsymbol{\Omega}}_\lambda = \tilde{\boldsymbol{D}} + \lambda\boldsymbol{\Omega}$. The variance σ_v^2 is chosen to be 5 per default, which can be thought of as uninformative since the β_i approximately also lie in $[-1, 1]$ (cf. control polygon characteristic in Section 3.3.2), which is the range of the transformed y-values.
The values of `a`, `b`, `g`, `k` and `coinc` are used to create the knot sequence with function `equiknots` (cf. Appendix D). The matrix $\boldsymbol{B}$ of B-spline basis functions of degree k on the interval $[a, b]$ with g inner knots, evaluated at the observed values $\boldsymbol{x}$, is determined with the function `splineDesign` from package `splines` (R Core Team, 2016).
The matrices and vectors $\boldsymbol{\beta}_0'\tilde{\boldsymbol{D}}\boldsymbol{\beta}_0$, $\boldsymbol{\beta}_0'\tilde{\boldsymbol{D}}$, $\boldsymbol{\beta}_0'\boldsymbol{\Omega}\boldsymbol{\beta}_0$, $\boldsymbol{\beta}_0'\boldsymbol{\Omega}$, $\frac{1}{\sigma^2}\boldsymbol{B}'\boldsymbol{B}$ and $\frac{1}{\sigma^2}\boldsymbol{y}'\boldsymbol{B}$ are calculated to avoid their repeated computation during the sampling procedure.
We use independent priors for the tuning parameter and the mode, that is $p(\lambda, m) = p(\lambda)p(m)$. The prior for λ is the Jeffreys prior $p(\lambda) \propto \frac{1}{\lambda}$, which is restricted to the interval $[e^{-3}, e^{10}] \approx [0.05, 22026.47]$ due to numerical problems that occur for very small or large values of λ and because an improper prior on the interval $(0, \infty)$ might result in an improper posterior. The prior distribution of the mode is simply the uniform distribution on all possible values, $p(m) = \frac{1}{d}\ \forall m \in \{-k, \ldots, g\}$.

The sampling scheme described in Section 3.6 is implemented as follows:

Approximating the posterior mode distribution and sampling from it
The integrals $\int w_2(\lambda|j)d\lambda$, $j \in \{-k, \ldots, g\}$ required for determining the posterior mode distribution, $p(m|\boldsymbol{y})$, are approximated by a Riemann sum using a grid of 200 tuning parameter values between $\lambda_1 = e^{-3}$ and $\lambda_{200} = e^{10}$ (exponentials of an equidistant sequence between -3 and 10). For each mode and each tuning parameter, we do the following:
The function `negloglikBayes` (cf. Appendix D), which computes the value of $w_2(\lambda|m)$ for given λ and m, is applied and the values `term12` and `term3` from the resulting object are retained in matrices $\boldsymbol{T}_{12}$ and $\boldsymbol{T}_3 \in \mathbb{R}^{d\times 200}$ (the mode varying in the rows and the tuning parameter varying in the columns). It turned out to produce better results, when `term3` is brought to a value range that is typical of negative log-likelihood values, namely $\mathbb{R}_0^+$. Then, the back-transformation to

likelihood-level with $\exp(-(\cdot))$ ends up in $[0, 1]$. Such values are achieved for `term3` by adding $t_{3,max} = \max(\boldsymbol{T}_3)$. Adding the same value to all negative log-likelihood values does not change the relationships between the likelihoods for different modes nor does it influence the slice sampling used below.
The repeated execution of `negloglikBayes` is done with the help of `sapply` across the grid of tuning parameters and across the different mode values with the help of `lapply` (if `nCores`=1) or `parLapply` (if `nCores` > 1).
The Riemann sum has to be calculated on the level of the likelihood function. So in the end, a matrix of likelihood values is calculated as $\boldsymbol{L} = \exp(-\boldsymbol{T}_{12} - \boldsymbol{T}_3 - t_{3,max} \cdot \mathbf{1}_{n\times n})$. For very small values, that is $\frac{l_{ij}}{\max(\boldsymbol{L})} < 10^{-4}$, we suppose that the corresponding values of λ are outside the support and set the likelihood to zero. The Riemann sum for mode m is then computed as

$$\sum_{k=1}^{199} (\lambda_{k+1} - \lambda_k) \frac{l_{m,k+1} + l_{m,k}}{2}.$$

The posterior mode distribution is obtained by dividing each Riemann sum by the sum of all Riemann sums.
A random sample of size N is drawn (with replacement) from the posterior distribution of the mode, which is a discrete distribution on $\{-k, \ldots, g\}$ and the absolute frequency N_m of each mode $m = -k, \ldots, g$ is retained.

Sampling from the marginal posterior of the tuning parameter

For each mode with $N_m > 0$, a sample of size N_m is drawn from the marginal posterior of λ given mode m and $\boldsymbol{y}$, or more precisely from $w_2(\lambda, m)$. This is done using an R implementation of the slice sampler introduced by Neal (2003), a Markov Chain Monte Carlo (MCMC) algorithm, which was kindly provided by Björn Bornkamp. The sampler works on the log-likelihood level and for computational efficiency updating is performed on $\log(\lambda)$-scale. In addition, samples are drawn from an approximation of the log-likelihood. In explicit, a smoothing spline is fitted with the R function `smooth.spline` (package `stats`, R Core Team, 2016) with argument `control.spar=list(low=-0.1)` to the rows of the matrix $\tilde{\boldsymbol{L}} = -\boldsymbol{T}_{12} - \boldsymbol{T}_3 - t_{3,max} \cdot \mathbf{1}_{n\times n}$ (that is, using the same grid of tuning parameters as before). We construct a function g_{log} with a tuning parameter value as its argument, which returns the predicted log-likelihood value according to the fitted spline at the argument, if the argument falls into the above calculated support of

the likelihood, and `-.Machine$double.xmax` otherwise.
To enable updating on $\log(\lambda)$-scale, another function $\tilde{g}_{log}$ has to be created. This is due to the following notion about the density of a transformed random variable (also called "change of variable"):

$$\text{Let } X \sim g \text{ and } Y = \phi(X). \text{ Then, } Y \sim \tilde{g} \text{ with } \tilde{g}(y) = g\left(\phi^{-1}(y)\right) \cdot \left|\frac{d\phi^{-1}}{dy}\right|$$

(cf. Härdle and Simar, 2007, p.106f).
Here, we have $Y = \phi(X) = \log(X)$ and $\tilde{g}(y) = g(\exp(y)) \cdot \exp(y)$. For the log-likelihood it holds that $\tilde{g}_{log}(y) = \log(\tilde{g}(y)) = \log(g(\exp(y)) + y = g_{log}(\exp(y)) + y$. Therefore, the function $\tilde{g}_{log}$ evaluates g_{log} at the exp of its own argument and returns the resulting value plus its argument.
The slice sampler draws from $\tilde{g}_{log}$ using a burn-in phase of 25 iterations and a thinning of two. The initial value of the MCMC chain is chosen as the tuning parameter value which maximizes $\tilde{g}_{log}$ (the maximizer is found with function `optim` in package `stats`). As opposed to the statement in Section 3.6, we do not generate only one random tuning parameter value for each posterior sample, but directly draw N_m times to reduce the computational burden. Determining $\tilde{g}_{log}$ and sampling from it for each mode is realized using `lapply` (if `nCores`=1) or `parLapply` (if `nCores` > 1).
The generated random values on log-level are back-transformed with $\exp(\cdot)$ to the original scale of the tuning parameter values and allocated to the respective mode samples.

Sampling from the marginal posterior of the B-spline coefficients
For each sampled mode and tuning parameter value, one sample from $p(\boldsymbol{\beta}|\lambda, m, \boldsymbol{y})$, that is, from the truncated multivariate normal distribution $\mathcal{N}_{\mathcal{S}_m}(\mathbf{e}_\lambda, \mathbf{E}_\lambda)$, is obtained using the R implementation `sampleTMVNIBF` (cf. Appendix D) of the inverse Bayes formulae sampler introduced by Yu and Tian (2011), which is described in Appendix C. The number of initial samples J is set to 50 and the starting point $\boldsymbol{r}^{(0)}$ for the EM-algorithm is chosen as follows: If $\boldsymbol{C}_m\mathbf{e}_\lambda \geq 0$, that is, if the mean is already a unimodal vector, $\boldsymbol{r}^{(0)} := \mathbf{e}_\lambda$. If $\mathbf{e}_\lambda$ is not unimodal, $\boldsymbol{r}^{(0)}$ equals its projection into the space $\mathcal{S}_m$ using formula 2.2 in Gunn and Dunson (2005) (see also Appendix E).

The sampled coefficient vectors are back-transformed to the original scale of the data via $scale \cdot \boldsymbol{\beta} + shift \cdot \mathbf{1}_d$ and a median coefficient vector $\hat{\boldsymbol{\beta}}_{med}$ (by components) is calculated afterwards.
A new estimate of σ^2 is calculated as the sample variance of the residuals $r_i = y_i - \boldsymbol{B}\hat{\boldsymbol{\beta}}_{med},\ i = 1, \ldots, n$.

Value

A list of

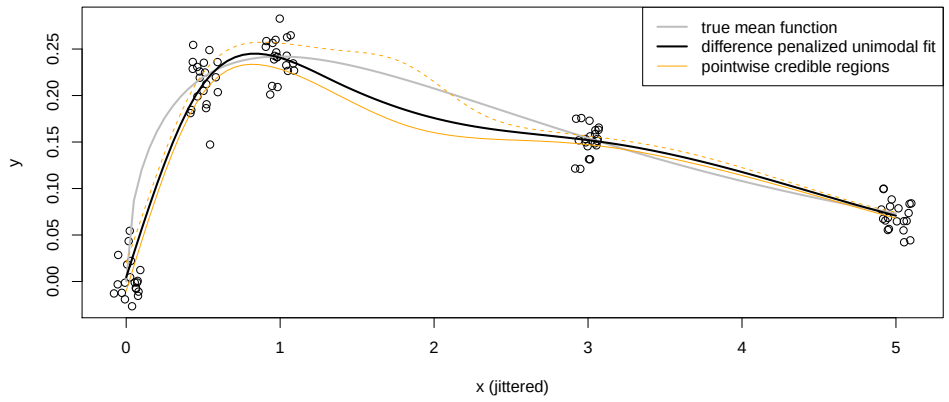
`fitted.values`	The fitted values $\boldsymbol{B}\hat{\boldsymbol{\beta}}_{med}$.
`betamed`	The median (by components) of the B-spline coefficient vector.
`betamean`	The mean (by components) of the B-spline coefficient vector.
`lambdamed`	The median tuning parameter.
`lambdamean`	The mean tuning parameter.
`modemed`	The median mode location.
`modemean`	The mean mode location.
`sigmasq`	The estimated residual variance.

Example

```
# generate some data
x <- sort(rep(c(0,0.2,1,3,5),20))
set.seed(41333)
func <- function(mu){rnorm(1,mu,0.02)}
y <- sapply(dchisq(x,3),func)

# fit with default settings
fit <- unibayes(x=x,y=y)
```

```
# plot of true function, fitted spline and credible intervals
plot(jitter(x), y, xlab="x (jittered)")
curve(dchisq(x,3), 0, 5, type="l", col="grey", add=TRUE, lwd=2)
z <- seq(min(x), max(x), length.out=100)
Bz <- splineDesign(fit$knotsequence, z, ord=4, outer.ok=TRUE)
matpoints(z,Bz%*%fit$coef, type="l", lwd=2)
matpoints(z,Bz%*%t(fit$betaquant), type="l", lwd=1, col="orange")
legend("topright", legend = c("true mean function", "difference
    penalized unimodal fit", "pointwise credible regions"),
    col=c("grey", "black", "orange"), lwd=c(2,2,1))
```




```
# estimated standard deviation
sqrt(fit$sigmasq) #[1] 0.01989864
```

5.3 Multimodal regression

This section describes the implementations of the multimodal regression models for homogeneous populations from Section 4.3. The implementations of all L_0-deconvolution models are based on the original implementation of the pointwise L_0 deconvolution model by de Rooi and Eilers (2011), which was kindly provided by the authors. The described extensions and combinations can be applied to data using the functions `parL0deco`, `uniL0deco`, `varL0deco` (cf. Sections 5.3.1, 5.3.2, 5.3.4).

The additive unimodal regression model is implemented in function `addUnireg`, see Section 5.3.3.
The unimodal penalized spline regressions used in the algorithms are performed using function `unireg` (cf. Section 5.1) with approximate REML to reduce the computational burden that arises from (repeatedly) estimating several unimodal regression functions.
A repeated step of the implementations is to transform numeric vectors so that their entries lie in $[0, 1]$ (with minimum value usually zero and maximum value one), which is described by the function $u : \mathbb{R}^n \to [0, 1]^n$ with $u(\boldsymbol{z}) = \begin{cases} \frac{\boldsymbol{z}-\min(\boldsymbol{z})}{\max(\boldsymbol{z})-\min(\boldsymbol{z})}, & \min(\boldsymbol{z}) < \max(\boldsymbol{z}) \\ \frac{\boldsymbol{z}}{\max(\boldsymbol{z})}, & \min(\boldsymbol{z}) = \max(\boldsymbol{z}). \end{cases}$

5.3.1 Deconvolution with a parametric peak shape

`parL0deco` *Fitting a parametric L_0-deconvolution model.*

Description

Fitting a deconvolution model with identical peak shapes using the L_0-penalty. The peak shape can be specified by a parametric function.

Usage

```
parL0deco(x, y, n_s, fpar, initvec, kappa=0.01, blind=FALSE,
zero.con=TRUE, nloop=50)
```

Arguments

`x`	Numeric vector of n x-observations.
`y`	Numeric vector of n y-observations.
`n_s`	Width of the peak shape measured as number of x-points (positive integer value).
`fpar`	Parametric function `fpar(x,parvec)` describing the peak shape. Its arguments are a vector `x` of n_s x-values and a vector `parvec` of length `npar` specifying the function parameters.
`initvec`	An initial parameter vector of length `npar`.
`kappa`	Positive tuning parameter value of the L_0-penalty, default 0.01.

`blind`	Logical indicating if blind deconvolution should be applied, i.e., if the peak shape should be optimized during estimation. If `FALSE` (default), L_0-deconvolution with fixed pointwise peak shape according to the parameters in `initvec` is applied.
`zero.con`	Logical indicating if the input pulses should be constrained to be positive (default: `TRUE`).
`nloop`	Number of iteration steps, 50 per default.

Details

This function fits a (blind) deconvolution model with identical peak shapes to data using the L_0-penalty as described in Section 4.3.2. The peak shape can be specified by a parametric function.

The implementation starts with transforming the n observed y-values to the interval $[0,1]^n$ with the help of function $u(\cdot)$. The parameter ϱ is initialized as 10^{-4} and the input pulses as $\boldsymbol{a}^{(0)} := \mathbf{0}_{n+n_s-1}$. If `zero.con=TRUE`, a variable η is set to exp(10), and 0 otherwise. The pointwise peak shape $\boldsymbol{s}$ is initialized as function `fpar` evaluated at $x_1, \ldots, x_{n_s}$ using the parameters in `initvec` and transformed with function $u(\cdot)$ to the interval $[0,1]$.

The vector $\boldsymbol{a}$ of input pulses can be estimated according to the L_0-penalty for a fixed convolution matrix $\boldsymbol{S}$ (holding shifted copies of $\boldsymbol{s}$ in its columns) using an iterative penalized least squares approach with re-weighted penalty matrix:

The function iterates `nloop` times between the (re-)calculation of

$$\boldsymbol{W}^{(t)} := \operatorname{diag}\left(\frac{1}{\left(a_1^{(t-1)}\right)^2 + \varrho}, \ldots, \frac{1}{\left(a_{n+n_s-1}^{(t-1)}\right)^2 + \varrho}\right),$$

$$\boldsymbol{V}^{(t)} := \operatorname{diag}\left(\mathbb{1}_{(-\infty,0)}\left(a_1^{(t-1)}\right), \ldots, \mathbb{1}_{(-\infty,0)}\left(a_{n+ns-1}^{(t-1)}\right)\right),$$

and the estimation of $\boldsymbol{a}^{(t)} := (\boldsymbol{S}'\boldsymbol{S} + \kappa\boldsymbol{W}^{(t)} + \eta\boldsymbol{V}^{(t)})^{-1}\boldsymbol{S}'\boldsymbol{y}$ to arrive at the L_0-penalized estimate of $\boldsymbol{a}$. If $\eta = \exp(10)$, there is a strong penalty on negative entries of $\boldsymbol{a}$ towards zero, otherwise there is no penalization. Additionally, a_j is set to zero, if $a_j < 0.0001$, which leads to faster convergence and avoids numerical problems, when a_j approaches zero.

If `blind=TRUE`, there is an additional step in every iteration that estimates an updated peak shape or, respectively, updates the parameters of function `fpar`. To

do this, a new R function is specified, which calculates the residual sum of squares, $||\boldsymbol{y} - \boldsymbol{A}^{(t-1)}(s(x_1|\boldsymbol{\beta}), \ldots, s(x_{n_s}|\boldsymbol{\beta}))'||^2$, where $s(\cdot|\boldsymbol{\beta})$ is the function `fpar` evaluated at a corresponding parameter vector and $\boldsymbol{A}^{(t-1)}$ holds shifted copies of $\boldsymbol{a}^{(t-1)}$ in its columns. This R function is optimized using `optim()` (package `stats`; R Core Team, 2016) and the updated peak shape $\boldsymbol{s}^{(t)}$ is obtained by evaluating `fpar` at the optimizing parameter vector and projecting it with function $u(\cdot)$ to $[0, 1]^{n_s}$. Restricting the peak shape to the same value range as the observations guarantees the identifiability of the model parameters.
Subsequent to the iteration, some back-transformation steps are performed. The estimated input pulses $\boldsymbol{a}$ are scaled by the factor $\max(\boldsymbol{y} - \min \boldsymbol{y})$ (original $\boldsymbol{y}$). An object `const` is created to which the minimal value of the original observations $\boldsymbol{y}$ is assigned. It can be interpreted as an intercept of the estimated model since $\boldsymbol{s} \in [0, 1]^{n_s}$ and thus $\min(\boldsymbol{Sa}) = 0$, if `zero.con=TRUE`. The fitted values $\hat{\boldsymbol{y}}$ can then be calculated as $\boldsymbol{Sa} +$ `const` on the original scale.

Value

A list of

`yhat`	The fitted values.
`a`	The estimated vector of input pulses.
`const`	Intercept.
`s`	The estimated peak shape (evaluated at the grid specified by the input) in $[0, 1]^{n_s}$.
`S`	The estimated convolution matrix with vector $\boldsymbol{s}$ in its columns.
`parvec`	The estimated parameter vector of the peak shape function. Caution is required when interpreting those estimated parameters, because $\boldsymbol{s}$ is estimated between 0 and 1, which might not be its "natural" range! Rescaling might be required for some (especially multiplicative) parameters.

Example

For the analysis of the FACT time series in Section 7.2.2 we use the following parametric function `wave` to describe the peak shapes and call function `parL0deco` with the following arguments (assuming that the measured voltages are stored in vector `y`). The result is displayed in Figure 7.3.

```
wave <- function(x,parvec){
    U0=parvec[1]
    tau1=parvec[2]
    tau2=parvec[3]
    return(U0*(x>=0)*(1-exp(-x/tau1))*exp(-x/tau2))
}

parL0deco(x=seq_along(y), y, n_s=150, fpar=wave,
    initvec=c(17.41,4.745,31.81), kappa=0.01, blind=TRUE,
    zero.con=TRUE, nloop=50)
```

5.3.2 Deconvolution with a unimodal peak shape

`uniL0deco` *Fitting a unimodal L_0-deconvolution model.*

Description

Fitting a deconvolution model with identical peak shapes using the L_0-penalty. The peak shape is described by a unimodal spline function.

Usage

```
uniL0deco(x, y, g, k=3, n_s, sigmasq, beta0, kappa=0.01, blind=FALSE,
zero.con=TRUE, nloop=30)
```

Arguments

`x`	Numeric vector of n x-observations.
`y`	Numeric vector of n y-observations.
`g`	A non-negative integer giving the number of inner knots of the spline (default: 10).
`k`	A non-negative integer choosing the degree of the spline. By default a cubic spline (`k=3`) is fitted.
`n_s`	Width of the peak shape measured as number of x-points (positive integer value).

`sigmasq`	A positive numeric value providing an estimate of the model variance.
`beta0`	A numeric vector of $d = g + k + 1$ B-spline coefficients against which the spline is penalized. For example, a parametric function evaluated at the knot averages as described in Section 3.3.2.
`kappa`	Positive tuning parameter value of the L_0-penalty, default 0.01.
`blind`	Logical indicating if blind deconvolution should be applied, i.e., if the peak shape should be optimized during estimation. If `FALSE` (default), L_0-deconvolution with fixed peak shape according to the B-spline coefficients in `beta0` is applied.
`zero.con`	Logical indicating if the input pulses should be constrained to be positive (default: `TRUE`).
`nloop`	Number of iteration steps, 50 per default.

Details

This function fits a (blind) deconvolution model with identical peak shapes to data using the L_0-penalty as described in Section 4.3.2. The peak shape is modelled by a unimodal spline function, which can be penalized against a parametric function fit as presented in Section 3.3.2.

The implementation starts with transforming the n observed y-values with function $u(\cdot)$ to the interval $[0, 1]^n$. The variance estimate `sigmasq` and penalty vector `beta0` are transformed accordingly. The parameter ϱ is initialized as 10^{-4} and the input pulses as $\boldsymbol{a}^{(0)} := \mathbf{0}_{n+n_s-1}$. If `zero.con=TRUE`, a variable η is set to $\exp(10)$, and 0 otherwise. A knot sequence with g inner knots and degree k as well as the corresponding B-spline basis matrix $\boldsymbol{B}$ evaluated at $x_1, \ldots, x_{n_s}$ are created. The peak shape is initialized as $\boldsymbol{s}^{(0)} := \boldsymbol{B}\boldsymbol{\beta}_0$ and transformed with function $u(\cdot)$ to the interval $[0, 1]$.

The vector $\boldsymbol{a}$ of input pulses can again be estimated according to the L_0-penalty for a fixed convolution matrix $\boldsymbol{S}$ (holding shifted copies of $\boldsymbol{s}$ in its columns) using an iterative penalized least squares approach with re-weighted penalty matrix:

The function iterates `nloop` times between the (re-)calculation of

$$\boldsymbol{W}^{(t)} := \operatorname{diag}\left(\frac{1}{\left(a_1^{(t-1)}\right)^2 + \varrho}, \ldots, \frac{1}{\left(a_{n+n_s-1}^{(t-1)}\right)^2 + \varrho}\right),$$

$$\boldsymbol{V}^{(t)} := \text{diag}\left(\mathbb{1}_{(-\infty,0)}\left(a_1^{(t-1)}\right), \ldots, \mathbb{1}_{(-\infty,0)}\left(a_{n+ns-1}^{(t-1)}\right)\right),$$

and the estimation of $\boldsymbol{a}^{(t)} := (\boldsymbol{S}'\boldsymbol{S} + \kappa\boldsymbol{W}^{(t)} + \eta\boldsymbol{V}^{(t)})^{-1}\boldsymbol{S}'\boldsymbol{y}$ to arrive at the L_0-penalized estimate of $\boldsymbol{a}$. If $\eta = \exp(10)$ there is a strong penalty on negative entries of $\boldsymbol{a}$ towards zero, otherwise there is no penalization. Additionally, a_j is set to zero, if $a_j < 0.0001$, which leads to faster convergence and avoids numerical problems, when a_j approaches zero.

If `blind=TRUE` there is an additional step in every iteration that estimates an updated peak shape by updating the B-spline coefficients. This is done with the help of a small modification of function `unireg`, which estimates a unimodal B-spline coefficient vector penalized against $\boldsymbol{\beta}_0$ (the penalty matrix is $\boldsymbol{\Omega} = \boldsymbol{I}_d$) using the product matrix $\boldsymbol{AB}$ as design matrix instead of only $\boldsymbol{B}$, where $\boldsymbol{A}$ holds shifted copies of $\boldsymbol{a}$ in its columns. The spline estimated in this way is evaluated at $x_1, \ldots, x_{n_s}$ and the resulting peak shape is projected to the interval $[0,1]^{n_s}$. Restricting the peak shape to this value range guarantees the identifiability of the model parameters.

Subsequent to the iterations, some back-transformation steps are performed. The input pulses $\boldsymbol{a}$ are scaled by the factor $\max(\boldsymbol{y} - \min(\boldsymbol{y}))$ (original $\boldsymbol{y}$). An object `const` is created to which the minimal value of the original observations $\boldsymbol{y}$ is assigned. It can be interpreted as an intercept of the estimated model since $\boldsymbol{s} \in [0,1]^n$ and thus $\min(\boldsymbol{Sa}) \geq 0$, if `zero.con=TRUE`. The fitted values $\hat{\boldsymbol{y}}$ can then be calculated as $\boldsymbol{Sa} + \texttt{const}$.

Value

A list of

`yhat`	The fitted values.
`a`	The estimated vector of input pulses.
`const`	Intercept.
`s`	The estimated peak shape (evaluated at the grid specified by the input) within $[0,1]^{n_s}$.
`S`	The estimated convolution matrix with vector $\boldsymbol{s}$ in its columns.
`parvec`	The estimated vector of B-spline coefficients.

5.3.3 Deconvolution with additive unimodal regression

`addUnireg` *Fitting an additive unimodal regression model.*

Description

Deconvolution with diverse peak shapes by fitting an additive unimodal regression, that is, an additive model with unimodal spline components.

Usage

```
addUnireg(x, y, g, k=3, sigmasq, Lvec)
```

Arguments

`x`	Numeric vector of n x-observations.
`y`	Numeric vector of n y-observations.
`g`	A non-negative integer giving the number of inner knots of the spline (default: 10).
`k`	A non-negative integer choosing the degree of the spline. By default a cubic spline (`k=3`) is fitted.
`sigmasq`	A positive numeric value providing an estimate of the model variance.
`Lvec`	Integer vector specifying the numbers of components with which to fit the model.

Details

This function fits an additive unimodal regression model as described in Section 4.3.3 to data. As pointed out before, this additive model can be estimated with the backfitting algorithm. This is done for all numbers of components specified in `Lvec`. For a model with L components the implementation carries out the following steps:

The procedure starts with estimating the intercept α using the mean of the supplied y-observations and initializing a matrix of fitted values $\boldsymbol{M}_f = (s_\ell(x_i))_{\ell=1,\ldots,L,i=1,\ldots,n}$ with zeros.

