2017-01-01

Estimating The Coefficients Of A Linear Differential Operator

Maria Ivette Barraza
University of Texas at El Paso, mibrios@gmail.com

ESTIMATING THE COEFFICIENTS OF A LINEAR DIFFERENTIAL OPERATOR

MARÍA IVETTE BARRAZA

Doctoral Program in Computational Science

APPROVED:

_____
Joan G. Staniswalis, Chair, Ph.D.

_____
Ori Rosen, Ph.D.

_____
Rodrigo A. Romero, Ph.D.

_____
Charles Ambler, Ph.D.
Dean of the Graduate School

Para J, en su compañía,

para A, L, D y A, en su cariño

ESTIMATING THE COEFFICIENTS OF A LINEAR DIFFERENTIAL OPERATOR

by

MARÍA IVETTE BARRAZA

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

Doctoral Program in Computational Science

THE UNIVERSITY OF TEXAS AT EL PASO

December 2017

Acknowledgements

Dr. Joan G. Staniswalis is a wonderful person and advisor. She has been a gracious, generous and determined guide, always providing support, motivation and inspiration. My deepest gratitude and admiration remain with her.

I sincerely thank committee members Dr. Ori Rosen and Dr. Rodrigo Romero for their insights, thoughtful support, and patience.

I recognize and am grateful for the generous support that this work received from the Department of Mathematical Sciences, and the support from grant NIH-SCORE 5SC3GM094073.

Abstract

Principal Differential Analysis (PDA; Ramsay, 1996) is used to obtain low dimensional representations of functional data, where each observation is represented as a curve. PDA seeks to identify a Linear Differential Operator (LDO) $L = \omega_0 I + \omega_1 D + ... + \omega_m D^m$, where I denotes the identity function and D^j the jth derivative, that satisfies as closely as possible that $Lx = 0$ for each functional observation x. A theorem from analysis establishes that the coefficients of the LDO are in the Sobolev space, and thus can be approximated by B-splines. Current PDA software used to estimate the LDO assumes that the leading coefficient is 1, and approximates the LDO coefficients by B-splines, even though the Sobolev space is not closed under division. We present a method that eliminates the restriction on the leading coefficient of the LDO. The proposed resampling method is inspired by results in linear regression (Frees, 1991 and Wu, 1986) that show that the weighted average of all $\binom{n}{m}$ pairwise slopes between n data points is equivalent to the least squares estimator of the regression line slope. In analyzing data, least-squares estimates and robust estimates of the LDO coefficients are computed, and results based on each estimate are compared. The `R` language implementation of the proposed method is available at https://www.utep.edu/science/computational-science/research/resources.html.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

A cochlear implant is a small electronic device, surgically placed under the skin behind the ear to provide a sense of sound to persons who are profoundly deaf. Hearing aides are designed to amplify sound, but a cochlear implant converts sound waves in the air to electrical impulses. The cochlear implant then transmits the signal (electrical impulses) to electrodes implanted in the brain, thereby stimulating auditory nerve fibers. Hearing with the cochlear implant may sound differently from unimpaired hearing, nevertheless many hearing impaired people learn to compensate for the difference allowing for full oral communication, in person or over the phone. Both children and adults are candidates for cochlear implants. The best age for implantation has not been determined, although, early implantation prior to the age of four seems to be most successful. The varying levels of success of cochlear implants are believed to be due to the plasticity of the central auditory system, that is, the ability of the brain to respond to sound signals with electrical activity. This electrical activity is quantified by a cortical auditory evoked potential (CAEP) curve. Much effort has been directed towards describing and understanding how the CAEP curves change with age in subjects with normal hearing, since it is believed to reflect changes in plasticity of the brain with age (Sharma, Dorman, and Spahr 2002). It is believed that better clinical outcomes are obtained when cochlear implantation takes place before plasticity of the brain is lost.