The following steps are repeated at most 20 times:
For all $\ell = 1, \ldots, L$ a unimodal regression is performed on the observations $(\boldsymbol{x}, \tilde{\boldsymbol{y}})$ using the function `unireg` with arguments `g`, `k`, `sigmasq`, `tuning=FALSE`, `abstol=NULL` and `ordpen=0` (cf. Section 5.1). The fitted values are afterwards centred around zero and the value $\frac{1}{n}\sum_{i=1}^{n} \hat{s}_\ell(x_i)$ used for centring is also subtracted from the estimated coefficient vector $\hat{\boldsymbol{\beta}}_\ell$. The fitted values (stored in the columns of matrix $\boldsymbol{M}_f$), coefficient vectors (stored in the columns of a matrix $\boldsymbol{M}_\beta$) and the effective dimension of the fit (cf. Section 4.3.5; stored in a vector v_{ed}) of all components are retained. The iteration is stopped early if the maximum relative change in the estimated coefficient vectors of all components is less than or equal to 5%, that is, if $\max\left(\frac{|\boldsymbol{M}_\beta^{(t)} - \boldsymbol{M}_\beta^{(t-1)}|}{\boldsymbol{M}_\beta^{(t-1)}}\right) \leq 0.05$, where the maximum is taken over all matrix entries.
Subsequent to the iterations, the values of the information criteria AIC and BIC are calculated for the estimated model (cf. Section 4.3.5). The matrices $\boldsymbol{M}_f$ and $\boldsymbol{M}_\beta$ are written into lists and the values of AIC, BIC and the number of iterations used are written into vectors.
The aforementioned steps are performed for all numbers of components L specified in `Lvec` and the model with lowest AIC value is determined. For this model, the vector of fitted values is calculated according to formula (4.5) as the sum of α and the fitted values in each component. The function returns the intercept α, the fitted values, the estimated coefficient vectors, AIC and BIC values and the numbers of performed backfitting iterations for all estimated models with component numbers specified by `Lvec`. Additionally, the knot sequence used for the unimodal regressions is returned, so that each fitted spline and therefore each model can be easily evaluated at desired predictor values.

Value

A list of

`alpha`	Estimate of intercept α.
`Lvec`	Vector specifying the numbers of components of all fitted models.
`L_aic`	Number of components resulting in smallest AIC value (out of all entries in `Lvec`).

`yhat_aic`	Fitted values of the model with `L_aic` components and smallest AIC.
`aics`	Vector of AIC values (one for each entry in `Lvec`).
`bics`	Vector of BIC values (one for each entry in `Lvec`).
`coeflist`	List of coefficient matrices $\boldsymbol{M}_\beta$, one for each L in `Lvec`. Each matrix $((g+k+1)\times L)$ contains estimated B-spline coefficients for the L spline components in its rows.
`knotseq`	The knot sequence used for spline fitting.
`n_it`	Vector informing about the number of backfitting iterations used for each model.

5.3.4 Deconvolution with diverse unimodal peak shapes

`varL0deco` *Fitting an L_0-deconvolution model with diverse peak shapes.*

Description

Deconvolution with diverse peak shapes using the L_0-penalty. Each peak shape is described by a unimodal regression spline.

Usage

```
varL0deco(x, y, sigmasq, g, k=3, kappa=0.002)
```

Arguments

`x`	Numeric vector of n x-observations.
`y`	Numeric vector of n y-observations.
`g`	A non-negative integer giving the number of inner knots of the spline (default: 10).
`k`	A non-negative integer choosing the degree of the spline. By Default a cubic spline (`k=3`) is fitted.
`sigmasq`	A positive numeric value providing an estimate of the model variance.
`kappa`	Positive tuning parameter value of the L_0-penalty, default 0.002.

Details

The L_0-deconvolution model with diverse unimodal peak shapes (cf. Section 4.3.4) can be fitted by iteratively estimating the peak shapes s_j with steps similar to the backfitting algorithm for the additive model and estimating the input pulses $\boldsymbol{a}$ using the L_0-penalty.

The implementation starts with transforming the n observed y-values with function $u(\cdot)$ to the interval $[0,1]^n$ and transforming the variance estimate `sigmasq` accordingly. The parameter ϱ is initialized as 10^{-5} and a variable η is created with value exp(10). Initial peak shapes are estimated by applying the function `unireg` with `tuning=FALSE`, `abstol=NULL`, `penalty="diag"` and `allfits=TRUE` (cf. Section 5.1). These spline fits, evaluated at the observation points $\boldsymbol{x}$ and projected to $[0,1]^n$, are used as columns of the convolution matrix $\boldsymbol{S}^{(0)} \in \mathbb{R}^{n\times d}$. All diagonal entries of the weight matrix $\boldsymbol{W}^{(0)}$ are set to κ. An initial estimate of the input pulses is given by $\boldsymbol{a}^{(0)} := (\boldsymbol{S}^{(0)\prime}\boldsymbol{S}^{(0)} + \boldsymbol{W})^{-1}\boldsymbol{S}^{(0)\prime}\boldsymbol{y}$. To get a more stable initial estimate $\boldsymbol{a}^{(0)}$ we re-calculate $\boldsymbol{W}^{(0)} := diag\left(\frac{1}{a_1^2+\varrho^2}, \ldots, \frac{1}{a_d^2+\varrho^2}\right)$ and $\boldsymbol{a}^{(0)} := (\boldsymbol{S}^{(0)\prime}\boldsymbol{S}^{(0)} + \boldsymbol{W}^{(0)})^{-1}\boldsymbol{S}^{(0)\prime}\boldsymbol{y}$. Afterwards, $a_j^{(0)}$ is set to zero, if $a_j^{(0)} < 0.0001$, $j = 1, \ldots, d$. This operation is replicated in later steps of the procedure, since it leads to faster convergence and avoids numerical problems, when the input pulses approach zero.

For $t = 1$ to 80 (outer loop), the function iteratively updates $\boldsymbol{S}^{(t)}$ from $\boldsymbol{a}^{(t-1)}$ and $\boldsymbol{a}^{(t)}$ from $\boldsymbol{S}^{(t)}$. The superscript (t) indicating the iteration is omitted for simplicity in the following unless it is essential.

To speed up the estimation process, the ℓ-th peak shape (ℓ-th column of $\boldsymbol{S}$) is only updated, if $\ell \in \mathcal{L} := \bigcup_{\{j:a_j>0\}}\{j-2, \ldots, j+2\} \cap \{1, \ldots, d\}$, that is, if there exists a truly positive input pulse in its direct neighbourhood.

In each step t the set $\mathcal{L}$ is determined. Let $\ell_{(1)}, \ldots, \ell_{(L)}$ be its L elements, sorted such that the corresponding input pulses are in descending order: $a_{\ell_{(1)}} \geq a_{\ell_{(2)}} \geq \ldots \geq a_{\ell_{(L)}}$.

For $\ell = \ell_{(1)}, \ldots, \ell_{(L)}$ (inner loop) we compute $\tilde{\boldsymbol{a}} := (a_1, \ldots, a_{\ell-1}, 0, a_{\ell+1}, \ldots, a_d)$ and $\tilde{\boldsymbol{y}} := \boldsymbol{y} - \boldsymbol{S}\tilde{\boldsymbol{a}}$ similar to the backfitting algorithm. Then we fit a spline s_ℓ with fixed mode ℓ to $\tilde{\boldsymbol{y}}$ using function `unireg` with argument `m=`ℓ. After each spline estimation the fitted values are transformed to the interval $[0,1]^n$ to ensure identifiability and interpretability (the input pulses describe the heights of the peaks). The fitted values are retained in a column of the matrix $\boldsymbol{M}_f$. The spline coefficients are transformed accordingly and stored in a column of matrix $\boldsymbol{M}_\beta$. The

fitted spline values are then used to update the ℓ-th column of the convolution matrix: $\boldsymbol{S}_{.\ell} := u(s_\ell(x_1), \ldots, s_\ell(x_n))$.
For each ℓ, that is, with the convolution matrix $\boldsymbol{S}$ updated in column ℓ, the following steps are repeated:

- $\boldsymbol{W} := diag\left(\frac{1}{a_1^2+\varrho^2}, \ldots, \frac{1}{a_d^2+\varrho^2}\right)$
- $\boldsymbol{V} := diag\left(\mathbb{1}_{(-\infty,0)}(a_1), \ldots, \mathbb{1}_{(-\infty,0)}(a_d)\right)$
- $\boldsymbol{a} := \left(\boldsymbol{S}'\boldsymbol{S} + \kappa\boldsymbol{W} + \eta\boldsymbol{V}\right)^{-1}\boldsymbol{S}'\boldsymbol{y}$.

This is done five times with fixed matrix $\boldsymbol{S}$ in order to make both penalties on $\boldsymbol{a}$ work, the L_0-penalty $\kappa\boldsymbol{W}$ as well as the positivity-penalty $\eta\boldsymbol{V}$. The latter one is induced by putting a strong penalty ($\eta = \exp(10)$) on negative entries of $\boldsymbol{a}$ towards zero.
The last operation in each iteration of the inner loop is setting a_j to zero, if $a_j < 0.001$, $j = 1, \ldots, d$.
In the outer loop, the alternating estimation of columns of the convolution matrix $\boldsymbol{S}$ and input pulses $\boldsymbol{a}$ is stopped early if the position and the height of the positive input pulses has stabilized, that is, if $\max(|\boldsymbol{a}^{(t)} - \boldsymbol{a}^{(t-1)}|) < 0.001$.
Subsequent to the iteration, some back-transformation steps are performed. The input pulses $\boldsymbol{a}$ are scaled by the factor $\max(\boldsymbol{y} - \min(\boldsymbol{y}))$ (original $\boldsymbol{y}$). An object `const` is created to which the minimal value of the original observations $\boldsymbol{y}$ is assigned. It can be interpreted as an intercept of the model since all estimated peak shapes lie in $[0, 1]^n$ and $\boldsymbol{a} \geq \boldsymbol{0}$, so that $\min(\boldsymbol{S}\boldsymbol{a}) \geq 0$. The fitted values $\hat{\boldsymbol{y}}$ can then be calculated as $\boldsymbol{S}\boldsymbol{a} + \texttt{const}$. The matrix $\boldsymbol{M}_\beta$ is reduced to $\tilde{L}$ columns corresponding to those input pulses that are truly positive in the end.

Value

A list of

`yhat`	The fitted values.
`a`	The estimated vector of input pulses.
`const`	Intercept.
`S`	The estimated convolution matrix.
`betamat`	A matrix ((`g` + `k` + 1) $\times\tilde{L}$) that contains estimated B-spline coefficients for the $\tilde{L}$ splines corresponding to $a_j \neq 0$ in its columns.
`knotseq`	The knot sequence used for spline fitting.

Example

For the analysis of the IMS spectra A and B in Section 7.2.3 we use the following call to function `varL0deco` (assuming that the respective measured inverse reduced mobilities and voltages are stored in vectors `x` and `y`). The results are displayed in Figures 7.5 and 7.6 .

```
sigmasq <- var(y[35:700])

x <- x[-(1:700)]
y <- y[-(1:700)]

varL0deco(x, y, g=200, k=3, sigmasq=sigmasq, kappa=0.002)
```

6 Simulation study for unimodal regression

In this chapter the proposed unimodal regression methods from Section 3 are evaluated and compared to existing methodologies. A simulation study is performed, which is motivated by the data situation typically observed in pharmaceutical dose-response clinical trials, where increasing levels of a pharmaceutical compound are administered in parallel to a large number of patients to investigate the dose-response relationship. First, the data generation process is described. Then, details follow on the settings, the fitting process and the way that the results are evaluated. All calculations are carried out using R, version 3.2.5 (R Core Team, 2016).
The article Köllmann et al. (2014) is based on parts of the material in this chapter.

6.1 Data generation process

The evaluation of the proposed methods primarily follows Bornkamp et al. (2007), where the simulation scenarios were selected so that they are realistic for Phase II trials (for example, in terms of the dose-response shape, the number of doses and the signal to noise ratio), but also tries to generalize to other applications.
The data generation process always yields two data vectors: the vector $\boldsymbol{x} = (x_1, \ldots, x_n)'$ of observed predictor values (for example, doses) and the vector $\boldsymbol{y} = (y_1, \ldots, y_n)'$ of observed response values. The predictor values range from 0 to 8 and it is assumed that the same predictor value is observed several times, as is typical in dose-response data. Thus the vector $\boldsymbol{x}$ is restricted to take values from one of the two equally spaced sequences a) $(0, 2, 4, 6, 8)$ or b) $(0, 1, 2, 3, 4, 5, 6, 7, 8)$. Each value is observed with the same frequency, namely $\frac{n}{5}$ or $\frac{n}{9}$ times.
Given the predictor values, the responses are generated according to nine function profiles, which are shown in Figure 6.1.

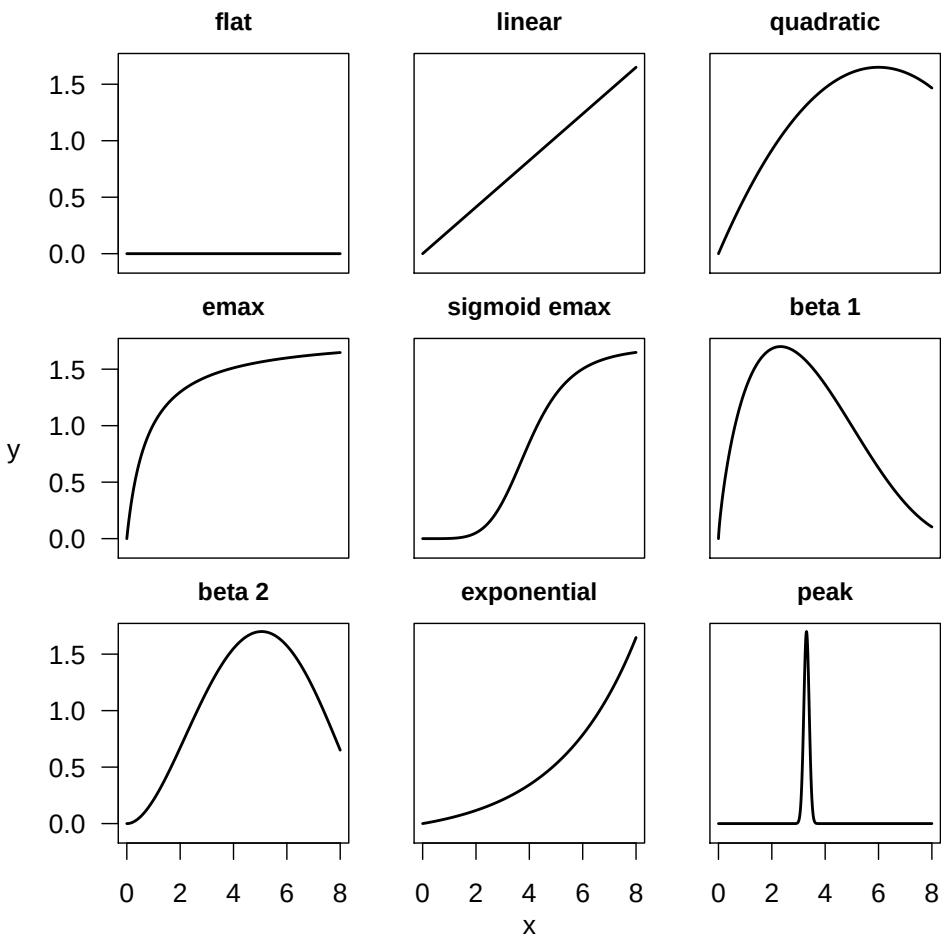


Figure 6.1: **The nine function profiles used in the simulation study.**

In comparison with Bornkamp et al. (2007) the logistic profile is left out as it is very similar to sigmoid E_{max}, and four additional profiles are included, three of which have a more pronounced unimodal shape (the two beta profiles and the peak profile) and an exponential function, which is monotone, but convex. The actual values y_i are simulated by $y_i = f(x_i) + \epsilon_i$, where $\epsilon_i \underset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$ and the exact functional forms f are given in Table 6.1.

The overall sample size and the standard deviation of the errors are set to $n = 250$ and $\sigma = \sqrt{4.5}$, taken from Bornkamp et al. (2007). In the case of repeated measurements at sequence (b), the samples size differs slightly: $n = 252$.

Altogether there are 18 combinations of those settings, see also Table A.1 in the Appendix. For each of those 18 scenarios 500 data sets are generated.

Table 6.1: **Functional forms of the nine function profiles used in the simulation study.** $\mathrm{Be}(\gamma_1, \gamma_2) = \frac{(\gamma_1+\gamma_2)^{\gamma_1+\gamma_2}}{\gamma_1^{\gamma_1}\gamma_2^{\gamma_2}}$ for $\gamma_1, \gamma_2 > 0$, $\varphi(x|\mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$ for $x, \mu \in \mathbb{R}, \sigma > 0$.

Profile name	Functional form
Flat	$f(x) \equiv 0$
Linear	$f(x) = \frac{1.65}{8}\, x \cdot \mathbb{1}_{[0,8]}(x)$
Umbrella	$f(x) = 1.65\left(\frac{1}{3}x - \frac{1}{36}x^2\right) \cdot \mathbb{1}_{[0,8]}(x)$
E_{max}	$f(x) = \frac{1.81x}{0.79+x} \cdot \mathbb{1}_{[0,8]}(x)$
sigmoid E_{max}	$f(x) = \frac{1.7x^5}{4^5+x^5} \cdot \mathbb{1}_{[0,8]}(x)$
Beta 1	$f(x) = 1.7 \cdot \mathrm{Be}(0.8, 2.5)\left(\frac{x}{9.6}\right)^{0.8}\left(1-\frac{x}{9.6}\right)^{2.5} \cdot \mathbb{1}_{[0,8]}(x)$
Beta 2	$f(x) = 1.7 \cdot \mathrm{Be}(2, 1.8)\left(\frac{x}{9.6}\right)^{2}\left(1-\frac{x}{9.6}\right)^{1.8} \cdot \mathbb{1}_{[0,8]}(x)$
Exponential	$f(x) = 0.123 \cdot (\exp(\frac{x}{3}) - 1) \cdot \mathbb{1}_{[0,8]}(x)$
Peak	$f(x) = 1.7 \cdot \frac{\varphi(x\|3.3, 0.25)}{\varphi(0\|0, 0.25)} \cdot \mathbb{1}_{[0,8]}(x)$,

6.2 Compared methods and fitting process

Several parametric and non-parametric methods are compared using the above specified simulation scheme. In comparison to the article Köllmann et al. (2014), the 24 approaches therein are appended here by the log-concave spline regression by Eilers (2005).

The methods can be divided into one non-parametric method, three methods based on parametric models and 21 semi-parametric methods, 17 of which are based on B-splines and 4 on Bernstein polynomials.

Regarding the proposed spline methods we compare methods with and without shape constraint and with and without penalization. We use cubic splines and the knot sequences are chosen as follows: By default, there are $g = 10$ equidistant inner knots in $[0, 8]$, which is already large compared to the number of distinct x-values (5 or 9, respectively). There are four coincident knots at each boundary, which yields a knot sequence of length $g + 2k + 2 = 18$ and a parameter vector of length $d = g + k + 1 = 14$. Only for the difference penalized methods the knots are equidistant also beyond the boundaries. In the case of the unpenalized splines (models "un" and "cn") the number of estimable

parameters (including σ) is bounded by the number u of distinct x-values in the particular data set, which is 5 or 9, respectively. Thus, the knot sequence is shorter here, the number of inner knots equals $g = u - 5$, which is 0 or 4, respectively.

The 11 frequentist spline regression methods are fitted using the function `unireg` in R package `uniReg` (Köllmann, 2016, version 1.1). We fit two unpenalized spline models, one unconstrained (model "un") and one constrained (model "cn"). The same is done for second and third order difference (models "ud2", "cd2", "ud3", "cd3") and sigmoid E_{max} penalized splines (models "us" and "cs"). When the unimodality constraint is active, we additionally vary the way of choosing the tuning parameter. The models "cd2", "cd3" and "cs" follow the REML approach, while the models "cda2", "cda3" and "csa" choose λ by approximate REML (argument `tuning=FALSE`), which means that the unimodality constraint is not accounted for during tuning parameter optimization, resulting in the same tuning parameter as for the unconstrained models.

Another spline method is the log-concave spline regression by Eilers (2005) (model "logCon"), which uses an iteratively penalized generalized linear model approach to impose concavity on the exp of the response. The shape penalty is combined with a third order differences penalty for smoothness. We use the same knot sequence as for the other difference penalized spline methods. The tuning parameter of the shape penalty is set to 10^6 to ensure a log-concave result and the tuning parameter of the difference penalty is optimized with cross-validation carried out via the function `cvTuning` from package `cvTools` (Alfons, 2012).

We also fit four models based on Bernstein polynomials (as used, for example, in Wang and Ghosh (2012)) by applying our penalized procedure with $g = 0$ inner knots and degree $k = 13$ (which results in the same number of parameters as for the other spline methods). This allows to evaluate the impact of a different selection of the spline basis and to provide a direct comparison of cubic splines and polynomials. Again, an unconstrained and a constrained model is fitted, each combined with the sigmoid E_{max} penalty (models "uBPs" and "cBPs") and, since the difference penalty does not make sense for the coincident boundary knots of Bernstein polynomials, with the ridge penalty $\sum_{j=-k}^{g} \beta_j^2$ (models "uBPr" and "cBPr").

Altogether there are 16 non-Bayesian procedures, 12 based on B-splines and 4 based on Bernstein polynomials. Moreover, we consider three Bayesian methods applying the proposed methodology with second and third order difference and sigmoid E_{max} increase penalty (methods "cdb2", "cdb3" and "csb") to enable comparison between maximization and averaging approaches. A fourth Bayesian method, "trafo", uses the second

order difference penalty and generates samples of the tuning parameter and the spline coefficients from the unconstrained spline model. The sampled coefficient vectors are subsequently mapped onto the space of unimodal vectors in $\mathbb{R}^d$ using the transformation procedure described in Gunn and Dunson (2005) (see also Appendix E). The unconstrained samples (before transformation) are also used as the fifth Bayesian method (model "udb") in the evaluation as well. For all Bayesian methods the resulting Monte Carlo sample from the joint posterior is of size 1000 and the parameters are estimated by their posterior means.

In addition to those 21 approaches, we include the following four models that do not make use of splines:

The model averaging method "modAve" takes a weighted average of the dose-response models `"linear"`, `"quadratic"`, `"emax"`, `"sigEmax"`, `"betaMod"` and `"exponential"`, fitted with the R function `fitMod` in R package `DoseFinding` (Bornkamp et al., 2016). The weights for averaging are the models' relative likelihood factors as described in Buckland et al. (1997). The functional forms of those models are as follows:

1. linear: $f(x) = E_0 + \gamma x$,
2. quadratic: $f(x) = E_0 + \gamma_1 x + \gamma_2 x^2$,
3. E_{max}: $f(x) = E_0 + E_1 \frac{x}{ED_{50} + x}$,
4. sigmoid E_{max}: $f(x) = E_0 + E_1 \frac{x^h}{ED_{50}^h + x^h}$,
5. beta: $f(x) = E_0 + E_{max} \cdot \text{Be}(\gamma_1, \gamma_2) \left(\frac{x}{scal}\right)^{\gamma_1} \left(1 - \frac{x}{scal}\right)^{\gamma_2}$,
6. exponential: $f(x) = E_0 + E_1 \cdot \left(\exp\left(\frac{x}{\gamma}\right) - 1\right)$,

where $\text{Be}(\gamma_1, \gamma_2) = \frac{(\gamma_1 + \gamma_2)^{\gamma_1 + \gamma_2}}{\gamma_1^{\gamma_1} \gamma_2^{\gamma_2}}$ for $\gamma_1, \gamma_2 > 0$ and $scal = 9.6$ is a fixed dose scaling parameter.

The models "sigE" and "beta" are the stand-alone parametric sigmoid E_{max} and the beta model and are also fitted using function `fitMod`.

The non-parametric method by Frisén (1986) (model "frisen") is a unimodal transformation of the response means at each dose, connected with straight lines. It is fitted by calculating the means of y-values with same x-value and applying the R function `ufit()` in R package `Iso` (Turner, 2015) to this sequence of means.

A short overview of all 25 methods including their abbreviations is given in Table A.2. All models are fitted to the 500 generated data sets of each scenario.

6.3 Evaluation methods

To compare the applied methods, four measures of mean relative loss (*MRL*), similar to the ones in Morell et al. (2013), are computed. Here, the loss of one method in a certain simulation scenario is related to the loss of the best method in that scenario and a mean value of the relative loss is obtained by averaging over a specified subset of scenarios. The closer the *MRL* is to zero, the less we lose on average by applying this method instead of the respective best methods. The losses are based on certain performance metrics.

The first metric is the average squared error (ASE) in function prediction, that is, the mean of the squared differences between the predicted and the true function values at specified points $\boldsymbol{z} = (z_1, \ldots, z_\eta)$ on the x-axis: $\text{ASE}(\hat{f}_{\rho,\nu}; \boldsymbol{z}) = \frac{1}{\eta} \sum_{i=1}^{\eta} (\hat{f}_{\rho,\nu}(z_i) - f_\rho(z_i))^2$, where f_ρ is the function of scenario $\rho \in \{1, \ldots, 18\}$ and $\hat{f}_{\rho,\nu}$ its estimate using method $\nu \in \{1, \ldots, 25\}$.

We obtain one value $\text{ASE}_\kappa(\hat{f}_{\rho,\nu}; \boldsymbol{z})$ of this measure for each data set $\kappa = 1, \ldots, 500$. To summarise the performance of method ν over all data sets of scenario ρ, we calculate the mean ASE-value and define a first measure of loss as follows:

$$\mathcal{L}_1(\rho, \nu) = \frac{1}{500} \sum_{\kappa=1}^{500} \text{ASE}_\kappa(\hat{f}_{\rho,\nu}; \boldsymbol{z} = (0, 0.01, \ldots, 8)), \ \nu = 1, \ldots, 25, \rho = 1, \ldots, 18.$$

Other aspects for comparison of the methods are the modal value $\max\limits_{x \in [0,8]} \hat{f}_{\rho,\nu}(x)$ and the location $\text{mod}(\hat{f}_{\rho,\nu}) = \arg\max\limits_{x \in [0,8]} \hat{f}_{\rho,\nu}(x)$ of the mode, which are interesting, for example, in dose-response analysis as they describe the maximum effect of a drug and the dose level which yields it.

Both values are determined with the help of the function `optimize()` in R package `stats` (R Core Team, 2016) and for both characteristics we can define a mean performance over all data sets of one scenario as above. In the case of the modal value we execute the above calculation steps of the mean ASE with $\boldsymbol{z} = (z_1) = \text{mod}(\hat{f}_{\rho,\nu})$ and define the loss

$$\mathcal{L}_2(\rho, \nu) = \frac{1}{500} \sum_{\kappa=1}^{500} \text{ASE}_\kappa(\hat{f}_{\rho,\nu}; \text{mod}(\hat{f}_{\rho,\nu})) \ \nu = 1, \ldots, 25, \rho = 1, \ldots, 18.$$

In the case of the mode location we also calculate the mean of a squared error as a measure of loss, namely

$$\mathcal{L}_3(\rho,\nu) = \frac{1}{500}\sum_{\kappa=1}^{500}(\mathrm{mod}(\hat{f}_{\rho,\nu}) - \mathrm{mod}(f_\rho))^2,\ \nu = 1,\dots,25, \rho = 3,\dots,18.$$

For scenarios $\rho = 1, 2$ this loss is not defined, because a true mode location $\mathrm{mod}(f_\rho)$ does not exist for the flat function profile.

For dose-response applications, another important characteristic is the minimum effective dose (MED), that is, the smallest (observed) predictor value that yields a response of at least $f(0) + \delta$ for a given $\delta \geq 0$. Here $\delta = 1.3$ as in Bornkamp et al. (2007) is used. The loss $\mathcal{L}_4(\rho,\nu)$ in this context is chosen to be the percentage of data sets of scenario ρ, in which the MED was incorrectly estimated with method ν (which is already a summary of all data sets of one scenario). In scenarios, where an MED exists, incorrect estimation means that it is not detected or that the estimate has a wrong value, while in scenarios, where no MED exists, it means that an MED is wrongly detected.

Since comparing the 25 methods over 18 data scenarios is still quite unmanageable, we follow Morell et al. (2013) by defining a relative loss (RL) of method ν in scenario ρ:

$$\mathrm{RL}_\ell(\rho,\nu) = \frac{\mathcal{L}_\ell(\rho,\nu) - \mathcal{L}_\ell(\rho,*)}{\psi_\ell(\rho)},\ \ell = 1,\dots,4,\ \rho = 1,\dots,18,\ \nu = 1,\dots,25,$$

where $\mathcal{L}_\ell(\rho,*) = \min_{\nu=1,\dots,25}\mathcal{L}_\ell(\rho,\nu)$ is the minimal loss value achieved in scenario ρ by one of the compared methods. The standardization factor $\psi_\ell(\rho)$ equals $\mathcal{L}_\ell(\rho,*)$ for $\ell = 1, 2$. Regarding the losses $\mathcal{L}_3$ and $\mathcal{L}_4$ we face the problem that there are scenarios for which the best method has zero loss. To avoid the problem of division by zero the standardization factor is chosen to be $\psi_3(\rho) = \max\{\mathrm{mod}(f_\rho), 8-\mathrm{mod}(f_\rho)\}$ for $\ell = 3$, which is the maximal possible error since $\mathrm{mod}(\hat{f}_{\rho,\nu})$ and $\mathrm{mod}(f_\rho)$ both lie in $[0, 8]$. The standardization factor for $\ell = 4$ is chosen as $\psi_4(\rho) = 1$, which is also the maximal possible error.

The closer $\mathrm{RL}_\ell(\rho,\nu)$ is to zero, the less we lose when applying method ν instead of the best method in scenario ρ.

Finally we obtain an even more aggregated performance measure for all four criteria when we take the mean of the relative loss of method ν over all scenarios or a specified subset $\mathcal{R}$ of scenarios:

$$\mathrm{MRL}_\ell(\nu) = \frac{1}{|\mathcal{R}|}\sum_{\rho\in\mathcal{R}}\mathrm{RL}_\ell(\rho,\nu), \quad \ell = 1,\dots,4,\ \nu = 1,\dots,25.$$

In the next subsection we will show the mean relative loss for the scenario subsets

$$\mathcal{R}_1 := \{\text{all scenarios with 5 doses}\} = \{1, 3, \ldots, 17\} \text{ and}$$
$$\mathcal{R}_2 := \{\text{all scenarios with 9 doses}\} = \{2, 4, \ldots, 18\}.$$

Notice again that the flat scenarios ($\rho = 1, 2$) have to be left out in the case of loss $\mathcal{L}_3$.

6.4 Results

The *MRL* values of the 25 methods are plotted in Figure 6.2. The subfigures in each row correspond to the scenario subsets with 5 doses ($\mathcal{R}_1$) and with 9 doses ($\mathcal{R}_2$).
Since the results for the mean relative loss in estimation of the minimum effective dose (MRL_4) are quite different from the others, we will start with discussing only the first three measures of loss. For these, always one of the constrained spline methods yields the smallest value. More explicitly, the constrained Bayesian splines with second or third order penalty perform best.
Obviously the inclusion of the constraint mostly reduces the loss measures for unpenalized and penalized splines as well as for Bayesian splines. Only in case of the mean relative loss in estimation of the mode location (MRL_3) the order is reversed for unpenalized and third order differences penalized frequentist splines.
The constrained Bayesian approach mostly yields smaller values than the frequentist ones, especially with regard to mean relative loss in estimation of the mode location and for the difference penalized splines with regard to mean relative loss in function estimation (MRL_1).
The approximate REML approach is most of the time better than the corresponding unconstrained spline and worse than the exact REML constrained spline or the three of them perform similarly. An exception are the spline methods with third order difference penalty in case of MRL_3.
The Bayesian unimodal transformation method (Gunn and Dunson, 2005) has higher loss MRL_3 compared to our proposed Bayesian method and comparable loss MRL_1 and MRL_2 (in case of the latter measure of loss even smaller).
The log-concave restricted splines with third order difference penalty perform similar to the other difference penalized splines, but with regard to MRL_1 and MRL_2, the unimodality constrained splines seem more suitable.

For the mean relative loss in estimation of the MED (MRL_4) the results are very different. Here, the beta model performs best (followed by the unpenalized constrained spline) and among the penalized splines the performances of the unconstrained, the constrained (exact and approximate REML) and the Bayesian versions are in reverse order. The Bayesian transformation approach yields smaller values than the proposed Bayesian method. But regarding this fourth measure of loss, we should remark that in some scenarios even the *minimal* relative frequency of incorrect detection (minimal loss) is very high, once as high as 0.79. The methods presumably *all* produce a smooth overall fit that is less beneficial for estimation of the MED.

The best of the non-spline-methods for the performance measures MRL_1 and MRL_2 is the model averaging technique and for MRL_3 and MRL_4 the beta model.

Increasing the number of observed doses does not seem to make much difference for the performance of the methods, except that the loss in estimation of the modal value slightly grows for all models and MRL_4 declines for the difference penalized models.

While the sigmoid E_{max} model itself is always one of the worst methods, the sigmoid E_{max} penalized splines have smaller MRL, the latter ones performing very similar among each other, especially regarding MRL_1 and MRL_2. This indicates that the penalty against a parametric function indeed safeguards against model mis-specification. Regarding the Bernstein polynomials the sigmoid E_{max} penalty even performs mostly better than the ridge penalty and comparably to the sigmoid E_{max} penalized B-spline methods, the loss in estimation of the modal value is even slightly smaller.

Altogether, the simulation study demonstrates a good performance of the proposed unimodal spline regression approaches compared to several competitors, in particular in terms of estimating the dose-response function. Especially for the estimation of the MED, other approaches are more favourable.

Additional insight in the performance of unimodal spline regression in the dose-response context can be found in the diploma thesis of Jan Rekowski (Rekowski, 2013), where the frequentist unimodal spline regression with second order difference penalty is compared to three other methods in terms of proof of concept (probability of establishing the dose-response), clinical relevance, MED estimation and average prediction error. This simulation study also examines different dose allocation strategies.

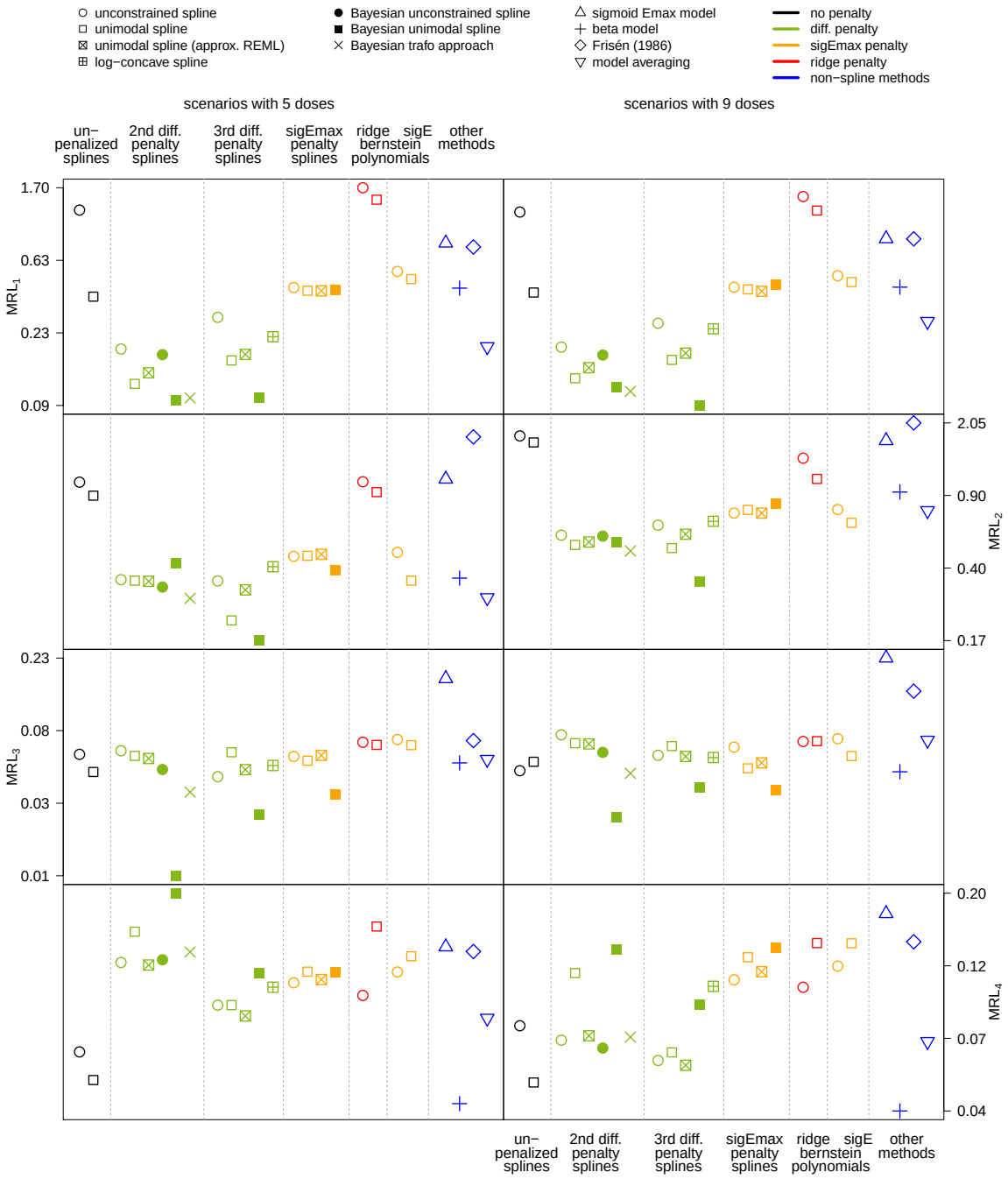


Figure 6.2: **Simulation results.**
Mean relative loss of the 25 methods in estimation of the function values at the grid $(0, 0.01, 0.02, \ldots, 7.99, 8)$ (MRL_1), of the modal value (MRL_2), of the mode location (MRL_3) and of the minimum effective dose (MRL_4). The rows correspond to the four performance measures and the columns to the two subsets of the scenarios, $\mathcal{R}_1$ and $\mathcal{R}_2$, over which the mean is taken. The values are spaced on $\log_{10}$-scale.

7 Applications

In this chapter we describe the application of unimodal and multimodal regression models to different real data sets. First, the proposed unimodal regression methods are applied to the growth hormone data set introduced in Section 2.1. Afterwards, the methods for multimodal regression from Section 4 are applied to data sets from three application areas: marine biology, breath gas analysis and astroparticle physics (see Sections 2.2 to 2.3). Section 7.3 gives an overview of research work by others, who applied the proposed methodologies for their purposes.

7.1 Unimodality: growth hormone dose-response analysis

The first application example analyses the four response variables ADG, Age, G/F and ADF in dependence of the dose of the growth hormone somatotropin (cf. Section 2.1), that is, we look at four univariate regression tasks:

$$y_{ij} = f_j(x_i) + \epsilon_{ij},\ i = 1, \ldots, 5,$$

where $j = 1, \ldots, 4$, indicates the j-th response variable and i indicates the i-th dose level of $x_1 = 0, \ldots, x_5 = 9$. The observed means, y_{ij}, are given in Table 2.1 and ϵ_{ij} are observations of $\mathcal{E}_{ij} \sim \mathcal{N}(0, \text{se}_{ij}^2)$, where se_{ij} are the standard errors at the respective doses (also given in Table 2.1).
We compare four different regression approaches:

1. the frequentist unimodal regression approach with difference penalty (cf. Section 3.5.1), where each f_j is a unimodal spline function,

2. the Bayesian unimodal regression approach with difference penalty (cf. Section 3.6), where each f_j is a unimodal spline function,

3. the frequentist unimodal spline regression with penalization against an exponential function fit (cf. Section 3.3.2), where each f_j is a unimodal spline function,

4. an exponential regression, which was found in McLaren et al. (1990) to fit the data best in terms of the coefficient of determination, where each f_j has the form $f_j(x) = u + v \cdot (1 - e^{wx})$ with parameters u, v, w.

McLaren et al. (1990) estimated the following exponential relationships in dependence of the dosage x for the four different response variables:

$$\begin{aligned} ADG(x) &= 0.7509 + 0.1523\left(1 - e^{-0.6204x}\right), \\ Age(x) &= 183.2 - 12.27\left(1 - e^{-0.5x}\right), \\ G/F(x) &= 0.2655 + 0.09982\left(1 - e^{-0.5036x}\right), \\ ADF(x) &= 2.932 - 0.7913\left(1 - e^{-0.2097x}\right). \end{aligned}$$

Those functions are evaluated at the knot averages to calculate the vector $\boldsymbol{\beta}_0$ for the penalty in approach 3 (cf. Equation (3.3)).
All calculations are carried out using R, version 3.2.5 (R Core Team, 2016). For approach 1 and 3 we use function `unireg` from R package `uniReg` in version 1.1, where for the argument `sigmasq` the standard errors from Table 2.1 are used and kept fix (argument `abstol=NULL`). For approach 2 we use the function `unibayes` (cf. Section 5.2) and `N=1000` samples are drawn from the posterior and averaged. In addition, we retain the pointwise 2.5%- and 97.5%-quantiles of the fitted functions to determine a 95% credible region. For all three fits, we use $g = 10$ equidistant inner knots on the interval $[0, 9]$. In the case of Age and ADF an inverse unimodal regression is applied.
For all four response variables, the three fitted unimodal regressions and the exponential fit are plotted in Figure 7.1 and the weighted residual sums of squares (wRSS) are given in Table 7.1.
From the plotted functions we can see that both fits with difference penalty do not differ very much (except for ADG), yet there are clear differences in their weighted residual sums of squares. The Bayesian method yields higher wRSS values and seems to oversmooth the data, but the credible regions lie close around the fitted curves and stay within the normal confidence intervals defined by the standard errors at the observed doses. For the variables ADG, Age and ADF the unimodal regression with exponential penalty and the exponential fit are nearly identical, but in the case of G/F they differ more clearly. For this variable the exponential model seems to be a mis-specification,

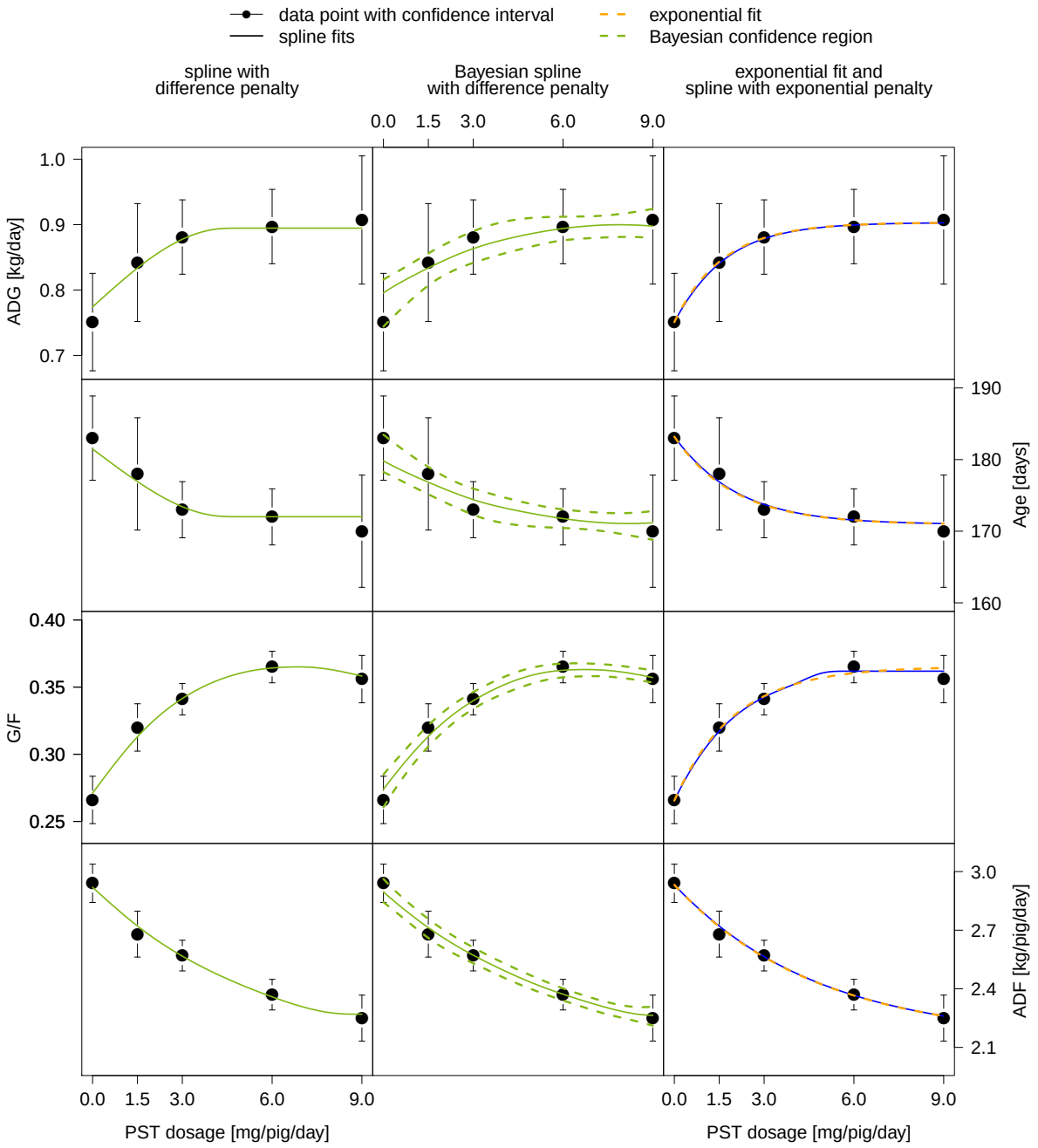


Figure 7.1: **Dose-response data with fitted spline functions.**
Scatterplots of PST dosage vs. means of the response variables ADG, Age, G/F and ADF. The bars indicate 95% confidence intervals of a normal distribution with the input means and standard errors. The rows correspond to the response variables and the columns to the fitted unimodal functions: frequentist and Bayesian spline with difference penalty and frequentist spline with exponential penalty. For the Bayesian method the 95% credible regions of the fitted functions are also shown.

Table 7.1: **Goodness of fit of the regression models.** Weighted residual sums of squares of the exponential fit by McLaren et al. (1990) and the three fitted unimodal regressions for all four response variables.

	Difference penalty	Bayes (Difference)	Exponential fit	Exponential penalty
ADG	0.486	1.844	0.018	0.023
Age	0.629	1.802	0.341	0.340
G/F	1.003	1.480	1.592	0.864
ADF	0.836	1.088	0.514	0.554

because all unimodal regressions, also the Bayesian one, have smaller wRSS. McLaren et al. (1990) used the exponential fits to determine optimal dosages. For the G/F variable such subsequent analyses would clearly yield different, presumably more appropriate results using the unimodal fit.
Since we do not know the true underlying relationship we can only asses the different results in terms of wRSS, from which we can derive that the frequentist unimodal spline with penalty against the preferred parametric function is more useful than just fitting the parametric model: the difference in wRSS is small, when the parametric model already yields a good fit, but the benefit can be substantial as in the case of the G/F variable.

7.2 Multimodality

In the following three subsections the methods presented in Chapter 4 are applied to multimodal data sets from marine biology, astroparticle physics and breath gas analysis (cf. Köllmann et al., 2016).
All analyses are performed using R (version 3.2.5; R Core Team, 2016). The unimodal penalized spline regressions are fitted using function `unireg` in R package `uniReg` (version 1.1; Köllmann, 2016). We use approximate REML to reduce the computational burden that arises from (repeatedly) estimating several unimodal regression functions.

7.2.1 Analysis of dive phases of marine animals

As already mentioned during the description in Section 2.2 diving depth data sets are, for example, used to detect dive phases (descent and ascent) of marine animals. An approach to determine such phases in TDR data is implemented in the R package `diveMove` (cf. Luque, 2007; Luque and Fried, 2011). In the current version (1.4.1) the procedure

starts with heuristically splitting the diving depth time series into dives using a depth threshold of three meters and fitting a smoothing spline to each dive (cf. Figure 7.2A). Therefore, in this step, the diving depth is the response variable modelled over time with a piecewise spline regression, that is, we are in the situation described in Section 4.3.1 and the regression function is given by

$$f(x) = \sum_{\ell=1}^{L} s_\ell(x) \mathbb{1}_{I_\ell}(x),$$

where $I_1, \ldots, I_L \subset [a, b]$ are L intervals corresponding to the pieces of the time axis found with the depth threshold and s_ℓ are the spline functions on the respective intervals.
Afterwards the derivative of the smoothing spline is used to identify the descent and ascent phase of each dive. This determination can be problematic since the uniqueness of the turning point depends on the choice of the smoothing parameter. This can be seen in Figure 7.2B, which shows derivatives of splines fitted to the dive second from right in Figure 7.2A with 33 observations. For a smoothing parameter chosen via data-driven cross-validation the derivative of the smoothing spline is quite wiggly and crosses the interesting region around the zero derivative several times. For a manually chosen (larger) smoothing parameter the derivative gets smoother and the zero line is only crossed once. However, such a manual choice is subjective and might be a difficult task for users.
If we replace the smoothing spline in the first analysis step and estimate each s_ℓ as a unimodal spline, in explicit, using piecewise unimodal regression (cf. Section 4.3.1), the derivative has only one sign change and the turning point from descent to ascent is unique, irrespective of the tuning parameter value. We fit a second order difference penalized spline with $g = 25$ inner knots. When applying the function `unireg` (see Section 5.1), the argument `abstol` is set to 0.01, so that the model coefficients and the variance are estimated iteratively, starting with an initial variance estimate of `sigmasq=2`. The derivative of the resulting unimodal spline is also shown in Figure 7.2B. An obvious and welcome side effect of the unimodality constraint is that in contrast to the smoothing spline approach the choice of the tuning parameter has per construction no influence on the uniqueness of the turning point. Tuning can be done via data-driven REML estimation and the user is not confronted with this task.
Since the animal needs to come back to the surface to breathe and the time series is divisible into the individual diversely shaped dives, piecewise unimodal regression is a suitable approach for this example. On the one hand, the different shapes of the dives make the application of L_0-deco impossible, while there is on the other hand no need to

employ a more complex deconvolution model with varying peak shapes since there is no overlap of the unimodal curves.

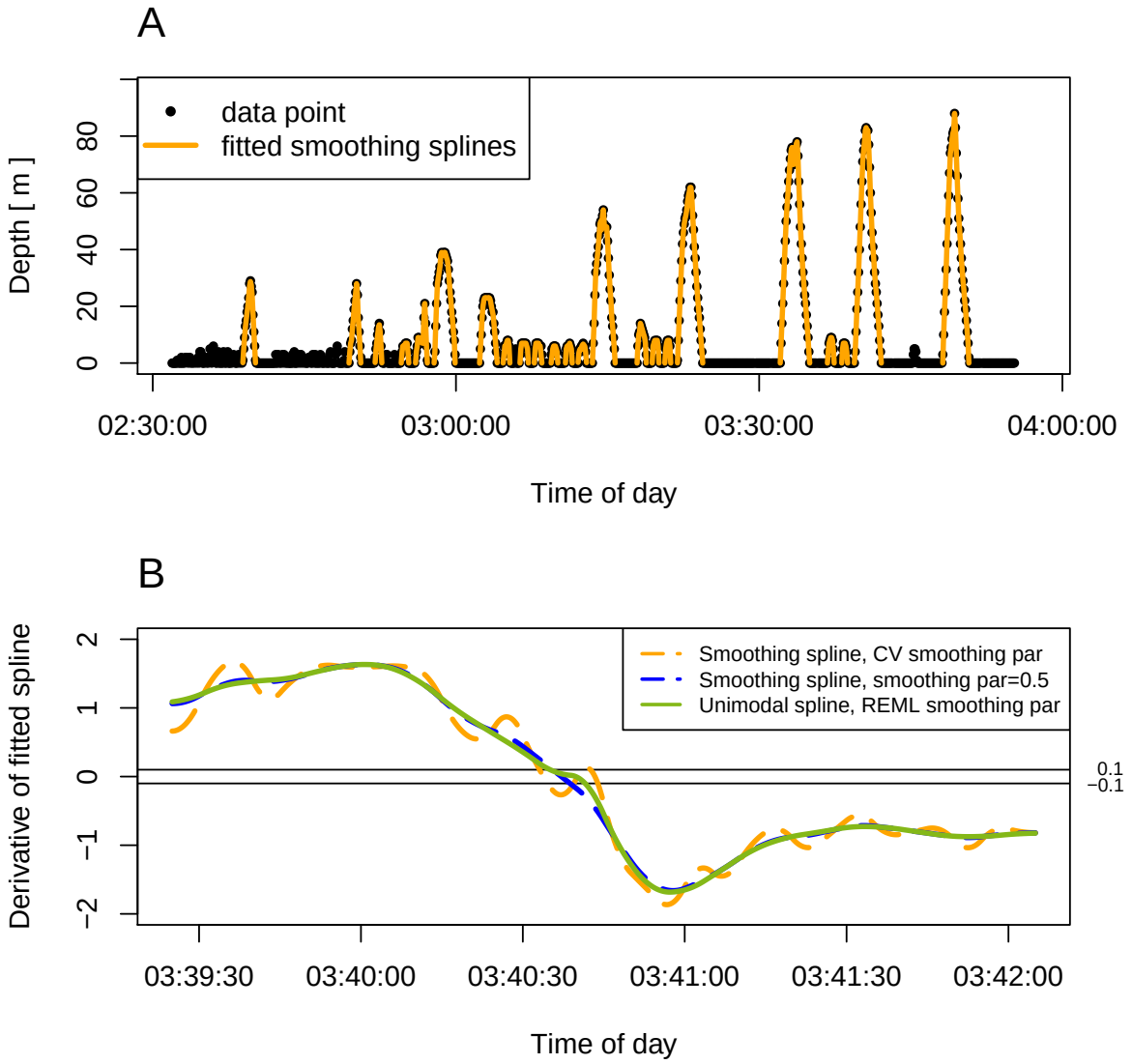


Figure 7.2: **Spline regression for the diving depth example.** (A) shows an excerpt from data set *divesTDR* and the smoothing splines fitted with R package `diveMove` (version 1.4.1, Luque, 2007). (B) shows derivatives of two fitted smoothing splines (smoothing parameter chosen via cross-validation (CV) and manually) and a unimodal spline (tuning parameter chosen via REML) for the dive second from right in (A). The x-axis is time of day on January 6th 2002.