The CAEP curves of $n = 74$ children with normal hearing are displayed in Figure 1.1. The children vary in age from a few months to less than 7 years of

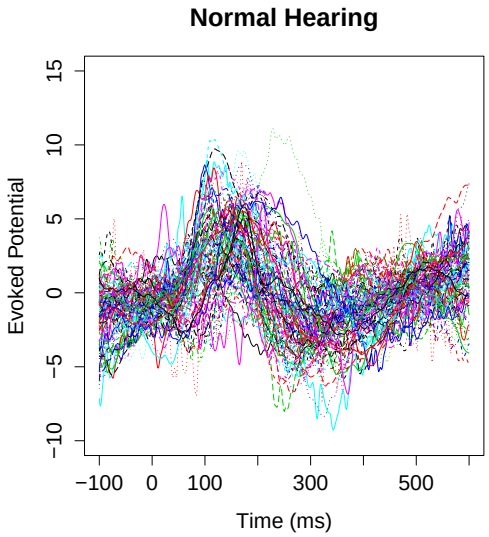


Figure 1.1: CAEP Curves ($n = 74$) Displayed with Light Smoothing (P-spline of order 4) for Children Under the Age of 7 with Normal Hearing.

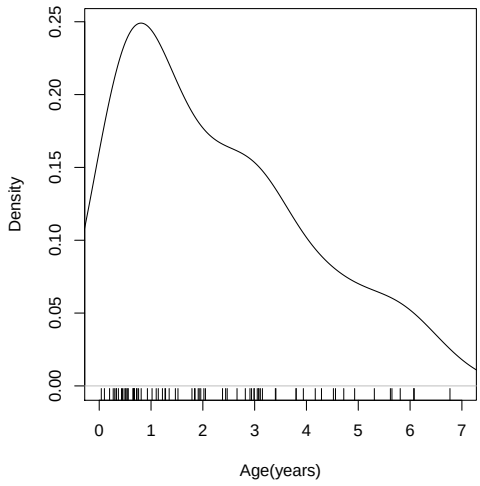


Figure 1.2: Age Distribution of the Children with Normal Hearing.

age; see Figure 1.2 for the distribution of ages. The stimulus that produced these responses (CAEP curves) was a synthesized speech syllable /ba/ presented at an interstimulus interval of 500 ms via a loudspeaker placed at an angle of 45° to the

side of the head; see Sharma, Dorman, and Spahr (2002). The displayed CAEP curves are each an average over two runs of three hundred replicates (sweeps) for each child. Each replicate constitutes presentation of the syllable /ba/ at time 0, recording begins 100 ms before presentation of the syllable and continues for 600 ms after onset. The replicability of the waveforms taken as an average over multiple sweeps (replicates) was investigated and demonstrated in Dau et al. 2000 (their Figure 2). The P1 wave of the CAEP is the first positive peak after presentation of the synthetic auditory stimulus and is manually scored by a trained audiologist. The location of the P1 wave in time, called the P1 latency, is used as a biomarker for maturation (plasticity) of central auditory pathways and has been reported to decrease with age (Sharma, Dorman, and Spahr 2002).

In this dissertation we will consider techniques from nonparametric regression and principal differential analysis (PDA introduced by Ramsay in 1996) to model such a collection of n curves as in Staniswalis, Dodoo, and Sharma (2015) and Jin, Staniswalis, and Mallawaarachchi (2013). PDA seeks to identify a Linear Differential Operator (LDO) $L = \omega_0 I + \omega_1 D + ... + \omega_m D^m$, where I denotes the identity function and D^j the jth derivative, that satisfies as closely as possible that $Lx = 0$ for each functional observation x. A theorem from analysis establishes that the coefficients of the LDO are in the Sobolev space, and thus can be approximated by B-splines. Current PDA software used to estimate the LDO assumes that the leading coefficient is 1, and approximates the LDO coefficients by B-splines, even though the Sobolev space is not closed under division.

This dissertation presents a method that eliminates the restriction on the leading coefficient of the LDO. A secondary result is that the method allows for a modification of the procedure according to (Wu 1986) when outliers are present in the data. Thirdly, PDA uses a regularization step when estimating the coefficients, whereas the proposed method estimates the pointwise coefficients and then includes a regularization step. The regularization step involves smoothing

the coefficients with cubic splines, and using cross-validation to assist in the selection of the smoothing parameter.

Chapters 2 and 3 serve as brief background in the area of nonparametric regression and splines, needed to understand PDA. Chapter 2 provides basic definitions of parametric and nonparametric regression analysis. The minimization criteria to find the "best" curve that describes a data set is defined in terms of parametric and nonparametric statistics, and the concept of a smoothing parameter is introduced. Chapter 3 presents definitions and key concepts on splines. In particular, B-splines are defined, which are the basis for current PDA methodology. Chapter 4 introduces the concept of functional data and its analysis through PDA methodology. The assumptions and algorithm of current PDA methodology are described in more detail. Chapter 5 presents the mathematical statistical foundation of the resampling method proposed in this dissertation. The chapter includes a justification for breaking up the problem into $\binom{n}{m}$ sub-problems in Theorem 2. Theorem 3 suggests how to choose the least squares solution to eliminate the restriction that $\omega_m = 1$, while Wu (1984) offers a variation of an estimator less sensitive to outliers. Chapter 6 shows the results obtained after applying the resampling method to two data sets represented by an LDO of order 2. Chapter 7 discusses the resampling method applied to data represented by LDO's of order $m = 4$. One of the data sets analyzed in the chapter is the set of P1 waves in the CAEP curves of children with normal hearing. In Chapter 8 the set of P1 waves are divided into subgroups by age, and each of the subgroups is analyzed. Overall conclusions and future directions of work are listed at the end of the chapter.

Chapter 2

Regression Analysis

Regression analysis is used to model and investigate relationships among variables. For example, the relationship between a dependent variable y and independent variable t may be studied through the model

$$y = f(t) + \epsilon, \quad t \in [a, b], \tag{2.1}$$

where ϵ is a random unobservable error with mean 0 and variance σ^2. Suppose we have data of the form (t_i, y_i) for $i = 1, \ldots n$, satisfying equation (2.1). There are two general approaches to estimation of the regression curve $f(t)$: parametric, with which most people are familiar; nonparametric, now commonly available in most statistical software packages. The rest of the chapter reviews concepts in parametric and nonparametric regression.

2.1 Parametric Regression

In parametric regression, it is assumed that the functional form of the regression curve $f(t)$ is known. For example, in equation (2.1) f is a known functional form $f(t) = \beta_o + \beta_1 t$ involving unknown parameters β_0 and β_1. The parameter space of the functional form is finite-dimensional and the number of parameters is much smaller than the sample size n. In the example, the information in the n data points regarding the pattern between the two variables t and y is summarized by the two parameter estimates: $\hat{\beta}_1$ for the slope and $\hat{\beta}_o$ for the y-intercept of the line. The parameter estimates can be obtained, for example, by ordinary least squares.

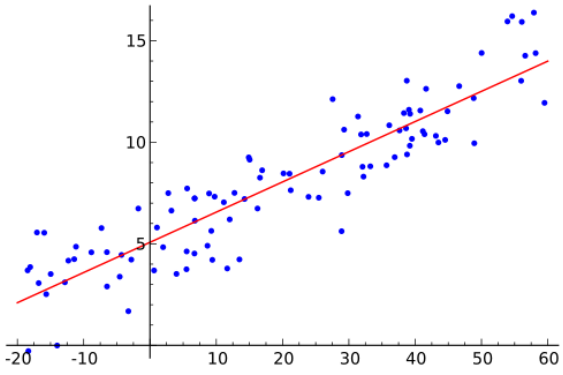


Figure 2.1: Linear Regression Example

The ordinary least squares estimators of the parameters β_0 and β_1 are given by

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}=(\beta_0,\beta_1)}{\operatorname{argmin}} \left[\sum_{i=1}^{n} (Y_i - \beta_0 - \beta_1 t_i)^2 \right].$$

The resulting fitted-line is superimposed upon the scatterplot of simulated data in Figure 2.1.