7.2.2 Astroparticle physics data analysis

The application example from astroparticle physics deals with data recorded by the FACT telescope as described in Section 2.3. The loading curves caused by each photon hitting a FACT camera pixel are known to have a unimodal shape. As already noted, single and multiple photons can arrive at any time. Employing a deconvolution model is suitable since divisibility into unimodal pieces can be excluded and the measured voltage is an accumulation of several peaks (loading curves). It is known that each photon induces the same peak in the voltage, which can roughly be described by the parametric wave of the form in Equation (2.1). For our purpose the parameters γ, n_p and t_0 can be fixed at certain values. In explicit, the baseline γ can be set to 0 since the deconvolution model already takes care of the fact that the individual waves start at a higher baseline voltage due to the convolution. In addition, the deconvolution model estimates implicitly the arrival time and the number of photons, so that t_0 and n_p can be set to 0 and 1, respectively, and each single wave is described by

$$U_{\text{single}}(t) = U_0 \cdot \mathbb{1}_{[0,\infty)}(t) \cdot \left(1 - e^{-\frac{t}{\xi_1}}\right) e^{-\frac{t}{\xi_2}}. \tag{7.1}$$

Thus, we model the data set from Figure 2.2B with the parametric L_0-deconvolution model from Section 4.3.2 with regression function

$$f(x_i) = \sum_{k=1}^{n_s} s_k a_{i-k}, \; i = 1, \ldots, 250,$$

using $s(x|\boldsymbol{\beta}) = U_0 \left(1 - e^{-\frac{x}{\xi_1}}\right) e^{-\frac{x}{\xi_2}}$ as the parametric peak shape. We estimate $\boldsymbol{\beta} = (U_0, \xi_1, \xi_2)'$ during the iterative algorithm in function `parL0deco` (cf. Section 5.3.1), starting with the parameter estimates derived in Buß (2013): $\boldsymbol{\beta}^{(0)} = (17.41, 4.745, 31.81)'$. In iteration k the current peak shape $\boldsymbol{s}^{(k)}$ is obtained via interim evaluations of $s(x|\hat{\boldsymbol{\beta}}^{(k-1)})$ at $x_1 = 0, \ldots, x_{151} = 150$ (that is, $n_s = 151$), which is roughly the region with non-zero voltage of the wave.

By visual inspection we choose a tuning parameter value for the L_0-penalty of $\kappa = 0.01$. The result is shown in Figure 7.3. Seven input pulses are estimated clearly different from zero and the model represents the data quite well. The parameters of the wave form are estimated slightly different from the starting values as $\hat{U}_0 = 19.08$, $\hat{\xi}_1 = 5.66$, $\hat{\xi}_2 = 28.79$. The resulting wave form is a little higher in the increasing and a little lower in the decreasing part of the peak than the initial peak shape.

Again, there is no need to employ the more complex varying L_0-deco model. Compared to the diving depth analysis, convolution is certainly present in this application, but the individual peaks all have the same shape so that the simpler deconvolution model with identical peaks suffices to describe such kind of data. If no information about the wave form had been present, unimodal L_0-deco could have been used.

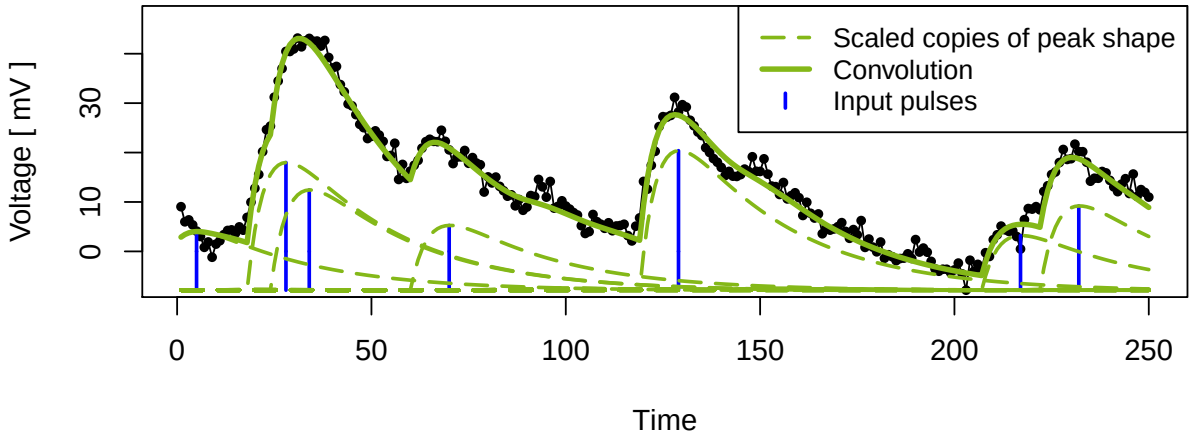


Figure 7.3: **FACT time series of length 250 and fitted deconvolution model.** The individual peaks of the fitted L_0-deconvolution model with parametric shapes are given by the dashed green lines and the accumulated signal by the solid green line. The input pulses and their heights are marked with vertical blue bars. The peak shape is modelled with the parametric wave function of form (7.1) and each of the seven peaks is a scaled version of it.

7.2.3 Breath gas analysis with ion mobility spectrometry

In breath gas analysis each spectrum of an IMS-MCC measurement consists of only a few peaks, which are mostly well-separated (see spectrum A and B in Figure 2.2C and D). The here analysed spectra both have 2499 observations.
Our first approach to model this kind of data is piecewise unimodal regression, that is, the regression function f from model (4.3) is given by

$$f(x) = \sum_{\ell=1}^{L} s_\ell(x) \mathbb{1}_{I_\ell}(x).$$

First, we obtain an estimate of the error variance from the measurements $y_{36}, \ldots, y_{700}$,

since it is known that these will definitely contain no peak. The first 35 observations are discarded because the movements of the ion shutter induce some fluctuations in the signal that do not correspond to the usual measurement error of the device. We fix σ^2 at the estimated value prior to model fitting. The first 700 of the 2499 observations are then discarded for further analyses.

For spectrum A, we determine $L = 3$ unimodal pieces using a threshold of 50 on the measured voltages and fit separate cubic unimodal splines with ridge penalty and $g = 100$ inner knots (function `unireg` with arguments `tuning=FALSE`, `penalty="diag"`, `abstol=NULL`). In Figure 7.4A we see that each of the three peaks is reproduced nicely using this procedure.

Problems can occur when the peaks from different molecules are close to each other as seen, for example, when applying the same procedure to spectrum B (see Figure 7.4B). The third and fourth peak are so close that their tails overlap. Thus, the intensity measured in between them results from both types of molecules and reflects the accumulation of both concentrations. This cannot be modelled appropriately with piecewise unimodal regression.

One might think of employing deconvolution models instead, which are able to handle overlapping as well as non-overlapping peaks. The peaks in IMS data cannot be said to have the same peak shape. Kopczynski et al. (2012), for example, apply a mixture model of inverse Gaussian distributions and thus, describe each peak with a different set of distribution parameters. Therefore, we propose fitting a deconvolution model with varying peak shapes. This reflects the accumulated intensities of the overlapping peaks as well as the diverse shapes of the peaks. Since the number of peaks is not known a priori, varying L_0-deco (cf. Section 4.3.4) is preferred to the additive unimodal regression.

The variance of the measurement error is estimated as described above and the first 700 observations are again discarded afterwards. The regression function f from model (4.3) is described by

$$f(x_i) = \sum_{j=1}^{d} s_j(x_i) a_j,$$

and is estimated with R function `varL0deco` (cp. Section 5.3.4) with $g = 200$ inner knots, spline degree $k = 3$ and tuning parameter $\kappa = 0.002$. We choose a higher number of inner knots, here, since the spline functions span the whole x-range in contrast to the piecewise splines above.

Figure 7.5 shows the estimated peaks in spectrum A and the fitted global function. Figure 7.6 shows the analogous plots for spectrum B of the IMS data set. As expected, three, respectively four peaks are modelled with one unimodal spline each. Comparing the convolution model to the piecewise model it is obvious that the two close peaks of spectrum B are much better represented by the deconvolution model.

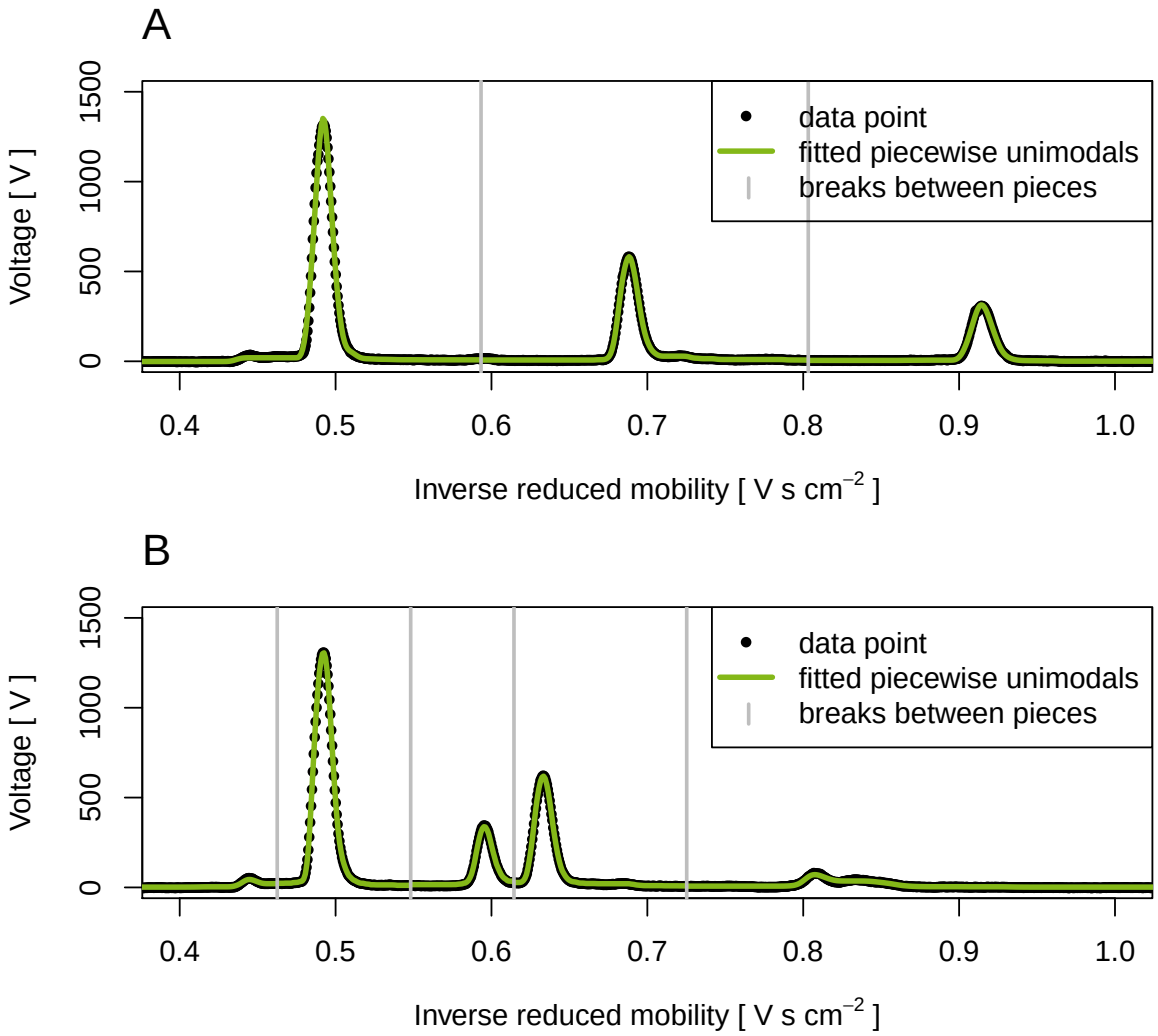


Figure 7.4: **Close-ups of IMS spectra A and B with fitted piecewise unimodal regressions.**
In both cases, the x-axis was divided into pieces according to a threshold of 50 volt on the intensities. The breaks between the pieces are indicated by vertical lines. A unimodal ridge-penalized spline was fitted to each of the pieces.

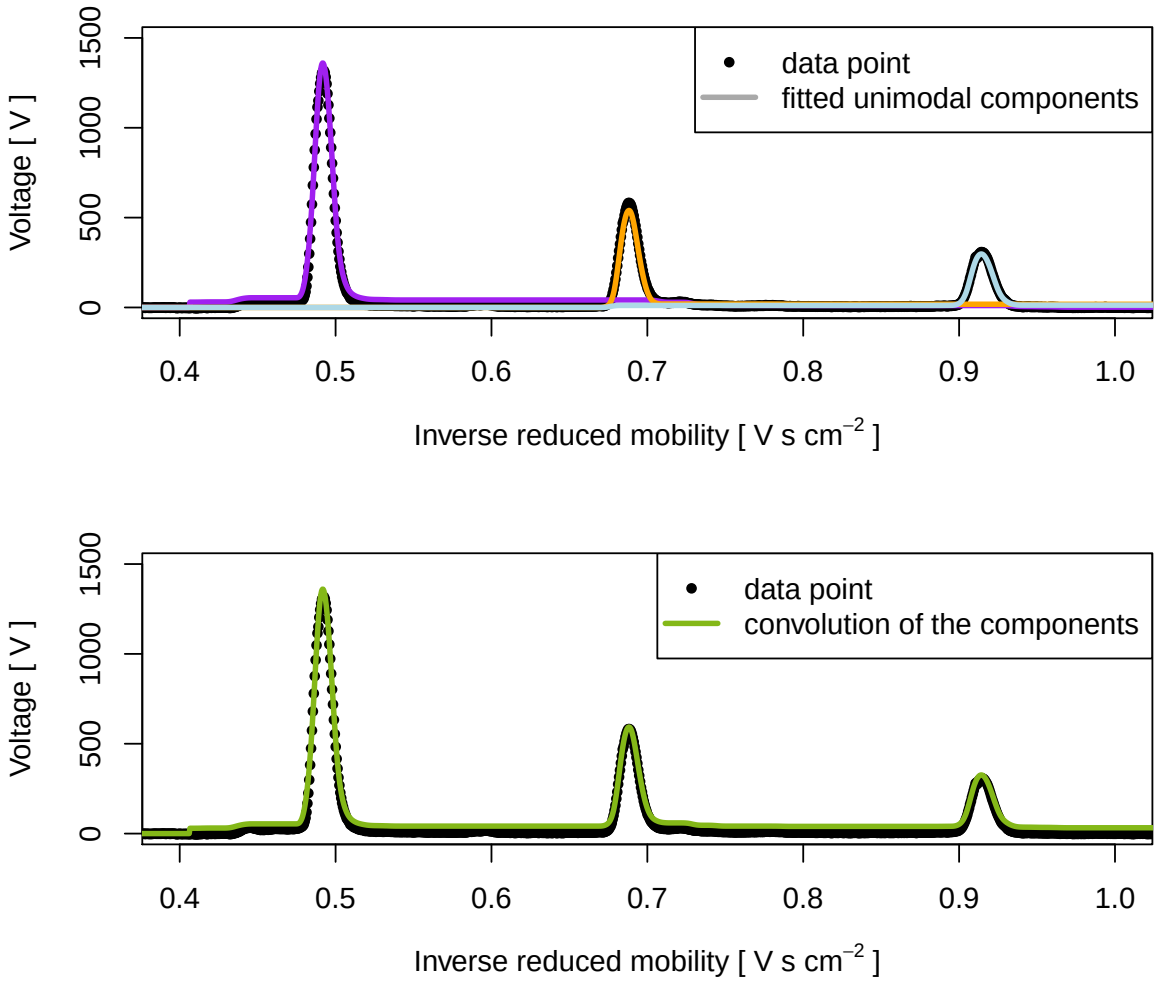


Figure 7.5: **Close-up of IMS spectrum A and L_0-deconvolution model fit with different peak shapes.**
Top: deconvolved peaks of different shapes. Each of the peaks (marked with different colours) is a unimodal spline regression with ridge penalty. Bottom: fitted global function, i.e., the convolution of the above peaks.

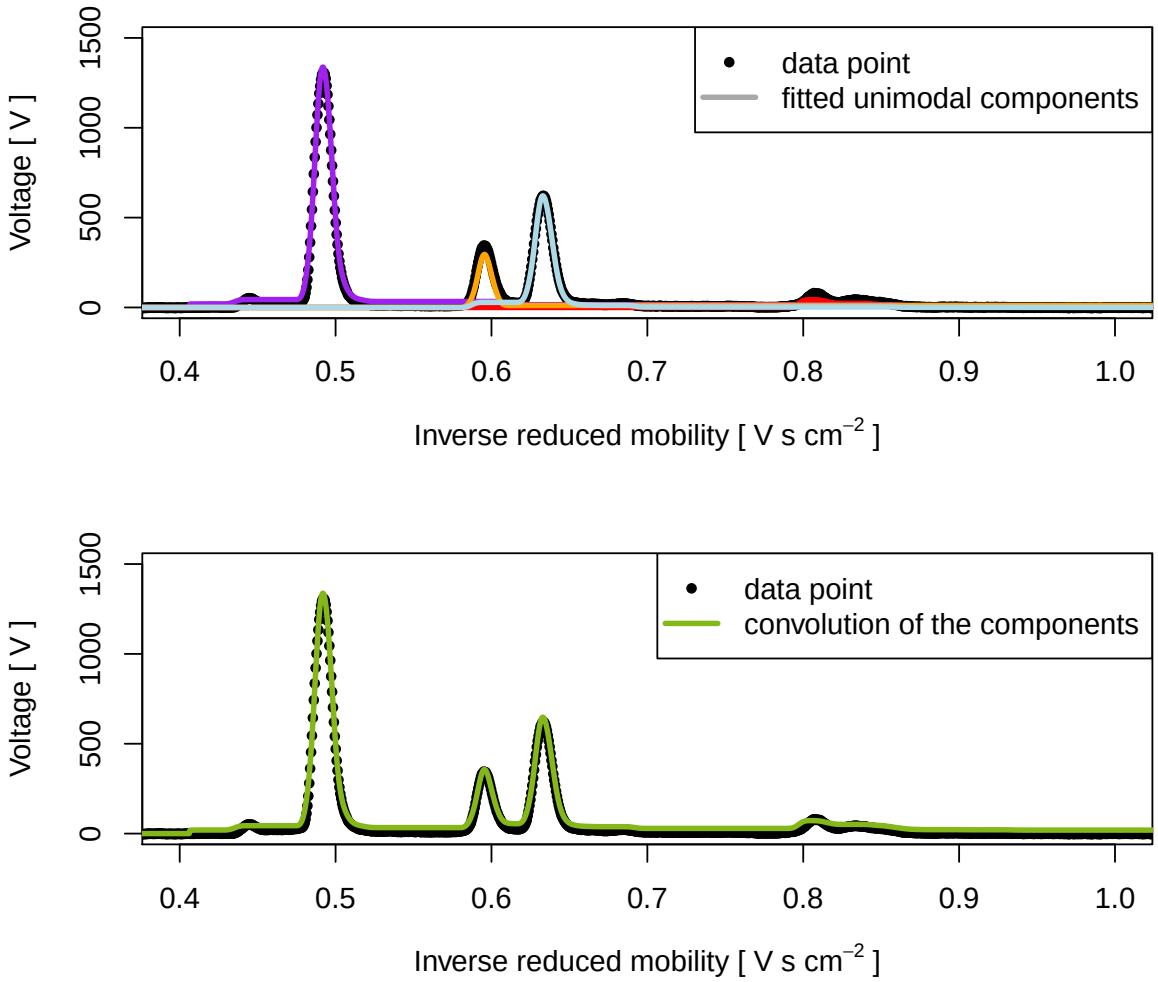


Figure 7.6: **Close-up of IMS spectrum B and L_0-deconvolution model fit with different peak shapes.**
Top: deconvolved peaks of different shapes. Each of the peaks (marked with different colours) is a unimodal spline regression with ridge penalty. Bottom: fitted global function, i.e., the convolution of the above peaks.

7.3 Further utilization of the proposed methodology

Parts of the methodology proposed in Chapters 3 and 4 were used in three final theses, which were co-supervised by the author of this PhD thesis. The results will be briefly summarised in the following subsections.

7.3.1 Robust unimodal spline regression

In her Master's thesis Vanessa Baumann conducts a simulation study to compare several non-robust and robust as well as unimodal and non-unimodal spline regression approaches (Baumann, 2014). The datasets are simulated in the style of Cantoni and Ronchetti (2001) with three sample sizes, two different true mean functions (one of them unimodal), five error distributions and five probabilities for outliers ranging from 0% to 30%. The performance of the regression approaches is assessed in terms of mean squared error.

The simulation results show that the unimodal regression clearly benefits from the robustifications described in Section 3.7.2. For small to moderate outlier probabilities the robust unimodal regression estimation procedures perform similar to the non-unimodal M-estimator by Lee and Oh (2007) for the unimodal true mean function. For high outlier probabilities the S-estimator of Tharmaratnam et al. (2010) outperforms the latter approaches. Thus, for applications with high expected outlier percentage, it would be interesting to develop a unimodal regression approach based on S-estimation.

7.3.2 Mixture of constant and unimodal regression

Hannah Bürger's Bachelor's thesis (Bürger, 2012) comprises a simulation study on the performance of the mixture model of regressions in Equation (4.2). The simulated datasets are intended to reflect experiments from systems biology, where a stimulus is given to populations of cells and the fraction of phosphorylated proteins in the cells is measured at several time points after the stimulus. It is known that the cell populations are inhomogeneous in the sense that there are cells, which react to the stimulus by phosphorylation of the cell proteins, and cells, which do not react or only very weakly. The fraction of phosphorylated proteins in the reacting cells will most probably have a unimodal course over time since proteins are dephosphorylated once the effect of the stimulus wears off. Thus, the mixture model of constant and unimodal regression seems appropriate in this context.

The simulation study by Hannah Bürger varies the shape of the unimodal mean function, the maximum distance between constant and unimodal mean function, the standard deviations in each subpopulation, the mixture weights and the observed time points. The simulation results are evaluated with respect to bias, mean squared error and misclassification rate. In general, the true shape of the unimodal component is recovered quite well and the standard deviations are estimated more accurately than the level of the constant component and the mixture weights. The shape of the unimodal mean function, the maximum distance between constant and unimodal mean function (the higher the better) and the observed time points (the more the better) are found to influence the quality of the estimated unimodal component, while varying the other parameters does not alter the results much.
Though this Bachelor's thesis only examines a special case of the mixture model of unimodal regressions, it indicates the usefulness of the methodology for inhomogeneous population scenarios.

7.3.3 Additive unimodal regression as an intermediate step for classification of IMS data sets

Laura Lange uses the additive unimodal regression model (4.5) presented in Section 4.3.3 in her Master's thesis (Lange, 2015) as a pre-step for classification of IMS datasets. The 119 data sets are known to fall into 12 different classes.
For each spectrum of a data set, the number L of peaks is first estimated using a heuristic algorithm. Then, the spectrum is deconvolved with additive unimodal regression using $L-1$, L and $L+1$ unimodal components. The fit with minimal AIC is chosen for the subsequent analyses, which turns out to be the model with $L-1$ components in most of the cases. The data matrices are smoothed with this procedure and the smoothed intensities are used in several clustering steps to generate a list of peaks present in at least one of the data sets. The intensities of these peaks in each data set are used as feature values for the classification of the IMS data sets via support vector machines. Tuning of the support vector machine is done using leave-one-out cross validation and results in a misclassification rate of about nine percent, namely, eleven out of 119 data sets are misclassified. This is a very satisfactory result, especially when reflecting that there are a lot of dependent steps in the analysis chain, each of which could also be conducted with a competing approach (for example classification trees instead of support vector machines).

Regarding the modelling step Lange (2015) discovers from extensive visual comparisons of observed and fitted intensities that the additive unimodal regression model fits the structures in the data well in most cases, but that it also sometimes fails to model a distinct peak, which is found by the heuristic. She suggests to modify the backfitting algorithm in a way that the components are constrained to have their peaks, where the heuristic finds them, for example, with the help of suitable penalties. This is an interesting topic for future research. A comparison to the varying L_0-deco approach, which estimates the number of peaks simultaneously to the peaks themselves, might also provide additional insights.

8 Summary and outlook

This Ph.D. thesis addressed non-parametric shape-constrained regression, in particular unimodal regression and possible extensions. We discussed the usefulness of this approach for different fields of application, where the predictor-response relationship ranged from shapes with a single mode, over piecewise unimodality to accumulations of identically or diversely shaped unimodal functions. Chapter 2 introduced real data examples from four such applications areas, namely dose-response analysis, marine biology, astroparticle physics and breath gas analysis.

As a feasible realization of the non-parametric concept of unimodal regression we proposed a range of semi-parametric shape-constrained spline regression techniques. To motivate this choice, Chapter 3 gave an introduction to spline functions and illustrated their approximation power using the characteristics of Bernstein-Schoenberg splines. In addition, we showed that unimodality of a spline function is guaranteed by the shape-preservation property of B-splines when placing a linear constraint on the vector of regression coefficients. Thus, least squares estimates can be found with quadratic programming algorithms.

The problem of knot placement was addressed by choosing a large number of knots and putting a penalty on the B-spline coefficients to avoid overfitting. As an automatic choice of the tuning parameter for arbitrary penalties we developed a restricted maximum likelihood (REML) estimate. Here, the second and third order difference penalties were used, which implicitly penalize against a linear and a quadratic model. We generalized this approach to allow for penalization against arbitrary parametric functions (for example, a sigmoid E_{max} model in dose-response applications). All penalties favour some parametric model, but allow for departures from this model, when the data suggest so.

We also proposed a Bayesian approach to unimodal spline regression which relies on Monte Carlo random sampling. Since the joint posterior distribution of the B-spline coefficients, the tuning parameter and the mode can be factorized into the marginal posteriors, samples can be drawn by successively sampling from the marginals. Thus,

the samples from the joint posterior are uncorrelated.
By using iteratively re-weighted least squares estimation, it is also possible to robustify the frequentist unimodal spline regression approaches for applications where outliers might occur.
In Chapter 4 we further developed the unimodal regression approaches and made them a building block in regression models for multimodal applications. For multimodality arising from an inhomogeneous population it is possible to use a mixture model of unimodal regressions, while there are several approaches for estimating a multimodal mean predictor-response relationship in a homogeneous population. Table 4.1 summarizes the different data situations and gives recommendations for the model choice for the homogeneous case. If there is only one mode, the unimodal regression approach from Chapter 3 is directly applicable. Piecewise unimodal regression can be used whenever the underlying process is divisible, in explicit, when there is no overlap between adjacent peaks as, for example, in diving depth data. On the contrary, a deconvolution model is preferable when there is overlap between the unimodal functions and they accumulate to the observed values. Deconvolution with L_0-penalty is applicable whenever the peaks have the same shape across the whole data set, as, for example, in the astroparticle physics example. If this is not the case, for example, in breath gas analysis data, the deconvolution models with varying peak shapes should be be used. Due to an increased computational burden for the additive model we recommend it only in applications where the number of peaks is known. In all other cases the varying L_0-deco approach is the method of choice and provides estimates of the number of peaks, their heights and shapes.
Chapter 5 was dedicated to the implementation of the proposed methodology. The unimodal spline regression approaches from Chapter 3 are implemented in R package `uniReg`, freely available on CRAN. Its main function for model fitting as well as the implementations of the Bayesian Monte Carlo sampling and the deconvolution models from Chapter 4.3 were described to give a picture of the numerical and implementational challenges (and our solutions), but also to introduce the usage.
The presented R package was also used for an extensive simulation study to compare the proposed unimodal regression methods among themselves and to other frequently used regression models in the dose-response analysis context. The results, summarized in Chapter 6, showed that the combination of unimodal shape constraint and a penalty is quite promising, since adding the constraint improves the fit. Averaging over the posterior mode distribution as in the Bayesian approach can improve the reproduction of the true function characteristics even further.

When applying our methods to a real dose-response data set of McLaren et al. (1990) in Chapter 7, we obtained evidence that penalized unimodal spline regression is to some extent able to safeguard against model mis-specification. The development of the proposed methods was initially motivated by dose-response analysis, but the methodology is very general and can be used in many other areas of application. We analysed data from three different multimodal applications – marine biology, astroparticle physics and breath gas analysis – to illustrate that unimodal regression is not only useful when a unimodal relationship between dependent and independent variable is likely, but also as a building block in situations where the relationship is multimodal and has increasing complexity: from piecewise unimodality to accumulations of identically or diversely shaped unimodal functions.

In comparison to parametric models as, for example, in dose-response analysis or the wave form in Section 7.2.2, spline regression is a very flexible tool. Prior knowledge about the shape of the underlying relationship can be incorporated by using a shape constraint (here: unimodality) and/or by a penalty (ridge penalty or penalizing against a parametric function). Another nice characteristic of splines is the simplicity of calculating derivatives and that the derivatives of shape-constrained splines also "inherit" shape properties. In the case of the marine biology data the monotonicity of the first derivative simplifies the subsequent analyses, namely the detection of descent and ascent phase of the dive by finding the zero of the derivative.
The analyses provide an indication for the usefulness of unimodal regression in the presented and in further applications. Of course there are situations where unimodality is not as likely as, for example, in the case of the peaks in IMS data or the FACT loading curves. Actually, one could argue that the dive of a marine animal is not strictly unimodal since the animal might also descend to a certain depth, make some smaller upward and downward movements and then ascend to the surface again. Those wiggles at the bottom are flattened out by the unimodal spline approach and the turning point that fits the data best divides the dive into descent and ascent. Thus, the unimodal model may be regarded as a simplification of a dive, but it is a suitable one and provides an automated estimation process for biological scientists. Adaptations of the method to aims other than the division into the two phases are also possible.
The results of this thesis indicate that the two approaches for increasing the efficiency of non- or semi-parametric regression approaches mentioned in the introduction, namely penalization and shape constraints, can be reasonably combined and yield proper esti-

mations in standard regression tasks as well as in deconvolution problems.

An interesting aspect for future research are confidence regions for unimodal spline regression. In the Bayesian approach, credible regions are straightforward to obtain (and have been included in the analysis of the dose-response data set, see Figure 7.1), but for the frequentist methods the derivation of the estimator's distribution under the shape constraint is not. Thus, bootstrapping suggests itself, but there will be need to investigate its theoretical properties in the presence of the shape constraint and its computational feasibility. Bootstrap techniques would also be suitable for the derivation of statistical tests. One might, for example, be interested in testing for unimodality versus monotonicity.

Furthermore, we expect the unimodal penalized spline regression to be large sample consistent. The considerations regarding the approximation power of spline functions in Section 3.2.2 and regarding the flexibility of unimodality-constrained splines shown in Section 3.4 (cf. Figure 3.3) are good indicators. A proof is beyond the scope of this work, but might be performed along the lines of Meyer (2012). The author shows therein that the convergence rate for monotonicity constrained penalized splines is at least that for unconstrained penalized splines, which were shown to be consistent regression function estimators in Claeskens et al. (2009).

So far, all L_0-deconvolution models have been tuned manually and there is still need to explore methods for automated choices of the tuning parameter. Due to the iterative estimation of the peak shapes and the input pulses, it is not straightforward to define a measure of the effective number of model parameters for calculating a model selection criterion like AIC. The same holds for the existence of a hat matrix, which would provide a short-cut for calculating a cross-validated tuning parameter. Thus, cross-validation would actually have to be performed, making the already computationally demanding deconvolution models even more so. This is why we think that a Bayesian formulation of the deconvolution models might be more suitable. Here, sparsity of the input pulses can be achieved using spike-and-slab priors. First results are quite promising and encourage future research in this direction. This is also the reason why we did not further pursue the cross-validation option and did not conduct a simulation study for the multimodal regression models, but settled in this thesis for demonstrating the performance of the proposed models on real data.

Additionally, subsequent analysis steps like classification or integration are often common in the described multimodal situations. Hence, systematic evaluations of the impact

of the modelling step on the final outcome are needed. Such performance studies could be conducted, for example, along the lines of Hauschild et al. (2013).

Throughout this thesis, we mentioned at many places that the proposed unimodal regression approaches are computationally demanding, the burden obviously growing with the number of possible modes of the coefficient vector and even multiplying with repeated execution for the multimodal regression models and when performing model selection via AIC or tuning via cross-validation. Version 1.1 of R package `uniReg` already enables the use of parallelization, exploiting that the coefficient vectors with different modes can be estimated independently. Other ideas for speeding up the estimation process are, for example, locally defined unimodal splines for the deconvolution models with divers peak shapes (as are already used for those with identically shaped peaks) or using some kind of shape-penalty instead of a strict shape constraint.

The shape penalty approach eases tuning of the smoothness penalty. We already mentioned the asymmetric penalty used by Eilers (2005) to induce log-concavity. Our simulation study indicated that the unimodal shape constraint performs better on average over a wide range of true unimodal relationships. This might be due to function shapes like convex-monotonicity which log-concave splines cannot fit. In other scenarios with strictly unimodal or even truly log-concave relationships, the approach of Eilers (2005) might perform better than unimodal regression. In addition, it is computationally much less demanding. A combination of both approaches is a unimodal asymmetric penalty. We suggest a thorough comparison of the three approaches with regard to statistical and computational aspects.

Bibliography

Alfons, A. (2012). *cvTools: Cross-validation tools for regression models*. R package version 0.3.2.

Anderhub, H., Backes, M., Biland, A., Boccone, V., Braun, I., Bretz, T., Buß J., Cadoux, F., Commichau, V., Djambazov, L., Dorner, D., Einecke, S., Eisenacher, D., Gendotti, A., Grimm, O., von Gunten, H., Haller, C., Hildebrand, D., Horisberger, U., Huber, B., Kim, K.-S., Knoetig, M. L., Koehne, J.-H., Kraehenbuehl, T., Krumm, B., Lee, M., E.Lorenz, Lustermann, W., Lyard, E., Mannheim, K., Meharga, M., Meier, K., Montaruli, T., Neise, D., Nessi-Tedaldi, F., Overkemping, A.-K., Paravac, A., Pauss, F., Renker, D., Rhode, W., Ribordy, M., Roeser, U., Stucki, J.-P., Schneider, J., Steinbring, T., Temme, F., Thaele, J., Tobler, S., Viertel, G., Vogler, P., Walter, R., Warda, K., Weitzel, Q., and Zaenglein, M. (2013). Design and operation of FACT - the first G-APD Cherenkov telescope. *Journal of Instrumentation*, 8(06):P06008.

Barlow, R. E., Bartholomew, D. J., Bremner, J. M., and Brunk, H. D. (1972). *Statistical Inference Under Order Restrictions*. John Wiley, London.

Baumann, V. (2014). Kombination von robuster und unimodaler Spline-Regression. Master's thesis, Faculty of Statistics, TU Dortmund University.

Bickel, P. J. and Fan, J. (1996). Some problems on the estimation of unimodal densities. *Statistica Sinica*, 6:23–45.

Biland, A., Bretz, T., Buß J., Commichau, V., Djambazov, L., Dorner, D., Einecke, S., Eisenacher, D., Freiwald, J., Grimm, O., von Gunten, H., Haller, C., Hempfling, C., Hildebrand, D., Hughes, G., Horisberger, U., Knoetig, M. L., Krähenbühl, T., Lustermann, W., Lyard, E., Mannheim, K., Meier, K., Mueller, S., Neise, D., Overkemping, A. K., Paravac, A., Pauss, F., Rhode, W., Röser, U., Stucki, J. P., Steinbring, T., Temme, F., Thaele, J., Vogler, P., Walter, R., and Weitzel, Q. (2014). Calibration

and performance of the photon sensor response of FACT - the first G-APD Cherenkov telescope. *Journal of Instrumentation*, 9(10):P10012.

Birge, L. (1997). Estimation of unimodal densities without smoothness assumptions. *The Annals of Statistics*, 25(3):970–981.

Bödeker, B. and Baumbach, J. I. (2009). Analytical description of ims-signals. *International Journal for Ion Mobility Spectrometry*, 12(3):103–108.

Bornkamp, B. (2010). *SEL: Semiparametric elicitation.* R package version 1.0-2.

Bornkamp, B., Bretz, F., Dmitrienko, A., Enas, G., Gaydos, B., Hsu, C.-H., et al. (2007). Innovative approaches for designing and analyzing adaptive dose-ranging trials. *Journal of Biopharmaceutical Statistics*, 17(6):965–995.

Bornkamp, B., Pinheiro, J., and Bretz, F. (2016). *DoseFinding: Planning and Analyzing Dose Finding experiments.* R package version 0.9-14.

Boyarshinov, V. and Magdon-Ismail, M. (2006). Linear time isotonic and unimodal regression in the L_1 and L_∞ norms. *Journal of Discrete Algorithms*, 4:676–691.

Braun, W. J. and Hall, P. (2001). Data sharpening for nonparametric inference subject to constraints. *Journal of Computational and Graphical Statistics*, 10(4):786–806.

Bretz, F., Pinheiro, J., and Branson, M. (2005). Combining multiple comparisons and modeling techniques in dose-response studies. *Biometrics*, 61:738–748.

Bro, R. and Sidiropoulos, N. D. (1998). Least squares algorithms under unimodality and non-negativity constraints. *Journal of Chemometrics*, 12(4):223–247.

Brunk, H. D. (1955). Maximum likelihood estimates of monotone parameters. *The Annals of Mathematical Statistics*, 26(4):607–616.

Buckland, S. T., Burnham, K. P., and Augustin, N. H. (1997). Model selection: an integral part of inference. *Biometrics*, 53(2):603–618.

Bürger, H. (2012). Ein EM-Algorithmus für die Mischung von konstanter und unimodaler Regression: Eigenschaften und Effizienz. Bachelor's thesis, Faculty of Statistics, TU Dortmund University.

Buß, J. B. (2013). Fact - signal calibration: Gain calibration and development of a single photon pulse template for the fact camera. Diploma thesis, Faculty of Physics, TU Dortmund University.

Cantoni, E. and Ronchetti, E. (2001). Resistant selection of the smoothing parameter for smoothing splines. *Statistics and Computing*, 11(2):141–146.

Carnicer, J. M. and Pena, J. M. (1994). Totally positive bases for shape preserving curve design and optimality of b-splines. *Computer Aided Geometric Design*, 11:633–654.

Chang, I.-S., Hsiung, C. A., Wu, Y.-J., and Yang, C.-C. (2005). Bayesian survival analysis using Bernstein polynomials. *Scandinavian Journal of Statistics*, 32:447–466.

Claeskens, G., Krivobokova, T., and Opsomer, J. D. (2009). Asymptotic properties of penalized spline estimators. *Biometrika*, 96(3):529–544.

D'Addario, M., Kopczynski, D., Baumbach, J. I., and Rahmann, S. (2014). A modular computational framework for automated peak extraction from ion mobility spectra. *BMC Bioinformatics*, 15(1).

de Boor, C. (1978). *A practical guide to splines*, volume 27 of *Applied Mathematical Sciences*. Springer, New York.

de Rooi, J. and Eilers, P. (2011). Deconvolution of pulse trains with the L_0 penalty. *Analytica Chimica Acta*, 705:218–226.

de Rooi, J., Ruckebusch, C., and Eilers, P. (2014). Sparse deconvolution in one and two dimensions: Applications in endocrinology and single-molecule fluorescence imaging. *Analytical Chemistry*, 86:6291–6298.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM-algorithm. *Journal of the Royal Statistical Society B*, 39:1–38.

Dierckx, P. (1993). *Curve and surface fitting with splines*. Oxford Science Publications. Clarendon Press, Oxford.

Dunson, D. and Neelon, B. (2003). Bayesian inference on order-constrained parameters in generalized linear models. *Biometrics*, 59(2):286–295.

Eilers, P. H. C. (2005). Unimodal smoothing. *Journal of Chemometrics*, 19:317–328.

Eilers, P. H. C. and Marx, B. D. (1996). Flexible smoothing with B-splines and penalties. *Statistical Science*, 11(2):89–121.

Fahrmeir, L., Kneib, T., Lang, S., and Marx, B. (2013). *Regression - Models, Methods and Applications*. Springer.

Frisén, M. (1986). Unimodal regression. *The Statistician*, 35:479–485.

Frisén, M. and Goteborg, S. (1980). U-shaped regression. In Barrit, M. M. and Wishart, D., editors, *COMPSTAT 1980.: Proceedings in Computational Statistics*, volume 4, pages 304–307, Heidelberg, Wien. COMPSTAT, Physica.

Genz, A., Bretz, F., Miwa, T., Mi, X., Leisch, F., Scheipl, F., et al. (2012). *mvtnorm: Multivariate Normal and t Distributions*. R package version 1.0-5.

Goodman, T. N. T. (1995). Bernstein-Schoenberg operators. In Daehlen, M., Lyche, T., and Schumaker, L., editors, *Mathematical methods for curves and surfaces*. Vanderbilt University Press, Nashville, London.

Grenander, U. (1956). On the theory of mortality measurement, Part II. *Skandinavisk Aktuarietidskrift*, 39:125–153.

Grün, B. and Leisch, F. (2008). Flexmix version 2: Finite mixtures with concomitant variables and varying and constant parameters. *Journal of Statistical Software*, 28(4):1–35.

Gunn, L. and Dunson, D. (2005). A transformation approach for incorporating monotone or unimodal constraints. *Biostatistics*, 6(3):434–449.

Halsey, L. G., Bost, C.-A., and Handrich, Y. (2007). A thorough and quantified method for classifying seabird diving behaviour. *Polar Biology*, 30:991–1004.

Härdle, W. and Simar, L. (2007). *Applied multivariate statistical analysis*. Springer, second edition.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer Series in Statistics. Springer, second edition.

Hauschild, A.-C., Kopczynski, D., D'Addario, M., Baumbach, J. I., Rahmann, S., and Baumbach, J. (2013). Peak detection method evaluation for ion mobility spectrometry by using machine learning approaches. *Metabolites*, 3:277–293.

Hazelton, M. L. and Turlach, B. (2011). Semiparametric regression with shape constrained penalized splines. *Computational Statistics and Data Analysis*, 55:2871–2879.

Hildenbrand, K. and Hildenbrand, W. (1986). On the mean income effect: a data analysis of the UK Family Expenditure. In Mas-Colell, A. and Hildenbrand, W., editors, *Contributions to mathematical economics: in honour of Gérard Debreu*, pages 247–268. North-Hollland, Amsterdam.

Hildreth, C. (1954). Point estimates of ordinates of concave functions. *Journal of the American Statistical Association*, 49:598–619.

Holland, P. W. and Welsch, R. E. (1977). Robust regression using iteratively reweighted least-squares. *Communications in Statistics - Theory and Methods*, 6(9):813–827.

Kelly, C. and Rice, J. (1990). Monotone smoothing and its applications to dose-response curves and the assessment of synergy. *Biometrics*, 46:1071–1085.

Khan, A. A., Perlstein, I., and Krishna, R. (2009). The use of clinical utility assessments in early clinical development. *The AAPS Journal*, 11(1):33–38.

Koenker, R. (2016). *quantreg: Quantile Regression*. R package version 5.24.

Köllmann, C. (2016). *uniReg: Unimodal penalized spline regression using B-splines*. R package version 1.1.

Köllmann, C., Bornkamp, B., and Ickstadt, K. (2014). Unimodal regression using Bernstein-Schoenberg-splines and penalties. *Biometrics*, 70:783–793. doi: 10.1111/biom.12193.

Köllmann, C., Ickstadt, K., and Fried, R. (2016). Beyond unimodal regression: modelling multimodality with piecewise unimodal regression or deconvolution models. *ArXiv e-prints*. http://arxiv.org/abs/1606.01666v1.

Kopczynski, D., Baumbach, J. I., and Rahmann, S. (2012). Peak modeling for Ion Mobility Spectrometry measurements. In *Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pages 1801–1805.

Kopczynski, D. and Rahmann, S. (2014). An online peak extraction algorithm for ion mobility spectrometry data. In Brown, D. and Morgenstern, B., editors, *Algorithms in Bioinformatics*, volume 8701 of *Lecture Notes in Computer Science*, pages 232–246. Springer Berlin Heidelberg.

Lange, L. (2015). Analyse von GC/IMS-Atemluftmessungen unter Berücksichtigung verschiedener Atemerfrischer. Master's thesis, Faculty of Statistics, TU Dortmund University.

Lee, T. C. M. and Oh, H.-S. (2007). Robust penalized regression spline fitting with application to additive mixed modeling. *Computational Statistics*, 22:159–171.

Luque, S. P. (2007). Diving behaviour analysis in R. *R News*, 7(3):8–14. Contributions from: J. P. Y. Arnould, L. Dubroca, and A. Liaw.

Luque, S. P. and Fried, R. (2011). Recursive filtering for zero offset correction of diving depth time series with GNU R package diveMove. *PLoS ONE*, 6(1):e15850.

Marsden, M. J. (1970). An identity for spline functions with applications to variation-diminishing spline approximation. *Journal of Approximation Theory*, 3:7–49.

McLaren, D. G., Bechtel, P. J., Grebner, G., Novakofski, J., McKeith, F. K., Jones, R., et al. (1990). Dose response in growth of pigs injected daily with porcine somatotropin from 57 to 103 kilograms. *Journal of Animal Science*, 68:640–651.

Meyer, M. C. (2012). Constrained penalized splines. *Canadian Journal of Statistics*, 40(1):190–206.

Meyer, M. C., Hackstadt, A. J., and Hoeting, J. A. (2011). Bayesian estimation and inference for generalised partial linear models using shape-restricted splines. *Journal of Nonparametric Statistics*, 23(4):867–884.

Morell, O., Otto, D., and Fried, R. (2013). On robust cross-validation for nonparametric smoothing. *Computational Statistics*, 28(4):1617–1637.

Neal, R. (2003). Slice sampling. *Annals of Statistics*, 31(3):705–767.

Oller-Moreno, S., Singla-Buxarrais, G., Jiménez-Soto, J., Pardo, A., Garrido-Delgado, R., Arce, L., and Marco, S. (2015). Sliding window multi-curve resolution: Application to gas chromatography-ion mobility spectrometry. *Sensors and Actuators B: Chemical*, 217:13 – 21.

Pomareda, V., Calvo, D., Pardo, A., and Marco, S. (2010). Hard modeling multivariate curve resolution using lasso: Application to ion mobility spectra. *Chemometrics and Intelligent Laboratory Systems*, 104(2):318 – 332.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Ramsay, J. O. (1988). Monotone regression splines in action. *Statistical Sciene*, 3(4):425–441.

Reinsch, C. H. (1967). Smoothing by spline functions. *Numerische Mathematik*, 10(3):177–183.

Reinsch, C. H. (1971). Smoothing by spline functions. ii. *Numerische Mathematik*, 16(5):451–454.

Rekowski, J. (2013). Gütekriterien von Methoden zur Modellierung von Dosis-Wirkungsbeziehungen unter Variation der Dosisstufen, des Stichprobenumfangs und seiner Verteilung. Diploma thesis, Faculty of Statistics, TU Dortmund University.

Rossoni, E. and Feng, J. (2006). A nonparametric approach to extract information from interspike interval data. *Journal of Neuroscience Methods*, 150:30–40.

Schoenberg, I. J. (1967). *On spline functions (with a supplement by T. N. E. Greville)*. Inequalities I. Academic Press, New York.

Seber, G. and Wild, C. (2003). *Nonlinear Regression*. Wiley Series in Probability and Statistics. Wiley.

Stout, Q. F. (2008). Unimodal regression via prefix isotonic regression. *Computational Statistics and Data Analysis*, 53:289–297.

Tauler, R. (1995). Multivariate curve resolution applied to second order data. *Chemometrics and Intelligent Laboratory Systems*, 30(1):133–146.

Tharmaratnam, K., Claeskens, G., Croux, C., and Salibián-Barrera, M. (2010). S-estimation for penalized regression splines. *Journal of Computational and Graphical Statistics*, 19(3):609–625.

Trautmann, H., Steuer, D., Mersmann, O., and Bornkamp, B. (2014). *truncnorm: Truncated normal distribution*. R package version 1.0-7.

Turlach, B. and Weingessel, A. (2013). *quadprog: Functions to solve Quadratic Programming Problems*. R package version 1.5-5, S original by Berwin A. Turlach, R port by Andreas Weingessel.

Turner, R. (2015). *Iso: Functions to Perform Isotonic Regression.* R package version 0.0-17.

Turner, T. R. and Wollan, P. C. (1997). Locating a maximum using isotonic regression. *Computational Statistics and Data Analysis*, 25:305–320.

Vogtland, D. and Baumbach, J. I. (2009). Breit-Wigner-Function and IMS-signals. *International Journal for Ion Mobility Spectrometry*, 12:109–114.

Wang, J. and Ghosh, S. K. (2012). Shape restricted nonparametric regression with Bernstein polynomials. *Computational Statistics and Data Analysis*, 56:2729–2741.

Wegmann, E. (1972). Nonparametric probability density estimation: I. a summary of available methods. *Technometrics*, 714(3):533–546.

Westhoff, M., Litterst, P., Freitag, L., Urfer, W., Bader, S., and Baumbach, J. I. (2009). Ion mobility spectrometry for the detection of volatile organic compounds in exhaled breath of lung cancer patients. *Thorax*, 64:744–748. April 2012.

Wood, S. (1994). Monotonic smoothing splines fitted by cross validation. *SIAM Journal on Scientific Computing*, 15:1126–1133.

Wood, S. N. (2011). Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society, Series B*, 73(1):3–36.

Woodworth, G. (1999). Approaches to Bayesian smooth unimodal regression. Technical report, URL: http://www.stat.uiowa.edu/~gwoodwor/unimodal.pdf.

Ye, J. (1998). On measuring and correcting the effects of data mining and model selection. *Journal of the American Statistical Association*, 93(441):120–131.

Yu, J.-w. and Tian, G.-l. (2011). Efficient algorithms for generating truncated multivariate normal distributions. *Acta Mathematicae Applicatae Sinica (English Series)*, 27(4):601–612.

Yuan, Y. and Yin, G. (2011). Dose-response estimation: A semiparametric mixture approach. *Biometrics*, 67:1543–1554.

A Additional tables

Table A.1: **Settings for data generation in the simulation study.**

no.	x-values	sample size	σ	profile
1	(0,2,4,6,8)	250	$\sqrt{4.5}$	Flat
2	(0,1,2,3,4,5,6,7,8)	252	$\sqrt{4.5}$	Flat
3	0,2,4,6,8)	250	$\sqrt{4.5}$	Linear
4	(0,1,2,3,4,5,6,7,8)	252	$\sqrt{4.5}$	Linear
5	(0,2,4,6,8)	250	$\sqrt{4.5}$	Quadratic
6	(0,1,2,3,4,5,6,7,8)	252	$\sqrt{4.5}$	Quadratic
7	(0,2,4,6,8)	250	$\sqrt{4.5}$	E_{max}
8	(0,1,2,3,4,5,6,7,8)	252	$\sqrt{4.5}$	E_{max}
9	(0,2,4,6,8)	250	$\sqrt{4.5}$	sigmoid E_{max}
10	(0,1,2,3,4,5,6,7,8)	252	$\sqrt{4.5}$	sigmoid E_{max}
11	(0,2,4,6,8)	250	$\sqrt{4.5}$	Beta 1
12	(0,1,2,3,4,5,6,7,8)	252	$\sqrt{4.5}$	Beta 1
13	(0,2,4,6,8)	250	$\sqrt{4.5}$	Beta 2
14	(0,1,2,3,4,5,6,7,8)	252	$\sqrt{4.5}$	Beta 2
15	(0,2,4,6,8)	250	$\sqrt{4.5}$	Exponential
16	(0,1,2,3,4,5,6,7,8)	252	$\sqrt{4.5}$	Exponential
17	(0,2,4,6,8)	250	$\sqrt{4.5}$	Peak
18	(0,1,2,3,4,5,6,7,8)	252	$\sqrt{4.5}$	Peak

Table A.2: **The 25 models used in the simulation study.** The table gives an abbreviation for each model and a short description of it. Additionally - depending on the model type - information is given about the number of inner knots, if the unimodality constraint is active or not, the type of the penalty, and the way of determining the tuning parameter and the mode location.

Abbr.	Model type	Inner knots	Constraint	Penalty	Estimation of λ	Mode estimation
un	cubic spline	$g = u - 5$	no	no	–	–
cn	cubic spline	$g = u - 5$	yes	no	–	minimizing RSS
ud2	cubic spline	$g = 10$	no	difference (2nd)	REML	–
cd2	cubic spline	$g = 10$	yes	difference (2nd)	REML	minimizing RSS
cda2	cubic spline	$g = 10$	yes	difference (2nd)	approx. REML	minimizing RSS
ud3	cubic spline	$g = 10$	no	difference (3rd)	REML	–
cd3	cubic spline	$g = 10$	yes	difference (3rd)	REML	minimizing RSS
cda3	cubic spline	$g = 10$	yes	difference (3rd)	approx. REML	minimizing RSS
logCon	cubic spline	$g = 10$	yes	difference (3rd)	cross-validation	–
us	cubic spline	$g = 10$	no	sigmoid E_{max}	REML	–
cs	cubic spline	$g = 10$	yes	sigmoid E_{max}	REML	minimizing RSS
csa	cubic spline	$g = 10$	yes	sigmoid E_{max}	approx. REML	minimizing RSS
udb	cubic spline (Bayes)	$g = 10$	no	difference (2nd)	posterior mean	–
cdb2	cubic spline (Bayes)	$g = 10$	yes	difference (2nd)	posterior mean	posterior mean
cdb3	cubic spline (Bayes)	$g = 10$	yes	difference (3rd)	posterior mean	posterior mean
csb	cubic spline (Bayes)	$g = 10$	yes	sigmoid E_{max}	posterior mean	posterior mean
trafo	cubic spline (Bayes)	$g = 10$	yes	difference (2nd)	posterior mean	transformation
uBPr	Bernstein polyn. ($k = 13$)	$g = 0$	no	ridge	REML	–
cBPr	Bernstein polyn. ($k = 13$)	$g = 0$	yes	ridge	REML	minimizing RSS
uBPs	Bernstein polyn. ($k = 13$)	$g = 0$	no	sigmoid E_{max}	REML	–
cBPs	Bernstein polyn. ($k = 13$)	$g = 0$	yes	sigmoid E_{max}	REML	minimizing RSS
sigE	sigmoid E_{max}	–	–	–	–	–
beta	beta	–	–	–	–	–
modAve	model averaging	–	–	–	–	–
frisen	Frisén (1986)	–	yes	–	–	minimizing RSS

B Proofs

Recall

Lemma 2. *Assuming the model*

$$\begin{aligned} \boldsymbol{\mathcal{Y}}|\boldsymbol{\beta} &\sim \mathcal{N}(\boldsymbol{B}\boldsymbol{\beta}, \sigma^2\boldsymbol{I}_n), \quad \sigma^2 > 0 \\ \boldsymbol{\beta}|\lambda &\sim \mathcal{N}_{\mathcal{M}}(\boldsymbol{\beta}_0, \lambda^{-1}\boldsymbol{\Omega}^{-1}), \quad \lambda > 0, \boldsymbol{\Omega} \text{ } pd, \mathcal{M} \subseteq \mathbb{R}^d \\ \lambda &\sim p(\lambda), \text{ } a \text{ } prior \text{ } density \text{ } on \text{ } (0,\infty), \end{aligned}$$

the marginal posterior density of λ is given by

$$p(\lambda|\boldsymbol{y}) \propto p(\lambda) \ |\lambda\boldsymbol{\Omega}|^{\frac{1}{2}} \ |\mathbf{E}_\lambda|^{\frac{1}{2}} \frac{c_\lambda^{post}}{c_\lambda^{prior}} \ \exp\left(\frac{1}{2}\mathbf{e}_\lambda'\mathbf{E}_\lambda^{-1}\mathbf{e}_\lambda - \frac{1}{2}\lambda\boldsymbol{\beta}_0'\boldsymbol{\Omega}\boldsymbol{\beta}_0\right),$$

where $\mathbf{E}_\lambda := \left(\frac{1}{\sigma^2}\boldsymbol{B}'\boldsymbol{B} + \lambda\boldsymbol{\Omega}\right)^{-1}$, $\mathbf{e}_\lambda' := \left(\frac{1}{\sigma^2}\boldsymbol{y}'\boldsymbol{B} + \lambda\boldsymbol{\beta}_0'\boldsymbol{\Omega}\right)\mathbf{E}_\lambda$ *and* c_λ^{prior} *is the probability of the truncation set* $\mathcal{M}$ *under* $\mathcal{N}(\boldsymbol{\beta}_0, \lambda^{-1}\boldsymbol{\Omega}^{-1})$ *and* c_λ^{post} *is its probability under* $\mathcal{N}(\mathbf{e}_\lambda, \mathbf{E}_\lambda)$.