2.2 Nonparametric Regression

In nonparametric regression, we only know that the true regression function $f(t)$ is smooth in t; its shape is estimated from the data. In the example plotted in the scatterplot of Figure 2.2, we see that the estimate $\hat{f}$ of the regression function f follows the general pattern of the points, and there is no reduction of the data to a finite-dimensional set of parameter estimates. The methods for nonparametric regression fall into two main categories: kernel methods and splines. Kernel methods (based on dampening Fourier coefficients) and spline (based on sets of basis functions) estimators have been shown to be asymptotically equivalent (see Messer (1991) and B. Silverman (1984)). In either case the following description holds:

- the regression function f is unknown, but assumed to be smooth;

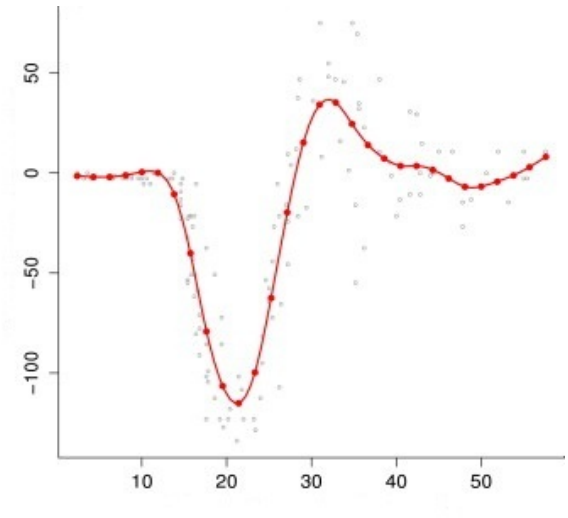


Figure 2.2: Nonparametric Regression Example

- the shape of f is recovered by filtering the noisy data;
- the dimension of the parameter space increases with the sample size, there is no reduction of the data.

Consider the nonparametric estimator of the regression curve obtained as the minimizer of

$$\frac{1}{n}\sum_{i=1}^{n}\left[y_i - f(t_i)\right]^2 + \lambda \int_a^b \left[f^{(r)}(t)\right]^2 dt, \tag{2.2}$$

over f in the Sobolev space denoted by $W_2^r[a,b]$. The Sobolev space of order r is the set of all functions with $r-1$ continuous derivatives and the integral of the square of the r^{th} derivative exists. The first term of the minimizing criteria (2.2) is a residual sum of squares, as was used in ordinary least squares regression. The second term involves a penalty, where λ is a positive scalar called the smoothing parameter. The first term ensures a fit that is close to the data, while the second term helps to obtain a smooth curve. If the smoothing parameter λ is close to zero, the smoothing spline is an interpolating spline, tracking the data closely. As λ increases without bound, the smoothing spline converges to a polynomial regression model. When the penalty uses the r^{th} derivative of the regression function,

the polynomial regression model is of order $2r - 1$. The solution f_λ to this minimization problem is called a smoothing spline (Wahba 1990) and it is a natural spline (Boor 1978). Basic concepts of splines are reviewed in the next chapter.

Chapter 3

Splines

A spline function $S(t)$ is a piecewise polynomial function defined on the entire real line, subject to a maximum number of continuity constraints. This chapter covers formal definitions and discussions of key concepts about splines (the truncated power basis, the B-splines, natural splines). Finally, we consider P-splines in place of smoothing splines.

3.1 The Truncated Power Basis for Splines

Definition 1 *A **spline** of order $k \geq 2$ with knots $t_1 < t_2 < \cdots < t_\eta$ is any function $S(t)$ of the form*

$$S(t) = \sum_{i=0}^{k-1} \alpha_i t^i + \sum_{i=1}^{\eta} \delta_i (t - t_i)_+^{k-1} \tag{3.1}$$

for real $\alpha_0, \cdots, \alpha_{k-1}, \delta_1, \cdots, \delta_\eta$ satisfying that $S(t)$ has $k-2$ continuous derivatives.