Proof.
From the model specification we know that the likelihood function is given by

$$p(\boldsymbol{y}|\boldsymbol{\beta}, \lambda) \propto \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{B}\boldsymbol{\beta})'(\boldsymbol{y} - \boldsymbol{B}\boldsymbol{\beta})\right\}$$

and the prior density of $\boldsymbol{\beta}$ is

$$p(\boldsymbol{\beta}|\lambda) \propto \frac{|\lambda\boldsymbol{\Omega}|^{\frac{1}{2}}}{c_\lambda^{prior}} \ \exp\left\{-\frac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\beta}_0)'\lambda\boldsymbol{\Omega}(\boldsymbol{\beta} - \boldsymbol{\beta}_0)\right\} \mathbb{1}_{\mathcal{M}}(\boldsymbol{\beta}),$$

where c_λ^{prior} is the normalizing constant of the prior, that is, the probability of the truncation set $\mathcal{M}$ under the $\mathcal{N}(\boldsymbol{\beta}_0, \lambda^{-1}\boldsymbol{\Omega}^{-1})$-distribution.

Then we derive the following for the joint posterior of $\boldsymbol{\beta}$ and λ:

$$
\begin{aligned}
& p(\boldsymbol{\beta}, \lambda | \boldsymbol{y}) \\
= \;& \frac{p(\boldsymbol{\beta}, \lambda, \boldsymbol{y})}{p(\boldsymbol{y})} \\
\propto \;& p(\boldsymbol{\beta}, \lambda, \boldsymbol{y}) \\
= \;& p(\boldsymbol{y} | \boldsymbol{\beta}, \lambda) p(\boldsymbol{\beta} | \lambda) p(\lambda) \\
\propto \;& \exp\left\{-\tfrac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{B}\boldsymbol{\beta})'(\boldsymbol{y} - \boldsymbol{B}\boldsymbol{\beta})\right\} \frac{|\lambda\boldsymbol{\Omega}|^{\frac{1}{2}}}{c_\lambda^{prior}} \exp\left\{-\tfrac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\beta}_0)'\lambda\boldsymbol{\Omega}(\boldsymbol{\beta} - \boldsymbol{\beta}_0)\right\} \mathbb{1}_{\mathcal{M}}(\boldsymbol{\beta})\; p(\lambda) \\
= \;& \frac{p(\lambda)\; |\lambda\boldsymbol{\Omega}|^{\frac{1}{2}}}{c_\lambda^{prior}} \exp\left\{-\frac{1}{2}\left(\frac{1}{\sigma^2}(\boldsymbol{y} - \boldsymbol{B}\boldsymbol{\beta})'(\boldsymbol{y} - \boldsymbol{B}\boldsymbol{\beta}) + (\boldsymbol{\beta} - \boldsymbol{\beta}_0)'\lambda\boldsymbol{\Omega}(\boldsymbol{\beta} - \boldsymbol{\beta}_0)\right)\right\} \mathbb{1}_{\mathcal{M}}(\boldsymbol{\beta}) \\
= \;& \frac{p(\lambda)\; |\lambda\boldsymbol{\Omega}|^{\frac{1}{2}}}{c_\lambda^{prior}} \exp\left\{-\frac{1}{2}\left((\boldsymbol{\beta} - \mathbf{e}_\lambda)'\mathbf{E}_\lambda^{-1}(\boldsymbol{\beta} - \mathbf{e}_\lambda) - \mathbf{e}_\lambda'\mathbf{E}_\lambda^{-1}\mathbf{e}_\lambda + \boldsymbol{\beta}_0'\lambda\boldsymbol{\Omega}\boldsymbol{\beta}_0\right)\right\} \mathbb{1}_{\mathcal{M}}(\boldsymbol{\beta})
\end{aligned}
$$

The last equality holds with $\mathbf{E}_\lambda = \left(\frac{1}{\sigma^2}\boldsymbol{B}'\boldsymbol{B} + \lambda\boldsymbol{\Omega}\right)^{-1}$ and $\mathbf{e}_\lambda' = (\frac{1}{\sigma^2}\boldsymbol{y}'\boldsymbol{B} + \lambda\boldsymbol{\beta}_0'\boldsymbol{\Omega})\mathbf{E}_\lambda$. This can be seen as follows:

$$
\begin{aligned}
\frac{1}{\sigma^2}(\boldsymbol{y} \;-\; & \boldsymbol{B}\boldsymbol{\beta})'(\boldsymbol{y} - \boldsymbol{B}\boldsymbol{\beta}) + (\boldsymbol{\beta} - \boldsymbol{\beta}_0)'\lambda\boldsymbol{\Omega}(\boldsymbol{\beta} - \boldsymbol{\beta}_0) \\
&= \frac{1}{\sigma^2}\boldsymbol{y}'\boldsymbol{y} - \frac{2}{\sigma^2}\boldsymbol{y}'\boldsymbol{B}\boldsymbol{\beta} + \frac{1}{\sigma^2}\boldsymbol{\beta}'\boldsymbol{B}'\boldsymbol{B}\boldsymbol{\beta} + \boldsymbol{\beta}'\lambda\boldsymbol{\Omega}\boldsymbol{\beta} - 2\boldsymbol{\beta}_0'\lambda\boldsymbol{\Omega}\boldsymbol{\beta} + \boldsymbol{\beta}_0'\lambda\boldsymbol{\Omega}\boldsymbol{\beta}_0 \\
&\propto \boldsymbol{\beta}' \underbrace{\left(\frac{1}{\sigma^2}\boldsymbol{B}'\boldsymbol{B} + \lambda\boldsymbol{\Omega}\right)}_{=:\mathbf{E}_\lambda^{-1}} \boldsymbol{\beta} - 2\left(\frac{1}{\sigma^2}\boldsymbol{y}'\boldsymbol{B} + \lambda\boldsymbol{\beta}_0'\boldsymbol{\Omega}\right)\boldsymbol{\beta} + \boldsymbol{\beta}_0'\lambda\boldsymbol{\Omega}\boldsymbol{\beta}_0 \\
&= \boldsymbol{\beta}'\mathbf{E}_\lambda^{-1}\boldsymbol{\beta} - 2\underbrace{\left(\frac{1}{\sigma^2}\boldsymbol{y}'\boldsymbol{B} + \lambda\boldsymbol{\beta}_0'\boldsymbol{\Omega}\right)\mathbf{E}_\lambda}_{=:\mathbf{e}_\lambda'}\mathbf{E}_\lambda^{-1}\boldsymbol{\beta} + \boldsymbol{\beta}_0'\lambda\boldsymbol{\Omega}\boldsymbol{\beta}_0 \\
&= \boldsymbol{\beta}'\mathbf{E}_\lambda^{-1}\boldsymbol{\beta} - 2\mathbf{e}_\lambda'\mathbf{E}_\lambda^{-1}\boldsymbol{\beta} + \mathbf{e}_\lambda'\mathbf{E}_\lambda^{-1}\mathbf{e}_\lambda - \mathbf{e}_\lambda'\mathbf{E}_\lambda^{-1}\mathbf{e}_\lambda + \boldsymbol{\beta}_0'\lambda\boldsymbol{\Omega}\boldsymbol{\beta}_0 \\
&= (\boldsymbol{\beta} - \mathbf{e}_\lambda)'\mathbf{E}_\lambda^{-1}(\boldsymbol{\beta} - \mathbf{e}_\lambda) - \mathbf{e}_\lambda'\mathbf{E}_\lambda^{-1}\mathbf{e}_\lambda + \boldsymbol{\beta}_0'\lambda\boldsymbol{\Omega}\boldsymbol{\beta}_0
\end{aligned}
$$

Thus we have

$$
p(\boldsymbol{\beta}, \lambda | \boldsymbol{y}) \propto \frac{p(\lambda)\; |\lambda\boldsymbol{\Omega}|^{\frac{1}{2}}}{c_\lambda^{prior}} \exp\left(\tfrac{1}{2}\mathbf{e}_\lambda'\mathbf{E}_\lambda^{-1}\mathbf{e}_\lambda - \tfrac{1}{2}\boldsymbol{\beta}_0'\lambda\boldsymbol{\Omega}\boldsymbol{\beta}_0\right) \exp\left\{-\tfrac{1}{2}(\boldsymbol{\beta} - \mathbf{e}_\lambda)'\mathbf{E}_\lambda^{-1}(\boldsymbol{\beta} - \mathbf{e}_\lambda)\right\} \mathbb{1}_{\mathcal{M}}(\boldsymbol{\beta}).
$$

Now Lemma 2 follows by integration over $\boldsymbol{\beta}$:

$$
\begin{aligned}
&p(\lambda|\boldsymbol{y}) \\
&= \int p(\boldsymbol{\beta}, \lambda|\boldsymbol{y}) d\boldsymbol{\beta} \\
&\propto \int_{\mathcal{M}} \frac{p(\lambda)\ |\lambda\boldsymbol{\Omega}|^{\frac{1}{2}}}{c_\lambda^{prior}}\ \exp\left(\frac{1}{2}\mathbf{e}_\lambda'\mathbf{E}_\lambda^{-1}\mathbf{e}_\lambda - \frac{1}{2}\boldsymbol{\beta}_0'\lambda\boldsymbol{\Omega}\boldsymbol{\beta}_0\right) \exp\left\{-\frac{1}{2}(\boldsymbol{\beta}-\mathbf{e}_\lambda)'\mathbf{E}_\lambda^{-1}(\boldsymbol{\beta}-\mathbf{e}_\lambda)\right\} d\boldsymbol{\beta} \\
&= \frac{p(\lambda)\ |\lambda\boldsymbol{\Omega}|^{\frac{1}{2}}}{c_\lambda^{prior}}\ \exp\left(\frac{1}{2}\mathbf{e}_\lambda'\mathbf{E}_\lambda^{-1}\mathbf{e}_\lambda - \frac{1}{2}\boldsymbol{\beta}_0'\lambda\boldsymbol{\Omega}\boldsymbol{\beta}_0\right) \int_{\mathcal{M}} \exp\left\{-\frac{1}{2}(\boldsymbol{\beta}-\mathbf{e}_\lambda)'\mathbf{E}_\lambda^{-1}(\boldsymbol{\beta}-\mathbf{e}_\lambda)\right\} d\boldsymbol{\beta} \\
&\propto \frac{p(\lambda)\ |\lambda\boldsymbol{\Omega}|^{\frac{1}{2}}}{c_\lambda^{prior}}\ \exp\left(\frac{1}{2}\mathbf{e}_\lambda'\mathbf{E}_\lambda^{-1}\mathbf{e}_\lambda - \frac{1}{2}\boldsymbol{\beta}_0'\lambda\boldsymbol{\Omega}\boldsymbol{\beta}_0\right) \\
&\qquad\qquad \times |\mathbf{E}_\lambda|^{\frac{1}{2}} \underbrace{\int_{\mathcal{M}} (2\pi)^{-\frac{d}{2}} |\mathbf{E}_\lambda|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{\beta}-\mathbf{e}_\lambda)'\mathbf{E}_\lambda^{-1}(\boldsymbol{\beta}-\mathbf{e}_\lambda)\right\} d\boldsymbol{\beta}}_{:=c_\lambda^{post}} \\
&= \frac{p(\lambda)\ |\lambda\boldsymbol{\Omega}|^{\frac{1}{2}}}{c_\lambda^{prior}}\ \exp\left(\frac{1}{2}\mathbf{e}_\lambda'\mathbf{E}_\lambda^{-1}\mathbf{e}_\lambda - \frac{1}{2}\boldsymbol{\beta}_0'\lambda\boldsymbol{\Omega}\boldsymbol{\beta}_0\right) |\mathbf{E}_\lambda|^{\frac{1}{2}}\ c_\lambda^{post} \\
&= p(\lambda)\ |\lambda\boldsymbol{\Omega}|^{\frac{1}{2}}\ |\mathbf{E}_\lambda|^{\frac{1}{2}} \frac{c_\lambda^{post}}{c_\lambda^{prior}}\ \exp\left(\frac{1}{2}\mathbf{e}_\lambda'\mathbf{E}_\lambda^{-1}\mathbf{e}_\lambda - \frac{1}{2}\lambda\boldsymbol{\beta}_0'\boldsymbol{\Omega}\boldsymbol{\beta}_0\right),
\end{aligned}
$$

where c_λ^{post} is the probability of the truncation set $\mathcal{M}$ under the $\mathcal{N}(\mathbf{e}_\lambda, \mathbf{E}_\lambda)$-distribution. More explicit, c_λ^{post} is the normalizing constant of the posterior of $\boldsymbol{\beta}$ since

$$
\begin{aligned}
&p(\boldsymbol{\beta}|\lambda, \boldsymbol{y}) \\
&= \frac{p(\boldsymbol{\beta}, \lambda|\boldsymbol{y})}{p(\lambda|\boldsymbol{y})} \\
&\propto \frac{\frac{p(\lambda)\ |\lambda\boldsymbol{\Omega}|^{\frac{1}{2}}}{c_\lambda^{prior}}\ \exp\left(\frac{1}{2}\mathbf{e}_\lambda'\mathbf{E}_\lambda^{-1}\mathbf{e}_\lambda - \frac{1}{2}\boldsymbol{\beta}_0'\lambda\boldsymbol{\Omega}\boldsymbol{\beta}_0\right) \exp\left\{-\frac{1}{2}(\boldsymbol{\beta}-\mathbf{e}_\lambda)'\mathbf{E}_\lambda^{-1}(\boldsymbol{\beta}-\mathbf{e}_\lambda)\right\} \mathbb{1}_{\mathcal{M}}(\boldsymbol{\beta})}{p(\lambda)\ |\lambda\boldsymbol{\Omega}|^{\frac{1}{2}}\ |\mathbf{E}_\lambda|^{\frac{1}{2}} \frac{c_\lambda^{post}}{c_\lambda^{prior}}\ \exp\left(\frac{1}{2}\mathbf{e}_\lambda'\mathbf{E}_\lambda^{-1}\mathbf{e}_\lambda - \frac{1}{2}\lambda\boldsymbol{\beta}_0'\boldsymbol{\Omega}\boldsymbol{\beta}_0\right)} \\
&\propto \frac{|\mathbf{E}_\lambda|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{\beta}-\mathbf{e}_\lambda)'\mathbf{E}_\lambda^{-1}(\boldsymbol{\beta}-\mathbf{e}_\lambda)\right\} \mathbb{1}_{\mathcal{M}}(\boldsymbol{\beta})}{c_\lambda^{post}},
\end{aligned}
$$

which is proportional to the density of the $\mathcal{N}(\mathbf{e}_\lambda, \mathbf{E}_\lambda)$-distribution truncated onto $\mathcal{M}$. □

Recall

Lemma 3. *Assume model (3.6), then*

(i) the marginal posterior density of $\boldsymbol{\beta}$ is given by

$$p(\boldsymbol{\beta}|\lambda, m, \boldsymbol{y}) \propto |\mathbf{E}_\lambda|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{\beta}-\mathbf{e}_\lambda)'\mathbf{E}_\lambda^{-1}(\boldsymbol{\beta}-\mathbf{e}_\lambda)\right\} \mathbb{1}_{\mathcal{S}_m}(\boldsymbol{\beta}),$$

that is

$$\boldsymbol{\beta}|\lambda, m, \boldsymbol{y} \sim \mathcal{N}_{\mathcal{S}_m}(\mathbf{e}_\lambda, \mathbf{E}_\lambda),$$

(ii) the marginal posterior density of $(\lambda, m)'$ is given by

$$p(\lambda, m|\boldsymbol{y}) \propto p(\lambda, m)\ |\tilde{\boldsymbol{\Omega}}_\lambda|^{\frac{1}{2}}\ |\mathbf{E}_\lambda|^{\frac{1}{2}}\ \frac{c_\lambda^{post}}{c_\lambda^{prior}} \exp\left(\frac{1}{2}\mathbf{e}_\lambda'\mathbf{E}_\lambda^{-1}\mathbf{e}_\lambda - \frac{1}{2}\boldsymbol{\beta}_0'\tilde{\boldsymbol{\Omega}}_\lambda\boldsymbol{\beta}_0\right),$$

where $\mathbf{E}_\lambda := \left(\frac{1}{\sigma^2}\boldsymbol{B}'\boldsymbol{B} + \tilde{\boldsymbol{\Omega}}_\lambda\right)^{-1}$, $\mathbf{e}_\lambda' := \left(\frac{1}{\sigma^2}\boldsymbol{y}'\boldsymbol{B} + \boldsymbol{\beta}_0'\tilde{\boldsymbol{\Omega}}_\lambda\right)\mathbf{E}_\lambda$ *and* c_λ^{prior} *is the probability of the truncation set* $\mathcal{S}_m$ *under* $\mathcal{N}(\boldsymbol{\beta}_0, \tilde{\boldsymbol{\Omega}}_\lambda^{-1})$ *and* c_λ^{post} *is its probability under* $\mathcal{N}(\mathbf{e}_\lambda, \mathbf{E}_\lambda)$.

Proof.
In model (3.6) the likelihood and prior density for $\boldsymbol{\beta}$ have the following form:

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{\beta}, \lambda, m) &\propto \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{y}-\boldsymbol{B}\boldsymbol{\beta})'(\boldsymbol{y}-\boldsymbol{B}\boldsymbol{\beta})\right\} \\ p(\boldsymbol{\beta}|\lambda, m) &\propto \frac{|\tilde{\boldsymbol{\Omega}}_\lambda|^{\frac{1}{2}}}{c_\lambda^{prior}} \exp\left\{-\frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\beta}_0)'\tilde{\boldsymbol{\Omega}}_\lambda(\boldsymbol{\beta}-\boldsymbol{\beta}_0)\right\} \mathbb{1}_{\mathcal{S}_m}(\boldsymbol{\beta}), \end{aligned}$$

where c_λ^{prior} is the normalizing constant of the prior, that is, the probability of the truncation set $\mathcal{S}_m$ under the $\mathcal{N}(\boldsymbol{\beta}_0, \tilde{\boldsymbol{\Omega}}_\lambda^{-1})$-distribution.
Then we derive the following for the joint posterior of $\boldsymbol{\beta}$ and λ and m:

$$\begin{aligned} & p(\boldsymbol{\beta}, \lambda, m|\boldsymbol{y}) \\ = \ & \frac{p(\boldsymbol{\beta}, \lambda, m, \boldsymbol{y})}{p(\boldsymbol{y})} \\ \propto \ & p(\boldsymbol{\beta}, \lambda, m, \boldsymbol{y}) \\ = \ & p(\boldsymbol{y}|\boldsymbol{\beta}, \lambda, m)p(\boldsymbol{\beta}|\lambda, m)p(\lambda, m) \\ \propto \ & \exp\left\{-\tfrac{1}{2\sigma^2}(\boldsymbol{y}-\boldsymbol{B}\boldsymbol{\beta})'(\boldsymbol{y}-\boldsymbol{B}\boldsymbol{\beta})\right\} \frac{|\tilde{\boldsymbol{\Omega}}_\lambda|^{\frac{1}{2}}}{c_\lambda^{prior}} \exp\left\{-\tfrac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\beta}_0)'\tilde{\boldsymbol{\Omega}}_\lambda(\boldsymbol{\beta}-\boldsymbol{\beta}_0)\right\} \mathbb{1}_{\mathcal{S}_m}(\boldsymbol{\beta})\ p(\lambda, m) \\ = \ & \frac{p(\lambda, m)\ |\tilde{\boldsymbol{\Omega}}_\lambda|^{\frac{1}{2}}}{c_\lambda^{prior}} \exp\left\{-\frac{1}{2}\left(\frac{1}{\sigma^2}(\boldsymbol{y}-\boldsymbol{B}\boldsymbol{\beta})'(\boldsymbol{y}-\boldsymbol{B}\boldsymbol{\beta}) + (\boldsymbol{\beta}-\boldsymbol{\beta}_0)'\tilde{\boldsymbol{\Omega}}_\lambda(\boldsymbol{\beta}-\boldsymbol{\beta}_0)\right)\right\} \mathbb{1}_{\mathcal{S}_m}(\boldsymbol{\beta}) \\ = \ & \frac{p(\lambda, m)\ |\tilde{\boldsymbol{\Omega}}_\lambda|^{\frac{1}{2}}}{c_\lambda^{prior}} \exp\left\{-\frac{1}{2}\left((\boldsymbol{\beta}-\mathbf{e}_\lambda)'\mathbf{E}_\lambda^{-1}(\boldsymbol{\beta}-\mathbf{e}_\lambda) - \mathbf{e}_\lambda'\mathbf{E}_\lambda^{-1}\mathbf{e}_\lambda + \boldsymbol{\beta}_0'\tilde{\boldsymbol{\Omega}}_\lambda\boldsymbol{\beta}_0\right)\right\} \mathbb{1}_{\mathcal{S}_m}(\boldsymbol{\beta}) \end{aligned}$$

The last equality holds with $\mathbf{E}_\lambda := \left(\frac{1}{\sigma^2}\boldsymbol{B}'\boldsymbol{B} + \tilde{\boldsymbol{\Omega}}_\lambda\right)^{-1}$ and $\mathbf{e}'_\lambda := (\frac{1}{\sigma^2}\boldsymbol{y}'\boldsymbol{B} + \boldsymbol{\beta}'_0\tilde{\boldsymbol{\Omega}}_\lambda)\mathbf{E}_\lambda$ analogous to the proof of Lemma 2.
Thus we have

$$
\begin{aligned}
& p(\boldsymbol{\beta}, \lambda, m|\boldsymbol{y}) \\
\propto\ & \frac{p(\lambda,m)\ |\tilde{\boldsymbol{\Omega}}_\lambda|^{\frac{1}{2}}}{c_\lambda^{prior}}\ \exp\left(\tfrac{1}{2}\mathbf{e}'_\lambda\mathbf{E}_\lambda^{-1}\mathbf{e}_\lambda - \tfrac{1}{2}\boldsymbol{\beta}'_0\tilde{\boldsymbol{\Omega}}_\lambda\boldsymbol{\beta}_0\right)\exp\left\{-\tfrac{1}{2}(\boldsymbol{\beta}-\mathbf{e}_\lambda)'\mathbf{E}_\lambda^{-1}(\boldsymbol{\beta}-\mathbf{e}_\lambda)\right\}\mathbb{1}_{\mathcal{S}_m}(\boldsymbol{\beta}).
\end{aligned}
$$

Now (ii) of Lemma 3 follows by integration over $\boldsymbol{\beta}$:

$$
\begin{aligned}
& p(\lambda, m|\boldsymbol{y}) = \int p(\boldsymbol{\beta}, \lambda, m|\boldsymbol{y})d\boldsymbol{\beta} \\
& \propto \int_{\mathcal{S}_m} \frac{p(\lambda,m)\ |\tilde{\boldsymbol{\Omega}}_\lambda|^{\frac{1}{2}}}{c_\lambda^{prior}}\ \exp\left(\frac{1}{2}\mathbf{e}'_\lambda\mathbf{E}_\lambda^{-1}\mathbf{e}_\lambda - \frac{1}{2}\boldsymbol{\beta}'_0\tilde{\boldsymbol{\Omega}}_\lambda\boldsymbol{\beta}_0\right)\exp\left\{-\frac{1}{2}(\boldsymbol{\beta}-\mathbf{e}_\lambda)'\mathbf{E}_\lambda^{-1}(\boldsymbol{\beta}-\mathbf{e}_\lambda)\right\}d\boldsymbol{\beta} \\
& = \frac{p(\lambda,m)\ |\tilde{\boldsymbol{\Omega}}_\lambda|^{\frac{1}{2}}}{c_\lambda^{prior}}\ \exp\left(\frac{1}{2}\mathbf{e}'_\lambda\mathbf{E}_\lambda^{-1}\mathbf{e}_\lambda - \frac{1}{2}\boldsymbol{\beta}'_0\tilde{\boldsymbol{\Omega}}_\lambda\boldsymbol{\beta}_0\right)\int_{\mathcal{S}_m}\exp\left\{-\frac{1}{2}(\boldsymbol{\beta}-\mathbf{e}_\lambda)'\mathbf{E}_\lambda^{-1}(\boldsymbol{\beta}-\mathbf{e}_\lambda)\right\}d\boldsymbol{\beta} \\
& \propto \frac{p(\lambda,m)\ |\tilde{\boldsymbol{\Omega}}_\lambda|^{\frac{1}{2}}}{c_\lambda^{prior}}\ \exp\left(\frac{1}{2}\mathbf{e}'_\lambda\mathbf{E}_\lambda^{-1}\mathbf{e}_\lambda - \frac{1}{2}\boldsymbol{\beta}'_0\tilde{\boldsymbol{\Omega}}_\lambda\boldsymbol{\beta}_0\right) \\
& \qquad\qquad \times |\mathbf{E}_\lambda|^{\frac{1}{2}}\int_{\mathcal{S}_m}(2\pi)^{-\frac{d}{2}}|\mathbf{E}_\lambda|^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}(\boldsymbol{\beta}-\mathbf{e}_\lambda)'\mathbf{E}_\lambda^{-1}(\boldsymbol{\beta}-\mathbf{e}_\lambda)\right\}d\boldsymbol{\beta} \\
& = \frac{p(\lambda,m)\ |\tilde{\boldsymbol{\Omega}}_\lambda|^{\frac{1}{2}}}{c_\lambda^{prior}}\ \exp\left(\frac{1}{2}\mathbf{e}'_\lambda\mathbf{E}_\lambda^{-1}\mathbf{e}_\lambda - \frac{1}{2}\boldsymbol{\beta}'_0\tilde{\boldsymbol{\Omega}}_\lambda\boldsymbol{\beta}_0\right)|\mathbf{E}_\lambda|^{\frac{1}{2}}\ c_\lambda^{post} \\
& = p(\lambda,m)\ |\tilde{\boldsymbol{\Omega}}_\lambda|^{\frac{1}{2}}\ |\mathbf{E}_\lambda|^{\frac{1}{2}}\frac{c_\lambda^{post}}{c_\lambda^{prior}}\ \exp\left(\frac{1}{2}\mathbf{e}'_\lambda\mathbf{E}_\lambda^{-1}\mathbf{e}_\lambda - \frac{1}{2}\boldsymbol{\beta}'_0\tilde{\boldsymbol{\Omega}}_\lambda\boldsymbol{\beta}_0\right) =: w_2(\lambda, m),
\end{aligned}
$$

where c_λ^{post} is the probability of the truncation set $\mathcal{S}_m$ under the $\mathcal{N}(\mathbf{e}_\lambda, \mathbf{E}_\lambda)$-distribution.