Definition 1 describes splines in terms of the polynomial functions: $1, t, \ldots, t^{k-1}$ and truncated power basis

$$(t - t_i)_+ = \begin{cases} (t - t_i), & \text{if } t \geq t_i \\ 0, & \text{if } t < t_i \end{cases} \qquad i = 1, \ldots, \eta$$

In simpler terms, a spline is a linear combination of polynomial functions and truncated polynomials that depend on the *knots* $t_1, \ldots t_\eta$ chosen. The polynomial and truncated polynomials are joined together smoothly at the knots so that the $(k-2)^{\text{th}}$ derivative of the spline exists and is continuous everywhere. Note that

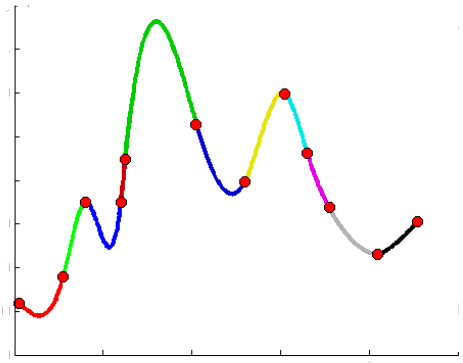


Figure from http://profmsaeed.org/wp-content/uploads/2011/03/CubicSpline.html

Figure 3.1: Example of cubic spline

if the spline is of order k, then the largest exponent of t is $k-1$, so the degree of the spline is $k-1$. For example, Figure 3.1 shows a spline of order 4. Each of the segments in different colors are pieces of cubic polynomials, that is, a cubic polynomial in each interval $[t_i, t_{i+1}]$, $i = 1, \ldots, \eta - 1$. These piecewise polynomials are joined smoothly at the knots, which are marked with red circles. All the derivatives up to and including the second derivative exist and match at the knots. To better approximate a curve using splines, one typically fixes the order of the polynomial and increases the number η of the knots.

It can be shown (Eubank 1999, p. 281) that Definition 1 is equivalent to

1. $S(t)$ is a piecewise polynomial of order k on any subinterval $[t_i, t_{i+1})$.

2. the $(k-1)^{\text{th}}$ derivative, $S^{(k-1)}(t)$ is discontinuous with jumps at the knots $t_1, \ldots, t_\eta$.

Let $S^k(t_1, \ldots, t^\eta)$ be the set of all functions of the form (3.1). Then $S^k(t_1, \ldots, t^\eta)$ is a vector space with dimension $\eta + k$.

3.2 Natural Spline

Natural splines are mentioned in the previous chapter as the minimizer of equation (2.2).

Definition 2 *A spline function $S(t)$ is a **natural spline** of order $k = 2r$ with knots at $t_1, \ldots, t_\eta$ if $S(t)$ is a polynomial order r (degree $r - 1$) outside $[t_1, t_\eta]$.*

The minimizer of equation (2.2) is in fact a natural spline of order $2r$ (Eubank 1999, p. 239). The dimension of the basis for natural splines is η.

3.3 The B-Spline Basis for Splines

B-splines can be used as a basis for splines, instead of the truncated power functions. In particular, f_λ, the smoothing spline estimator can be represented with B-splines. Numerical computation with B-splines is more stable than with the truncated power basis. B-splines can be computed as differences of truncated power functions, but they are usually computed following the Cox-de Boor recursion formula. To initialize the recursion, we define an additional k knots and "stack" them on the endpoints of the interval $[a, b]$ we are considering. Note that if $k = 2r$ is the order of the spline, we stack r knots on each endpoint of the interval. Given knots $t_1, \cdots, t_\eta$ on $[a, b]$, define $k = 2r$ additional knots as follows:

$$t_{-r+1} = \cdots = t_0 = a$$

$$b = t_{\eta+1} = \cdots = t_{\eta+r}.$$

Then the B-spline of order k is defined by the recursion

$$B_{i,1}(t) = \begin{cases} 1, & t_i \leq t < t_{i+1} \\ 0, & \text{otherwise} \end{cases} \qquad i = 0, \ldots, \eta$$

$$B_{i,k}(t) = \left(\frac{t - t_i}{t_{i+k-1} - t_i}\right) B_{i,k-1}(t) + \left(\frac{t_{i+k} - t}{t_{i+k} - t_{i+1}}\right) B_{i+1,k-1}(t) \qquad \text{for } i = -(k-1), \ldots, \eta.$$

The recursion for computing the B-splines starts with the indicator functions (i.e., splines of order 1) that are zero everywhere except in the interval between two consecutive knots. In the recursion, a spline of order k depends on two adjacent splines of order $k-1$. The B-splines of order 2 are called sawtooth functions. The B-splines of order 3 are piecewise quadratic polynomials. The B-splines of order 4 are piecewise cubic polynomials. Figure 3.2 shows graphs of the B-splines of orders 1, 2, 3, and 4. Note that it is harder to pinpoint the location of the knots on the B-splines of orders 3 and 4. Also, as the order of the spline increases, the support of the spline increases. Following are some of the properties of the B-spline basis:

1. $B_{i,k}$ has small support, i.e., $B_{i,k}(t) = 0$ if $t \notin [t_i, t_{i+k}]$;
2. $\sum_i B_{i,k}(t) = 1$ for $t \in [a, b]$;
3. $B_{i,k}(t)$ is positive on its support $[t_i, t_{i+k}]$.

3.4 P-splines

Consider the spline regression model $f(t) = \sum_j \alpha_j B_{j,k}(t)$, where $\{B_{j,k}\}_{j=-(k-1)}^{n}$ are B-splines basis functions of order k. (Eilers and Marx 1996) proposed penalizing the ordinary least squares problem for fitting a spline regression model by a difference penalty to control smoothness. More precisely, rather than minimizing the function

$$\sum_{i=1}^{n} \left[y_i - \sum_j \alpha_j B_{j,k}(t_i) \right]^2 + \lambda \int_a^b \left[\sum_j \alpha_j B_{j,k}^{(r)}(t) \right]^2 dt,$$

(Eilers and Marx 1996) proposed to replace the penalty with higher-order finite differences of the coefficients of adjacent B-splines, and instead minimize the function

$$\sum_{i=1}^{\eta}\left[y_i - \sum_j \alpha_j B_{j,k}(t_i)\right]^2 + \lambda \sum_j (\Delta^r \alpha_j)^2,$$

where

$$\begin{aligned}
\Delta\alpha_j &= \alpha_j - \alpha_{j-1} \\
\Delta^2\alpha_j &= \Delta(\Delta\alpha_j) \\
&= (\alpha_j - \alpha_{j-1}) - (\alpha_{j-1} - \alpha_{j-2}) \\
&= \alpha_j - 2\alpha_{j-1} + \alpha_{j-2} \\
&\vdots \\
\Delta^r\alpha_j &= \Delta(\Delta^{r-1}\alpha_j).
\end{aligned}$$

The solution to this minimization problem is termed the P-spline for the penalized regression spline.

The sum of the squared differences of the coefficients can be written in matrix form:

$$\sum_j (\Delta^r \alpha_j)^2 = \boldsymbol{\alpha}^T \mathbf{D}_r^T \mathbf{D}_r \boldsymbol{\alpha}.$$

For example, for $r = 2$

$$\mathbf{D_2} = \begin{bmatrix} 1 & -2 & 1 & 0 & \cdots \\ 0 & 1 & -2 & 1 & \cdots \\ 0 & 0 & 1 & -2 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}_{(J-2)\times J} , J = n + k.$$

For P-splines, the parameter λ still provides control over smoothing. There is a strong connection between second order finite difference penalty and integrated square of the second derivative. However, unlike smoothing splines, the order of the derivative (difference) in the penalty is not necessarily linked to the order of

the B-spline basis used to approximate the nonparametric regression function. In other words, we are not restricted to r satisfying $k = 2r$.