Furthermore, (i) of Lemma 3 can be seen as follows:

$$
\begin{aligned}
& p(\boldsymbol{\beta}|\lambda, m, \boldsymbol{y}) = \frac{p(\boldsymbol{\beta}, \lambda, m|\boldsymbol{y})}{p(\lambda, m|\boldsymbol{y})} \\
\propto\ & \frac{\frac{p(\lambda,m)\ |\tilde{\boldsymbol{\Omega}}_\lambda|^{\frac{1}{2}}}{c_\lambda^{prior}}\ \exp\left(\frac{1}{2}\mathbf{e}'_\lambda\mathbf{E}_\lambda^{-1}\mathbf{e}_\lambda - \frac{1}{2}\boldsymbol{\beta}'_0\tilde{\boldsymbol{\Omega}}_\lambda\boldsymbol{\beta}_0\right)\exp\left\{-\frac{1}{2}(\boldsymbol{\beta}-\mathbf{e}_\lambda)'\mathbf{E}_\lambda^{-1}(\boldsymbol{\beta}-\mathbf{e}_\lambda)\right\}\mathbb{1}_{\mathcal{S}_m}(\boldsymbol{\beta})}{p(\lambda,m)\ |\tilde{\boldsymbol{\Omega}}_\lambda|^{\frac{1}{2}}\ |\mathbf{E}_\lambda|^{\frac{1}{2}}\frac{c_\lambda^{post}}{c_\lambda^{prior}}\ \exp\left(\frac{1}{2}\mathbf{e}'_\lambda\mathbf{E}_\lambda^{-1}\mathbf{e}_\lambda - \frac{1}{2}\boldsymbol{\beta}'_0\tilde{\boldsymbol{\Omega}}_\lambda\boldsymbol{\beta}_0\right)} \\
\propto\ & \frac{|\mathbf{E}_\lambda|^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}(\boldsymbol{\beta}-\mathbf{e}_\lambda)'\mathbf{E}_\lambda^{-1}(\boldsymbol{\beta}-\mathbf{e}_\lambda)\right\}\mathbb{1}_{\mathcal{S}_m}(\boldsymbol{\beta})}{c_\lambda^{post}}
\end{aligned}
$$

which is proportional to the density of the $\mathcal{N}(\mathbf{e}_\lambda, \mathbf{E}_\lambda)$-distribution truncated onto $\mathcal{S}_m$ and thus $\boldsymbol{\beta}|\lambda, m, \boldsymbol{y} \sim \mathcal{N}_{\mathcal{S}_m}(\mathbf{e}_\lambda, \mathbf{E}_\lambda)$. □

C The inverse Bayes formulae sampler for truncated multivariate normal random sampling

Yu and Tian (2011) proposed two novel approaches for sampling from a multivariate normal distribution which is truncated onto a set defined by linear inequality constraints. Here, we present the inverse Bayes formulae sampler, which is used in the Monte Carlo random sampling scheme for the Bayesian unimodal regression described in Section 3.6.

Suppose first, that a sample from the d-dimensional distribution $\mathcal{N}_{\mathcal{M}_1}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is desired, where the truncation set is defined by $\mathcal{M}_1 = \{\boldsymbol{r} \in \mathbb{R}^d | \boldsymbol{l} \leq \boldsymbol{r} \leq \boldsymbol{u}\}$ with $\boldsymbol{l}$ and $\boldsymbol{u} \in \mathbb{R}^d$. The procedure starts with finding an approximation $\tilde{\boldsymbol{r}}$ of the mode of this truncated distribution by an EM algorithm. Let $\boldsymbol{\Gamma} = (\gamma_{ik})_{i,k=1,\ldots,d}$ be the upper triangular matrix of the Cholesky decomposition such that $\boldsymbol{\Sigma} = \boldsymbol{\Gamma}'\boldsymbol{\Gamma}$ and $\boldsymbol{v} := \boldsymbol{\Gamma}\boldsymbol{\mu}$. According to Yu and Tian (2011) the steps are given as follows:

E-step: $\vartheta_k^{(t)} = \dfrac{1}{\sum_{i=1}^d \gamma_{ik}^2} \sum_{i=1}^d \gamma_{ik} \left[\gamma_{ik} \tilde{r}_k^{(t)} + \dfrac{v_i - \sum_{\ell=1}^d \gamma_{i\ell} \tilde{r}_\ell^{(t)}}{d} \right], \quad k = 1, \ldots, d.$

M-step: $\tilde{r}_k^{(t+1)} = \min\{\max\{l_k, \vartheta_k^{(t)}\}, u_k\}, , \quad k = 1, \ldots, d.$

Yu and Tian (2011) do not address the question on how to choose a starting point $\tilde{\boldsymbol{r}}^{(0)} \in \mathcal{M}_1$, though it will surely influence the number of iterations necessary to obtain a good approximation of the mode. We address this issue for the special case of sampling unimodal spline coefficients when describing the implementation of the Bayesian unimodal regression in Section 5.2.

The approximated mode $\tilde{\boldsymbol{r}}$ can now be used in the sampling scheme, which is based on an augmented data setting with $d(d-1)$ independent latent variables $\boldsymbol{\mathcal{Z}} = \{Z_{ik},\ i =$

$1, \ldots, d,\ k = 1, \ldots, d-1\}$ and observed data $\boldsymbol{v}$ from distribution $\mathcal{N}_{\mathcal{M}_1}(\boldsymbol{\Gamma r}, \boldsymbol{I}_d)$. By defining $Z_{id} := v_i - \sum_{k=1}^{d-1} Z_{ik}$, the complete data is given by $\{Z_{ik},\ i = 1, \ldots, d,\ k = 1, \ldots, d\}$. Yu and Tian (2011) assume the following distributions,

$$Z_{ik} \underset{\text{ind.}}{\sim} \mathcal{N}\left(\gamma_{ik} r_k, \frac{1}{d}\right),\ i = 1, \ldots, d,\ k = 1, \ldots, d,\ l_k \leq r_k \leq u_k,$$

and they show that the $d(d-1)$-dimensional conditional predictive distribution $f_{\boldsymbol{\mathcal{Z}}}(\boldsymbol{z}|\boldsymbol{v}, \boldsymbol{r})$ of the random vector $\boldsymbol{\mathcal{Z}}$ is given by a product of d multivariate normal distributions of dimension $d-1$:

$$\prod_{i=1}^{d} \mathcal{N}(E(\boldsymbol{\mathcal{Z}}_i|\boldsymbol{v}, \boldsymbol{r}), \boldsymbol{F}_d),$$

where $\boldsymbol{\mathcal{Z}}_i = (Z_{i1}, \ldots, Z_{i,d-1})'$, $E(\boldsymbol{\mathcal{Z}}_i|\boldsymbol{v}, \boldsymbol{r}) = (\Gamma_{i1} r_1, \ldots, \Gamma_{i,d-1} r_{d-1})' + \mathbf{1}_{d-1} \frac{v_i - \sum_{k=1}^{d} \gamma_{ik} r_k}{d}$, $i = 1, \ldots, d$, and $\boldsymbol{F}_d = \frac{1}{d}(\boldsymbol{I}_{d-1} - \frac{1}{d}\mathbf{1}_{d-1}\mathbf{1}'_{d-1})$.
Yu and Tian (2011) state that using a diffuse prior on $\boldsymbol{r}$, the complete-data posterior $f(\boldsymbol{r}|\boldsymbol{v}, \boldsymbol{z})$ is given by the product of d univariate truncated normal distributions:

$$\prod_{k=1}^{d} \mathcal{N}_{\{r_k \in \mathbb{R} | l_k \leq r_k \leq u_k\}} \left(\frac{\sum_{i=1}^{d} \gamma_{ik} z_{ik}}{\sum_{i=1}^{d} \gamma_{ik}^2}, \frac{1}{n \sum_{i=1}^{d} \gamma_{ik}^2} \right).$$

By defining $\boldsymbol{z}_0 := E(\boldsymbol{\mathcal{Z}}|\boldsymbol{v}, \tilde{\boldsymbol{r}})$ and using the observed posterior density $f(\boldsymbol{r}|\boldsymbol{v}) \propto \frac{f(\boldsymbol{r}|\boldsymbol{v}, \boldsymbol{z}_0)}{f_{\boldsymbol{\mathcal{Z}}}(\boldsymbol{z}_0|\boldsymbol{v}, \boldsymbol{r})}$, a sample of size N approximately from $\mathcal{N}_{\mathcal{M}_1}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ can be obtained by the following sampling/importance resampling procedure:

- Draw $J > N$ samples $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_J$ from $f(\boldsymbol{r}|\boldsymbol{v}, \boldsymbol{z}_0)$.
- Calculate weights $w_j = \frac{f_{\boldsymbol{\mathcal{Z}}}^{-1}(\boldsymbol{z}_0|\boldsymbol{v}, \boldsymbol{r}_j)}{\sum_{l=1}^{J} f_{\boldsymbol{\mathcal{Z}}}^{-1}(\boldsymbol{z}_0|\boldsymbol{v}, \boldsymbol{r}_l)},\ j = 1, \ldots, J.$
- Resample a subset of size N without replacement according to the weights.

The higher J, the better the approximation (cf. Yu and Tian, 2011).

Suppose now that a sample from the d-dimensional distribution $\mathcal{N}_{\mathcal{M}_2}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is desired, where the truncation set $\mathcal{M}_2$ is defined by $d^* \leq d$ linear inequality constraints, that is, $\mathcal{M}_2 = \{\boldsymbol{r} \in \mathbb{R}^d | \boldsymbol{l} \leq \boldsymbol{C}\boldsymbol{r} \leq \boldsymbol{u}\}$ with $\boldsymbol{l}, \boldsymbol{u} \in \mathbb{R}^{d^*}$, $\boldsymbol{C} \in \mathbb{R}^{d^* \times d}$. This is exactly the situation in Section 3.6, where a random B-spline coefficient vector has to be drawn from the multivariate normal marginal posterior $\mathcal{N}_{\mathcal{S}_m}(\mathbf{e}_\lambda, \mathbf{E}_\lambda)$. The set $\mathcal{S}_m$ of unimodal vectors with mode m is obtained with $\boldsymbol{l} = -\infty \cdot \mathbf{1}_{d-1}$, $\boldsymbol{u} = \mathbf{0}_{d-1}$ and $\boldsymbol{C} = -\boldsymbol{C}_m \in \mathbb{R}^{(d-1) \times d}$ as defined in Equation (3.4).

The key idea for sampling from distributions truncated in form of $\mathcal{M}_2$ is to draw a random sample $\{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_N\}$ from $\mathcal{N}_{\mathcal{M}_1}(\boldsymbol{C\mu}, \boldsymbol{C\Sigma C}')$, where $\mathcal{M}_1 = \{\boldsymbol{r} \in \mathbb{R}^d | \boldsymbol{l} \leq \boldsymbol{r} \leq \boldsymbol{u}\}$ as above. This can be done with the procedure described in the preceding paragraph. If $d^* = d$ and $\boldsymbol{C}$ is invertible, the transformations $\boldsymbol{C}^{-1}\boldsymbol{r}_1, \ldots, \boldsymbol{C}^{-1}\boldsymbol{r}_N$ will form a random sample from the desired distribution $\mathcal{N}_{\mathcal{M}_2}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. If $d^* < d$, it is possible to find $\boldsymbol{l}^* \in \mathbb{R}^d, \boldsymbol{u}^* \in \mathbb{R}^d, \boldsymbol{C}^* \in \mathbb{R}^{d \times d}$ such that $\mathcal{M}_2 = \{\boldsymbol{r} \in \mathbb{R}^d | \boldsymbol{l} \leq \boldsymbol{Cr} \leq \boldsymbol{u}\} = \{\boldsymbol{r} \in \mathbb{R}^d | \boldsymbol{l}^* \leq \boldsymbol{C}^* \boldsymbol{r} \leq \boldsymbol{u}^*\}$, for example, by adding $d - d^*$ rows to matrix $\boldsymbol{C}^*$ that "constrain" the last $d - d^*$ components between $-\infty$ and ∞. This ensures that $\boldsymbol{C}^*$ is invertible. Thus, $\mathcal{M}_2$ can be written in the manner of $\mathcal{M}_1$ and random sampling as described is possible (cf. Yu and Tian, 2011).

The case of $d^* > d$ is also treated in Yu and Tian (2011), but is not presented here, since this case does not appear for the shape constraints examined in this thesis.

For the matrix $\boldsymbol{C}_m$ of the unimodal shape constraint we have $d^* = d - 1$ and we can choose $\boldsymbol{l}^* = -\infty \cdot \boldsymbol{1}_d$, $\boldsymbol{u}^* = \begin{pmatrix} \boldsymbol{0}_{d-1} \\ \infty \end{pmatrix}$ and $\boldsymbol{C}^* = \begin{pmatrix} \boldsymbol{C}_m \\ \boldsymbol{0}_{d-1}\ 1 \end{pmatrix}$.

D Documentations of auxiliary R functions

The functions are listed in alphabetical order.

Function `equiknots`

`equiknots` *Determine the knot sequence.*

Description

Determines $g + 2k + 2$ knots for a spline basis of degree k on the interval $[a, b]$.

Usage

```
equiknots(a, b, g, k, coinc)
```

Arguments

`a`	The left numeric boundary of the interval $[a, b]$.
`b`	The numeric right boundary of the interval $[a, b]$.
`g`	A non-negative integer giving the number of inner knots.
`k`	A non-negative integer specifying the degree of the spline basis.
`coinc`	Logical indicating, if the outer knots should be coincident with the boundary knots or not. If `coinc=TRUE`, there are k coincident outer knots at a as well as at b.

Details

The function `equiknots` determines a knot sequence, where the g inner knots lie equidistant in $[a, b]$. If `coinc=TRUE`, the outer knots (k on each side of the interval)

are placed coincident with a and b, otherwise the outer knots are also equidistant beyond $[a, b]$

Value

A numeric vector of length $g + 2k + 2$ with knot locations.

Examples

```
equiknots(0,5,3,3,TRUE)
# [1] 0.00 0.00 0.00 0.00 1.25 2.50 3.75 5.00 5.00 5.00 5.00
equiknots(0,5,3,3,FALSE)
# [1] -3.75 -2.50 -1.25 0.00 1.25 2.50 3.75 5.00 6.25 7.50 8.75
```

Function `negloglik`

`negloglikBayes` *Evaluate the negative marginal log-posterior of* λ.

Description

This function implements the negative of the marginal log-posterior of the tuning parameter λ in the Bayesian unimodal spline regression model.

Usage

```
negloglikBayes(lambda, prior, tBSB, tySB, sigmaest, tbeta0,
tbeta0beta, tbetaDtildebeta, tbetaDtilde, Dtilde, beta0, Om, Cm, m)
```

Arguments

`lambda`	Positive tuning parameter value at which to evaluate the negative marginal log-posterior.
`prior`	A function returning the value of the prior density $p(\lambda, m)$.
`tBSB`	The value of $\boldsymbol{B}' diag(\boldsymbol{w}) \boldsymbol{B}$ (passed over from function `unibayes`, cf. Section 5.2).
`tySB`	The value of $\boldsymbol{y}' diag(\boldsymbol{w}) \boldsymbol{B}$ (passed over from function `unibayes`).
`sigmaest`	The estimate of σ^2.

`tbeta0`	The value of $\boldsymbol{\beta}_0'\boldsymbol{\Omega}$ (passed over from function `unibayes`).
`tbeta0beta`	The value of $\boldsymbol{\beta}_0'\boldsymbol{\Omega}\boldsymbol{\beta}_0$ (passed over from function `unibayes`).
`tbetaDtildebeta`	The value of $\boldsymbol{\beta}_0'\tilde{\boldsymbol{D}}\boldsymbol{\beta}_0$ (passed over from function `unibayes`).
`tbetaDtilde`	The value of $\boldsymbol{\beta}_0'\tilde{\boldsymbol{D}}$ (passed over from function `unibayes`).
`Dtilde`	$\tilde{\boldsymbol{D}} \in \mathbb{R}^{d\times d} = \frac{1}{\sigma_v^2}\boldsymbol{I}_d$.
`beta0`	The penalty vector $\boldsymbol{\beta}_0$, a length d numeric vector.
`Om`	The penalty matrix $\boldsymbol{\Omega}$, a numeric $d \times d$ matrix.
`m`	The mode location.
`Cm`	Constraint matrix $\boldsymbol{C}_m$.

Details

The function `negloglikBayes` determines the negative of the marginal log-posterior of the tuning parameter up to an additive constant according to Lemma 3 (ii).
The implementation starts with the calculation of matrices $\tilde{\boldsymbol{\Omega}}_\lambda$, $\boldsymbol{\beta}_0'\tilde{\boldsymbol{\Omega}}_\lambda\boldsymbol{\beta}_0$, $\mathbf{E}_\lambda := \left(\frac{1}{\sigma^2}\boldsymbol{B}'\boldsymbol{B} + \tilde{\boldsymbol{\Omega}}_\lambda\right)^{-1}$ and vector $\mathbf{e}_\lambda' = \left(\frac{1}{\sigma^2}\boldsymbol{y}'\boldsymbol{B} + \boldsymbol{\beta}_0'\tilde{\boldsymbol{\Omega}}_\lambda\right)\mathbf{E}_\lambda$ from the arguments `tBSB`, `tySB`, `Dtilde`, `Om`, `tbetaDtildebeta`, `tbeta0beta`, `tbeta0` and `tbeta0tilde` and `lambda`.
The normalizing constants are computed as

$$c_\lambda^{prior} = P(\boldsymbol{\psi}_1 \geq \mathbf{0}) \text{ with } \boldsymbol{\psi}_1 = \boldsymbol{C}_m\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{C}_m\boldsymbol{\beta}_0, \boldsymbol{C}_m\tilde{\boldsymbol{\Omega}}_\lambda\boldsymbol{C}_m')$$

$$\text{and } c_\lambda^{post} = P(\boldsymbol{\psi}_2 \geq \mathbf{0}) \text{ with } \boldsymbol{\psi}_2 \sim \mathcal{N}(\boldsymbol{C}_m\mathbf{e}_\lambda, \boldsymbol{C}_m\mathbf{E}_\lambda\boldsymbol{C}_m').$$

Both probabilities can be computed with function `pmvnorm` from package `mvtnorm` (Genz et al., 2012).
The function value is then calculated as minus the logarithm of $w_2(\lambda, m)$:

$$-\log(p(\lambda,m)) - \frac{1}{2}\log(|\tilde{\boldsymbol{\Omega}}_\lambda|) + \frac{1}{2}\log(|\mathbf{E}_\lambda^{-1}|) - \log(c_\lambda^{post}) + \log(c_\lambda^{prior}) - \frac{1}{2}\mathbf{e}_\lambda'\mathbf{E}_\lambda^{-1}\mathbf{e}_\lambda + \frac{1}{2}\boldsymbol{\beta}_0'\tilde{\boldsymbol{\Omega}}_\lambda\boldsymbol{\beta}_0.$$

This is done using three sub-terms: $\text{term}_1 = \log(c_\lambda^{prior})$, $\text{term}_2 = -\log(c_\lambda^{post})$ and term_3 is the remainder. The first two are again combined to term_{12}. If $\text{term}_2 = \infty$, that is, if the posterior normalizing constant is zero, then term_{12} is set to ∞. If $\text{term}_1 = -\infty$, that is, if the prior normalizing constant is zero or close to zero ($\leq 5 \cdot 10^{-6}$), leading to division by zero in the likelihood, then term_{12} is set to ∞ as well, since zero prior probability will also result in zero posterior probability. In all other cases, term_{12} is just the sum of term_1 and term_2.
If $p(\lambda, m) = 0$, term_3 is directly set to ∞ without calculating the remaining sum-

mands.

To avoid numerical problems, the determinant $|\mathbf{E}_\lambda^{-1}|$ in term_3 is replaced by $|\hat{\sigma}\mathbf{E}_\lambda^{-1}|$, which only changes the value of $-\log(w_2(\lambda, m))$ by an additive constant.

Value

A list of

`term12`	The sum of term_1 and term_2.
`term3`	term_3 / the remaining summands of $w_2(\lambda, m)$.

Function `negloglikFREQ`

`negloglikFREQ` *Evaluate the negative restricted log-likelihood.*

Description

This function implements the negative of the restricted log-likelihood of the tuning parameter in a (shape-constrained) spline regression model.

Usage

```
negloglikFREQ(lambda, tBSB, tySB, sigmaest, tbetaV, tbetaVbeta,
Dtilde, beta0, Om, rangV, Cm, constr)
```

Arguments

`lambda`	Positive tuning parameter value at which to evaluate the negative restricted log-likelihood.
`tBSB`	The value of $\boldsymbol{B}' diag(\boldsymbol{w})\boldsymbol{B}$ (passed over from function `unireg`, cf. Section 5.1).
`tySB`	The value of $\boldsymbol{y}' diag(\boldsymbol{w})\boldsymbol{B}$ (passed over from `unireg`).
`sigmaest`	The estimate of σ^2 (passed over from `unireg`).
`tbetaV`	The value of $\boldsymbol{\beta}_0'\boldsymbol{\Omega}$ (passed over from `unireg`).
`tbetaVbeta`	The value of $\boldsymbol{\beta}_0'\boldsymbol{\Omega}\boldsymbol{\beta}_0$ (passed over from `unireg`).
`Dtilde`	$\tilde{\boldsymbol{D}} \in \mathbb{R}^{d\times d} = \frac{1}{\sigma_v^2}\boldsymbol{I}_d$ if $\text{rank}(\boldsymbol{\Omega}) < d$ and a zero matrix else (passed over from `unireg`).

`beta0`	The penalty vector $\boldsymbol{\beta}_0$, a length d numeric vector.
`Om`	The penalty matrix $\boldsymbol{\Omega}$, a numeric $d \times d$ matrix.
`rangV`	Rank r of $\boldsymbol{\Omega}$, integer between 1 and d.
`Cm`	Constraint matrix $\boldsymbol{C}_m$ corresponding to `constr`.
`constr`	A character string specifying the shape constraint for the fit. Can be one of "unimodal", "none" and "isotonic".

Details

The function `negloglikFREQ` is an internal function of R package `unireg` (Köllmann, 2016, see also Section 5.1). It determines the negative logarithm of the function $w_1(\cdot)$ from Equation (3.5):

$$-\log(p(\lambda))+\frac{r}{2}\log(\lambda)+\frac{1}{2}\log(|\mathbf{E}_\lambda^{-1}|)-\log(c_\lambda^{post})+\log(c_\lambda^{prior})-\frac{1}{2}\mathbf{e}_\lambda'\mathbf{E}_\lambda^{-1}\mathbf{e}_\lambda+\frac{1}{2}\lambda\boldsymbol{\beta}_0'\boldsymbol{\Omega}\boldsymbol{\beta}_0.$$

To avoid numerical problems the determinant $|\mathbf{E}_\lambda^{-1}|$ is replaced by $|\hat{\sigma}\mathbf{E}_\lambda^{-1}|$, which only changes the function value by an additive constant.
The function starts with calculation of matrix $\mathbf{E}_\lambda = (\frac{1}{\sigma^2}\boldsymbol{B}'\boldsymbol{B} + \lambda\boldsymbol{\Omega})^{-1}$ and vector $\mathbf{e}_\lambda = \left(\frac{1}{\sigma^2}\boldsymbol{y}'\boldsymbol{B} + \lambda\boldsymbol{\beta}_0'\boldsymbol{\Omega}\right)\mathbf{E}_\lambda$ from the arguments `tBSB`, `lambda`, `Om`, `tySB` and `tbetaV`. In case of a shape constraint, the normalizing constants c_λ^{prior} and $c_\lambda^{post}$are computed. For the former one, the full-rank matrix $\tilde{\boldsymbol{\Omega}}_\lambda$ has to be determined in case that $\boldsymbol{\Omega} = \boldsymbol{D}_q'\boldsymbol{D}_q$ for some q, that is, $\tilde{\boldsymbol{\Omega}}_\lambda = \tilde{\boldsymbol{D}} + \lambda\boldsymbol{D}_q'\boldsymbol{D}_q$. The constant c_λ^{prior} is then equal to $P(\boldsymbol{\beta} \in \mathcal{S}_m)$, $\mathcal{S}_m = \{\boldsymbol{\beta} \in \mathbb{R}^d | \boldsymbol{C}_m\boldsymbol{\beta} \geq \mathbf{0}\}$, where $\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\beta}_0, \tilde{\boldsymbol{\Omega}}_\lambda)$, or equivalently $P(\boldsymbol{\psi}_1 \geq \mathbf{0})$ with $\boldsymbol{\psi}_1 = \boldsymbol{C}_m\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{C}_m\boldsymbol{\beta}_0, \boldsymbol{C}_m\tilde{\boldsymbol{\Omega}}_\lambda\boldsymbol{C}_m')$. The slightly modified penalty matrix $\tilde{\boldsymbol{\Omega}}_\lambda$ is used only in this step. Analogously, $c_\lambda^{post} = P(\boldsymbol{\psi}_2 \geq \mathbf{0})$ with $\boldsymbol{\psi}_2 \sim \mathcal{N}(\boldsymbol{C}_m\mathbf{e}_\lambda, \boldsymbol{C}_m\mathbf{E}_\lambda\boldsymbol{C}_m')$. Both probabilities can be computed with the function `pmvnorm` from package `mvtnorm` (Genz et al., 2012). If there is no shape constraint (`constr="none"`), the normalizing constants are set to one.
This function can minimized to find the optimal tuning parameter value according to REML (which is done in function `unisplinem`, see next Section). For this purpose, it is necessary that `negloglikFREQ` returns a single numeric value. So if the resulting log-likelihood value is $\pm$`Inf`, it is set to the largest normalized floating-point number, `.Machine$double.xmax`, instead. All calculations are performed on $\log(\lambda)$ level so that optimization of the tuning parameter can be done more efficiently.

Function `points.unireg`

`points.unireg` *Points method for* ***`unireg`*** *objects.*

Description

Plotting a unimodal regression object into an existing plot.

Usage

```
## S3 method for class 'unireg'
points(x, type="l", ...)
```

Arguments

`x`	Object of class `"unireg"`, a result of function `unireg`.
`type`	Per default plotting type `"l"` is used for the fitted spline.
`...`	Other parameters to be passed through to the generic `points` function.