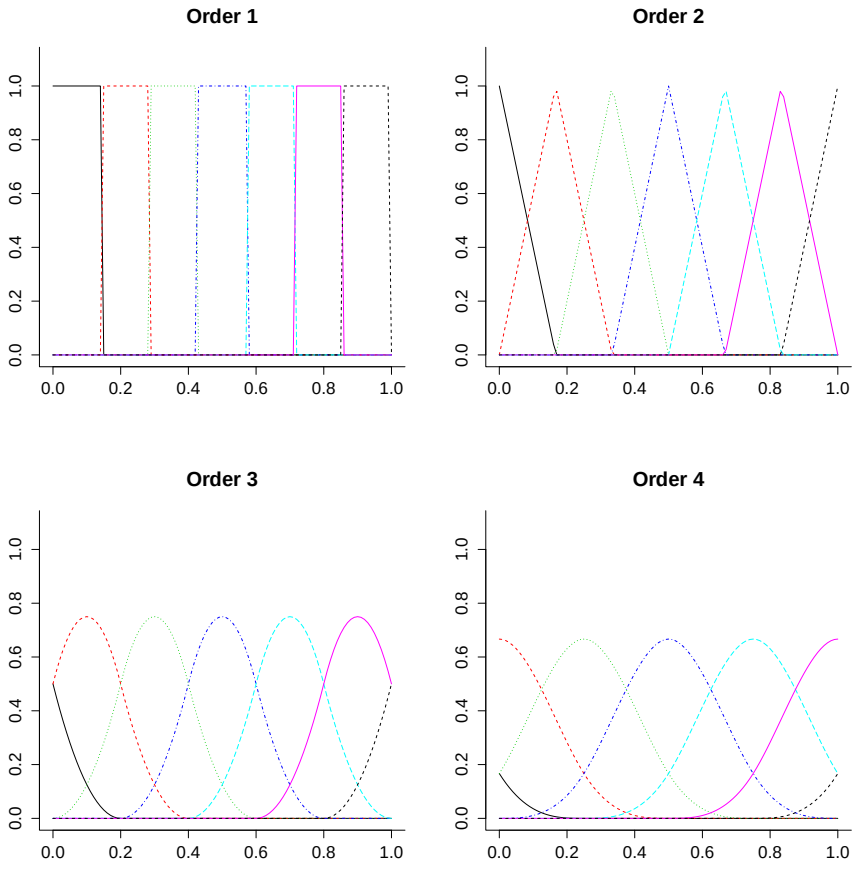


Figure 3.2: B-splines of orders 1, 2, 3 and 4

Chapter 4

Functional Data

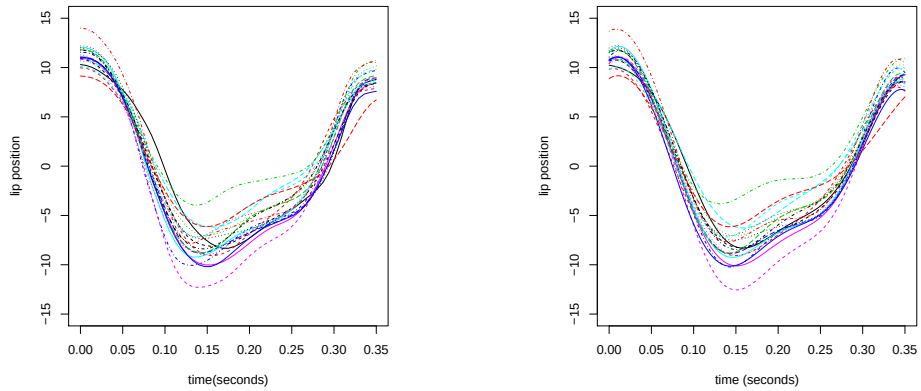
Functional data analysis arose because advances in technology allow for the recording of responses on a finer grid. For example, atmospheric phenomena that used to be monitored once a day are now monitored (almost) continuously. The CAEP curves in Figure 1.1 are an example of functional data. A child contributes a single response described by a CAEP curve.