Details

This is a `points` method for unimodal regression objects, in explicit, objects of class `"unireg"` obtained as a result of function `unireg` (cf. Section 5.1). The spline function is plotted using a grid of x-values equally spaced across the interval on which the spline is defined. The distance between the grid values is given by $\frac{\min(\boldsymbol{x})}{10}$, where $\boldsymbol{x}$ is the observed predictor vector originally used for fitting.

Value

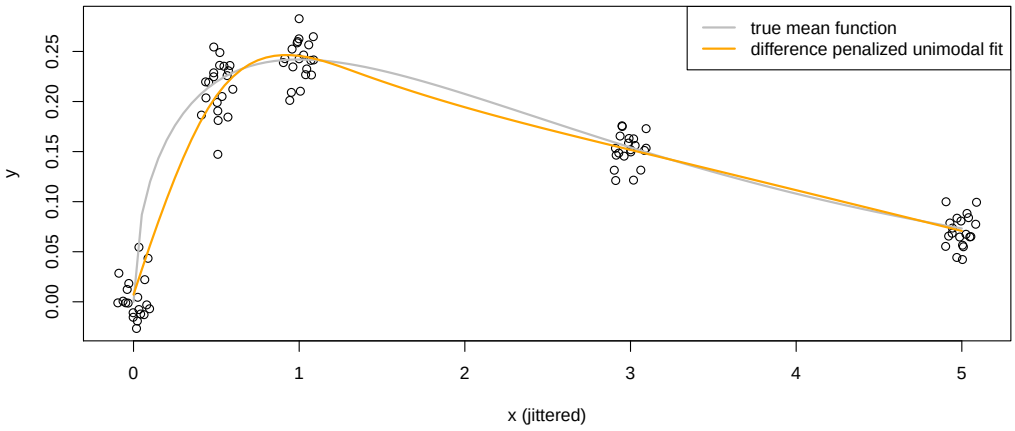
`NULL`

Examples

```
# generate some data
x <- sort(rep(c(0,0.5,1,3,5),20))
set.seed(41333)
func <- function(mu){rnorm(1,mu,0.02)}
y <- sapply(dchisq(x,3),func)
```

```
# fit with default settings
fit <- unireg(x, y)

# plot of true and fitted function
plot(jitter(x), y, xlab="x (jittered)")
curve(dchisq(x,3), 0, 5, type="l", col="grey", lwd=2, add=TRUE)
points(fit, lwd=2, col="orange")
legend("topright", legend = c("true mean function",
    "difference penalized unimodal fit"),
    col=c("grey","orange"),lwd=c(2,2))
```



Function `plot.unireg`

`plot.unireg` *Plot method for* ***`unireg`*** *objects.*

Description

Plotting a unimodal regression object.

Usage

```
## S3 method for class 'unireg'
plot(x, onlySpline=FALSE, type="l", xlab="x", ylab=NULL,
col="black", ...)
```

Arguments

`x`	Object of class `"unireg"`, a result of function `unireg`.
`onlySpline`	Logical indicating whether only the fitted spline or also the original data points should be plotted. Defaults to FALSE (plotting both).
`type`	Per default plotting type `"l"` is used for the fitted spline.
`xlab`	Per default the x-axis is labelled with "x".
`ylab`	If the user does not specify a label for the y-axis, that is, if `ylab=NULL` (default), pre-specified labels like "Fitted unimodal spline function" (depending on the constraint) are used.
`col`	Colour of the spline function to be plotted (default: black).
`...`	Other parameters to be passed through to the generic `plot` function.

Details

This is a `plot` method for unimodal regression objects, i.e., objects of class `"unireg"` obtained as a result of function `unireg` (cf. Section 5.1). The spline function is plotted using a grid of x-values spaced equally between in the interval on which the spline is defined. The distance between the grid values is given by $\frac{\min(\boldsymbol{x})}{10}$, where $\boldsymbol{x}$ is the observed predictor vector originally used for fitting.

Value

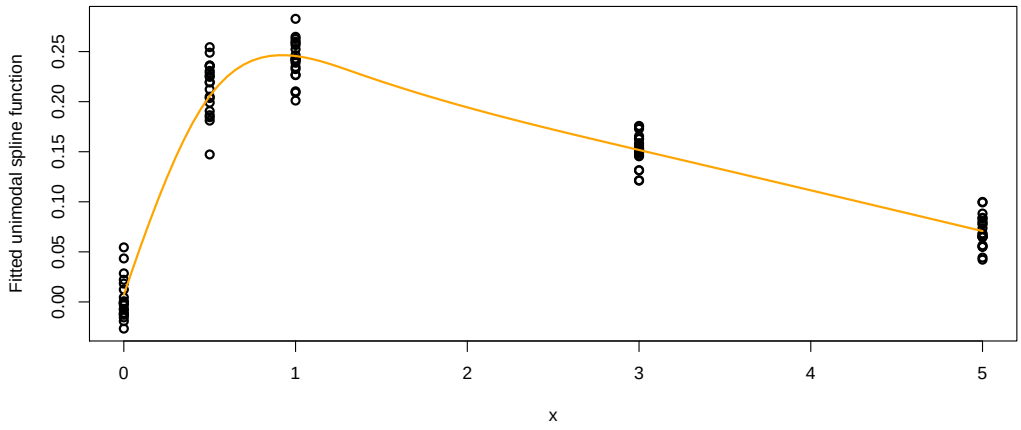
`NULL`

Examples

```
# generate some data
x <- sort(rep(c(0,0.5,1,3,5),20))
set.seed(41333)
func <- function(mu){rnorm(1,mu,0.02)}
y <- sapply(dchisq(x,3),func)

# fit with default settings
fit <- unireg(x, y)

# plot of fitted spline with data
plot(fit, col="orange")
```



Function `predict.unireg`

`predict.unireg` *Predict method for* ***`unireg`*** *objects.*

Description

Predict values based on a unimodal regression object.

Usage

```
## S3 method for class 'unireg'
predict(object, newdata, ...)
```

Arguments

`object`	Object of class `"unireg"`, a result of function `unireg`.
`newdata`	A numeric vector of values at which to evaluate the fitted spline function.
`...`	Further arguments (currently not used).

Details

This is a `predict` method for unimodal regression objects, i.e., objects of class `"unireg"` obtained as a result of function `unireg` (cf. Section 5.1).

`predict.unireg` produces predicted values by evaluating the fitted regression spline function at the values in `newdata`.

Value

A numeric vector of predicted function values.

Examples

```
# generate some data
x <- sort(rep(c(0,0.5,1,3,5),20))
set.seed(41333)
func <- function(mu){rnorm(1,mu,0.02)}
y <- sapply(dchisq(x,3),func)

# fit with default settings
fit <- unireg(x, y)

# prediction at interim values
predict(fit, c(1.5,2.5,3.5,4.5))
# [1]  0.22016487 0.17213029 0.13166340 0.09113394
```

Function `print.unireg`

`print.unireg` *Print method for* ***`unireg`*** *objects.*

Description

Prints unimodal regression objects.

Usage

```
## S3 method for class 'unireg'
print(x, ...)
```

Arguments

`x`	Object of class `"unireg"`, a result of function `unireg`.
`...`	Further arguments (currently not used).

Details

This is a `print` method for unimodal regression objects, i.e., objects of class `"unireg"` obtained as a result of function `unireg` (cf. Section 5.1).

`print.unireg` prints a short overview of a fitted unimodal regression object to the console, namely, the type of the fitted model (including degree of the spline and type of constraint and penalty), the coefficients and their mode location, the tuning parameter and the variance estimate.

Value

Invisibly returns the input `x`.

Examples

```
# generate some data
x <- sort(rep(c(0,0.5,1,3,5),20))
set.seed(41333)
func <- function(mu){rnorm(1,mu,0.02)}
y <- sapply(dchisq(x,3),func)
```

```
# fit with default settings
fit <- unireg(x, y)

# short overview of the fitted spline
fit
#Fitted unimodal spline of degree 3 with difference penalty of order 2
#
#Coefficients          -0.24 0.01 0.22 0.26 0.23 0.2 0.18 0.16 0.14 ...
#Mode of coefficients  4
#Tuning parameter      20.09
#Variance estimate     0
```

Function `sampleTMVNIBF`

`sampleTMVNIBF` *Sampling from a truncated multivariate normal distribution.*

Description

Sampling from a d-dimensional truncated multivariate normal distribution using inverse Bayes formulae (IBF). The truncation set is given by

$$\{\boldsymbol{r} \in \mathbb{R}^d | \boldsymbol{l} \leq \boldsymbol{C}\boldsymbol{r} \leq \boldsymbol{u}\}.$$

Usage

```
sampleTMVNIBF(N=1000, J=3000, mu=c(0,0,0), sigma=diag(1,3),
l=c(-10,-10,-10), u=c(10,10,10), C, r0=c(0,0,0))
```

Arguments

`N`	Positive integer specifying the number of samples to be drawn.
`J`	Positive integer ($\geq N$) giving the size of an initial sample, from which a subset of size N is resampled.
`mu`	Numeric mean vector (length d) of the multivariate normal distribution.

`sigma`	Numeric covariance matrix ($d \times d$) of the MVN distribution.
`l`	Numeric vector of lower truncation limits (length $d^* \leq d$).
`u`	Numeric vector of upper truncation limits (length $d^* \leq d$).
`C`	Numeric matrix of constraint coefficients of Dimension $d^* \times d$ ($d^* \leq d$).
`r0`	Numeric vector (length d) specifying a starting point for iterative estimation of the mode of the MVN distribution.

Details

This function samples from a d-dimensional truncated multivariate normal (TMVN) distribution using the inverse Bayes formulae sampler introduced by Yu and Tian (2011) (cf. Appendix C). The truncation set is given by d^* linear inequality constraints of the form

$$\{\boldsymbol{r} \in \mathbb{R}^d | \boldsymbol{l} \leq \boldsymbol{C}\boldsymbol{r} \leq \boldsymbol{u}\},$$

where $\boldsymbol{l}, \boldsymbol{u} \in R^{d^*}$, $\boldsymbol{C} \in \mathbb{R}^{d^* \times d}$. Choosing $\boldsymbol{C} = \boldsymbol{I}_d$ results in "typical" truncation of the form $\{\boldsymbol{r} \in \mathbb{R}^d | \boldsymbol{l} \leq \boldsymbol{r} \leq \boldsymbol{u}\}$.

If $d^* < d$, the truncation limits $\boldsymbol{l}, \boldsymbol{u}$ and the constraint matrix $\boldsymbol{C}$ are replaced with appropriately chosen $\boldsymbol{l}^* \in \mathbb{R}^d, \boldsymbol{u}^* \in \mathbb{R}^d, \boldsymbol{C}^* \in \mathbb{R}^{d \times d}$ (cf. Appendix C) prior to the actual sampling procedure. So the following description is only for $d^* = d$:

If $d = 1$, the implementation simply calls the function `rtruncnorm` from package `truncnorm` (Trautmann et al., 2014). For $d > 1$, the function samples N times from the TMVN $\mathcal{N}_{\{\boldsymbol{l} \leq \boldsymbol{r} \leq \boldsymbol{u}\}}(\boldsymbol{C}\mu, \boldsymbol{C}\Sigma\boldsymbol{C}')$ as described in Section C. The main steps are a Cholesky decomposition, estimation of the mode $\tilde{\boldsymbol{r}}$ with EM-algorithm and calculation of $\boldsymbol{\mathcal{Z}}_0$. A first sample of size J is drawn, again with the help of function `rtruncnorm` for one-dimensional sampling from a truncated normal. After calculating weights according to inverse Bayes formulae, a subset of size N is resampled without replacement, yielding a sample from $\mathcal{N}_{\{\boldsymbol{r} \in \mathbb{R}^d | \boldsymbol{l} \leq \boldsymbol{r} \leq \boldsymbol{u}\}}(\boldsymbol{C}\mu, \boldsymbol{C}\Sigma\boldsymbol{C}')$.

After back-transformation with the inverse $\boldsymbol{C}^{-1}$ (or $(\boldsymbol{C}^*)^{-1}$, respectively) the draws form a random sample from the originally desired $\mathcal{N}_{\{\boldsymbol{r} \in \mathbb{R}^d | \boldsymbol{l} \leq \boldsymbol{C}\boldsymbol{r} \leq \boldsymbol{u}\}}(\mu, \Sigma)$ distribution.

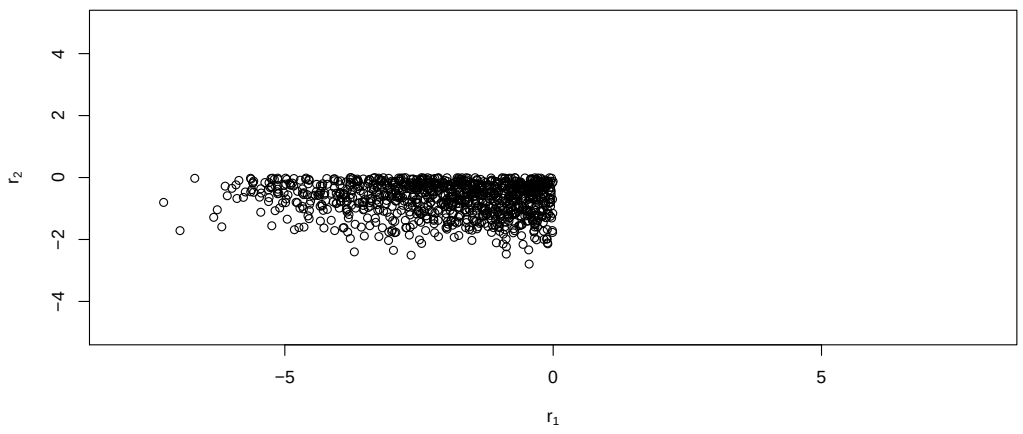
Value

A numeric matrix ($N \times d$) with N samples from the d-dimensional truncated multivariate normal distribution in its rows.

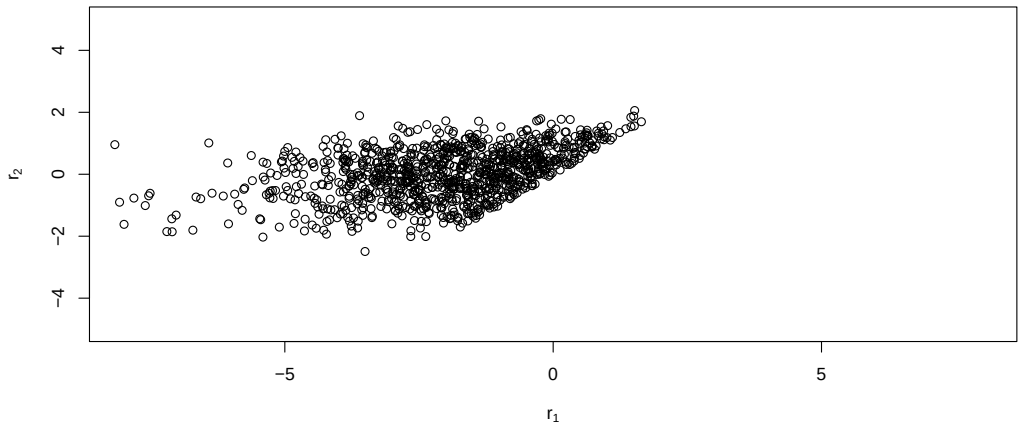
Examples

```
mu <- c(0, 0)
sigma <- matrix(c(10, 0, 0, 1), 2, 2)

# "typical" truncation
C <- diag(1, length(mu))
l <- rep(-Inf, length(mu))
u <- rep(0, length(mu))
R <- sampleTMVNIBF(N=1000, J=3000, mu, sigma, l, u, C, r0=c(0, 0))
plot(R, xlab=expression(r[1]), ylab=expression(r[2]), xlim=c(-8, 8),
    ylim=c(-5, 5))
```




```
# truncation with linear inequality constraints and dstar < d
C <- rbind(c(1, -1))
l <- -Inf
u <- 0
R <- sampleTMVNIBF(N=1000, J=3000, mu, sigma, l, u, C, r0=c(0, 0))
plot(R, xlab=expression(r[1]), ylab=expression(r[2]), xlim=c(-8, 8),
    ylim=c(-5, 5))
```



Function `unimat`

`unimat` *Create the matrix of unimodality constraints.*

Description

Returns a matrix $\boldsymbol{C}_m$ that can be used to specify linear constraints $\boldsymbol{C}_m\boldsymbol{b} \geq 0$ to impose unimodality with mode m on a numeric vector $\boldsymbol{b} \in \mathbb{R}^p$.

Usage

```
unimat(p, m)
```

Arguments

`p`	Integer (≥ 2) giving the length of the vector $\boldsymbol{b}$.
`m`	Location of the mode within the vector $\boldsymbol{b}$. Should be an integer between 1 and p.

Details

The function `unimat` determines the matrix $\boldsymbol{C}_m \in \mathbb{R}^{(p-1)\times p}$ from Section 3.4, which can be used to specify linear constraints of the form $\boldsymbol{C}_m\boldsymbol{b} \geq \boldsymbol{0}$ to impose unimodality with mode at the m-th element on a numeric vector $\boldsymbol{b} \in \mathbb{R}^p$.

Value

Matrix $\boldsymbol{C}_m$ with coefficients for the linear constraints.

Examples

```
unimat(4,2)
#     [,1] [,2] [,3] [,4]
#[1,]   -1    1    0    0
#[2,]    0    1   -1    0
#[3,]    0    0    1   -1
unimat(5,3)
#     [,1] [,2] [,3] [,4] [,5]
#[1,]   -1    1    0    0    0
#[2,]    0   -1    1    0    0
#[3,]    0    0    1   -1    0
#[4,]    0    0    0    1   -1
```

Function `unimatind`

`unimatind` *Specify the matrix of unimodality constraints by two matrices.*

Description

Returns two matrices which specify the constraint matrix $\boldsymbol{C}_m$ in a special way. Those can be used as arguments in `solve.QP.compact` to specify linear constraints $\boldsymbol{C}_m\boldsymbol{b} \geq 0$ to impose unimodality with mode m on a numeric vector $\boldsymbol{b} \in \mathbb{R}^p$

Usage

```
unimatind(p, m)
```

Arguments

`p`	Integer (≥ 2) giving the length of the vector $\boldsymbol{b}$.
`m`	Location of the mode within the vector $\boldsymbol{b}$. Should be an integer between 1 and p.

Details

The function `unimatind` determines the matrix $\boldsymbol{C}_m \in \mathbb{R}^{(p-1)\times p}$ from Section 3.4 in terms of two matrices required by function `solve.QP.compact` (used inside function `unisplinem`, see below).

The first matrix, `Amat`, holds column-wise all non-zero elements of $\boldsymbol{C}_m$. The second matrix, `Aind`, specifies how many non-zero elements each column of $\boldsymbol{C}_m$ has and indicates their positions.

Value

A list of

`Amat`	The first matrix.
`Aind`	The second matrix.

Examples

```
unimatind(4,2)
#$Amat
#     [,1] [,2] [,3]
#[1,]   -1    1    1
#[2,]    1   -1   -1

$Aind
#     [,1] [,2] [,3]
#[1,]    2    2    2
#[2,]    1    2    3
#[3,]    2    3    4
```

Function `unisplinem`

`unisplinem`	*Penalized unimodal regression with fixed mode m.*

Description

Perform tuning parameter optimization and estimation of regression coefficients for a fixed mode m in a (shape-constrained) spline regression model.

Usage

```
unisplinem(m, tBSB, tySB, sigmaest, tbetaV, tbetaVbeta, Dtilde,
rangV, B, beta0, Om, constr, inverse, penalty, tuning)
```

Arguments

`m`	The specified mode m of the B-spline coefficients.
`tBSB`	The value of $\boldsymbol{B}' diag(\boldsymbol{w}) \boldsymbol{B}$ (passed over from function `unireg`, cf. Section 5.1).
`tySB`	The value of $\boldsymbol{y}' diag(\boldsymbol{w}) \boldsymbol{B}$ (passed over from function `unireg`).
`sigmaest`	The estimate of σ^2 (passed over from function `unireg`).
`tbetaV`	The value of $\boldsymbol{\beta}_0' \boldsymbol{\Omega}$ (passed over from function `unireg`).
`tbetaVbeta`	The value of $\boldsymbol{\beta}_0' \boldsymbol{\Omega} \boldsymbol{\beta}_0$ (passed over from function `unireg`).
`Dtilde`	$\tilde{\boldsymbol{D}} \in \mathbb{R}^{d \times d} = \frac{1}{\sigma_v^2} \boldsymbol{I}_d$ if $\text{rank}(\boldsymbol{\Omega}) < d$ and a zero matrix else (passed over from function `unireg`).
`rangV`	Rank r of $\boldsymbol{\Omega}$, integer between 1 and d.
`B`	The matrix of B-spline basis functions evaluated at the values in $\boldsymbol{x}$. $\boldsymbol{B} \in \mathbb{R}^{n \times d}$.
`beta0`	The penalty vector $\boldsymbol{\beta}_0$, a length d numeric vector.
`Om`	The penalty matrix $\boldsymbol{\Omega}$, a numeric $d \times d$ matrix.
`constr`	A character string specifying the shape constraint for the fit. Can be one of "unimodal", "none" and "isotonic".
`inverse`	Either -1 or 1. In case of -1, the shape constraint is "reversed" from unimodal to inverse unimodal or from isotonic to antitonic.

`penalty`	A character specifying the penalty of the model. Can be one of `"diff"`, `"none"`, `"sigEmax"` or `"self"`.
`tuning`	Logical indicating, if the tuning parameter lambda should be optimized with (`tuning=TRUE`, default, computationally expensive) or without (`tuning=FALSE`) consideration of the shape constraint.

Details

The function `unisplinem` is an internal function of R package `unireg` (Köllmann, 2016, see also Section 5.1). It performs tuning parameter optimization and estimation of B-spline coefficients with fixed mode m in a given (shape-constrained) spline regression model.

First of all, the constraint matrix $\boldsymbol{C}_m$ is calculated with the help of function `unimat` (see above) as `inverse*unimat(d,m)`.

If there is no penalty, the optimal tuning parameter `lambdaopt` is set to zero. Otherwise the function `negloglikFREQ` (see above) is optimized over the interval $[3, 10]$. Due to the $\log(\lambda)$-level in `negloglikFREQ`, this corresponds to tuning parameter values ranging from exp(3) to exp(10). This range turned out to be sufficiently wide when the observed y-values are transformed to the interval $[-1, 1]$ as is done in function `unireg` (cf. Section 5.1). The results of the simulation study and applications (cf. Chapters6 and 7) give no hints at a to small tuning parameter range.

If `tuning = FALSE`, the optimization is done with `constr = "none"` , corresponding to approximate REML (both normalizing constants are set to one).

The optimized tuning parameter value λ_{opt} is then used to calculate $\hat{\boldsymbol{\beta}}$. In the unconstrained case, we have $\hat{\boldsymbol{\beta}} = \left(\frac{1}{\sigma^2}\boldsymbol{B}'\boldsymbol{B} + \lambda_{opt}\boldsymbol{\Omega}\right)^{-1}\left(\frac{1}{\sigma^2}\boldsymbol{B}'\boldsymbol{y} + \lambda_{opt}\boldsymbol{\Omega}\boldsymbol{\beta}_0\right) = \boldsymbol{e}_{\lambda_{opt}}$.

In presence of a shape constraint, the function `solve.QP.compact` (faster than `solve.QP`) from package `quadprog` (Turlach and Weingessel, 2013) is used to minimize the objective function $\frac{1}{\sigma^2}\left\|\boldsymbol{y} - \boldsymbol{B}\boldsymbol{\beta}\right\|_2^2 + \lambda\left\|\boldsymbol{\Omega}^{\frac{1}{2}}(\boldsymbol{\beta} - \boldsymbol{\beta}_0)\right\|_2^2$ subject to $\boldsymbol{C}_m\boldsymbol{\beta} \geq \boldsymbol{0}$.

For the function `solve.QP.compact` the constraint matrix has to be specified by two matrices: the first holds all non-zero elements of $\boldsymbol{C}_m$ and the second specifies how many non-zero elements each column of $\boldsymbol{C}_m$ has and indicates their positions. The function `unimatind` (see above) calculates those matrices.

In rare cases the function `solve.QP.compact` returns coefficient vectors with `NaN`-

entries.[1] To avoid continuation of `NaN`-values, the coefficient vector is tested for such entries and if they exist, the execution of `solve.QP.compact` is repeated with the argument `factorized=TRUE` requiring a preceding Cholesky decomposition of $\mathbf{E}_\lambda^{-1}$. This approach is slower but does not fail in those situations.
At last, the fitted values are calculated as $\hat{\boldsymbol{y}} = \boldsymbol{B}\hat{\boldsymbol{\beta}}$. and

Value

A list of

`coef`	The estimated B-spline coefficients with mode m.
`fitted.values`	The fitted values.
`lambdaopt`	The optimized tuning parameter value.

[1] It was not yet possible to explain when and why this behaviour occurs. The package maintainer was contacted, but has not yet responded.

E Projecting vectors into the space of unimodal vectors with fixed mode

Let $\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with mean vector $\boldsymbol{\mu} \in \mathbb{R}^d$, covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ and let $\tilde{\boldsymbol{\beta}} \in \mathbb{R}^d$ be a random draw from this distribution, which we wish to project into the space $\mathcal{S}_m \subset \mathbb{R}^d$ of unimodal vectors with mode m. According to Gunn and Dunson (2005) this can be done using their formula 2.2, that is, each element of the constrained vector can be calculated as follows:

$$\boldsymbol{\beta}_j^{*,m} = \min_{t \in U_j^m} \max_{s \in L_j^m} \left\{ \frac{\mathbf{1}'_{|t-s|+1} \boldsymbol{\Sigma}^{-1}_{[s:t]} \tilde{\boldsymbol{\beta}}_{[s:t]}}{\mathbf{1}'_{|t-s|+1} \boldsymbol{\Sigma}^{-1}_{[s:t]} \mathbf{1}_{|t-s|+1}} \right\},$$

with the subscript $[s:t]$ representing the respective sub-matrices or sub-vectors. The set L_j^m contains all indices $j' \in \{1, \ldots, d\}$ for which the ordering $\beta_{j'}^{*,m} \leq \beta_j^{*,m}$ is *known* and the set U_j^m contains analogously all indices $j' \in \{1, \ldots, d\}$ for which the ordering $\beta_{j'}^{*,m} \geq \beta_j^{*,m}$ is *known* (cf. Gunn and Dunson, 2005).
Let $j < m$. Then $L_j^m = \{1, \ldots, j\}$ and $U_j^m = \{j, \ldots, m\}$. Analogously $L_j^m = \{j, \ldots, d\}$ and $U_j^m = \{m, \ldots, j\}$ for $j > m$. If $j = m$, we have $L_j^m = \{1, \ldots, d\}$ and $U_j^m = \{m\}$.
Each element of this projection can be interpreted as a weighted average of elements in $\tilde{\boldsymbol{\beta}}$. The weights are chosen with respect to the covariances in $\boldsymbol{\Sigma}$ and the elements of $\tilde{\boldsymbol{\beta}}$ contributing to the sub-vector are chosen with respect to the constraint (cf. Dunson and Neelon, 2003).