Figure 4.1 illustrates another example of functional data. This collection of curves considered by Ramsay and B.W. Silverman (1997), represent the position of the center of the bottom lip of a person saying the syllable "bob" 20 times. The methodology called Principal Differential Analysis (PDA) was developed by Ramsay (1996). Its objective is to "use a set of n functional observations (curves) $\{x_i\}_{i=1}^n$ to define a much smaller set of m functions $\{u_j\}_{j=1}^m$ on the basis of which we can obtain efficient approximations of the observed functions" (Ramsay and B.W. Silverman 1997, p.239). For example, applied to the lip data, PDA uses the set of 20 curves to define a set of 2 functions u_1 and u_2 such that linear combinations of u_1 and u_2 capture the behavior of each of the 20 curves.

4.1 Principal Differential Analysis

PDA assumes that there exists a linear differential operator (LDO) defined by

$$L = \omega_0 I + \omega_1 D + \cdots + \omega_{m-1} D^{m-1} + \omega_m D^m, \tag{4.1}$$



(a) Unregistered lip data (b) Registered lip data

Figure 4.1: Lip data, 20 curves. The registered lip data is used throughout this document.

where I denotes the identity function and D^j denotes the jth derivative, that satisfies $Lx_i = 0$ for each functional observation $x_i(t)$, $i = 1, \cdots, n$. The coefficients of the LDO (the functions $\omega_0, \cdots, \omega_m$, which depend on t) need to be estimated such that the LDO annihilates the collection of curves as closely as possible. Recall, that the null space of the LDO is the collection of all functions annihilated by the LDO.

We know from results in analysis (Coddington and Levinson 1955, Theorem 6.2) that a collection of m functions from a Sobolev space has such an annihilating LDO of order m. The following theorem (Jin, Staniswalis, Mallawaarachchi 2012) goes further by providing conditions under which the coefficients of an annihilating LDO are also in a Sobolev space, and thus can be approximated well by B-splines.

Theorem 1 *Let $f_1, \cdots, f_m \in$ Sobolev Space of order $2m$ (denoted by $W_2^{2m}(a, b)$) be linearly independent functions, and consider $\mathcal{M} = span\{f_1, \cdots, f_m\}$. Then $\mathcal{M}$ has an m^{th} order annihilating LDO of the form $L = \omega_0 I + \omega_1 D + \cdots + \omega_{m-1} D^{m-1} + \omega_m D^m$* **with coefficients in the Sobolev Space of order** *m and $Lf = 0$ for all $f \in \mathcal{M}$.*

The constructive proof of the theorem gives us a way to find the LDO coefficients using Cramer's rule. Let $\omega_r(t) = (-1)^{m-r}\mathbf{M_r(t)}$, for $r = 0, \cdots, m$, where $\mathbf{M_r(t)}$ is the determinant of the Wronskian minor, obtained by deleting the $(r+1)^{\text{th}}$ column of the Wronskian matrix:

$$\begin{bmatrix} f_1 & Df_1 & \cdots & D^m f_1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ f_m & Df_m & \cdots & D^m f_m \end{bmatrix}_{m \times (m+1)}$$

Note that the coefficients of the LDO are not unique, since the null space is invariant to multiplication of the LDO by any non-zero function from the Sobolev space. Ramsay (1996) avoids this problem by assuming that the coefficient of the leading derivative in the LDO is $\omega_m(t) = 1$. This is equivalent to dividing through by ω_m in the LDO in order to obtain a unique solution for the coefficients. However, the Sobolev space is not closed under division, in which case we can no longer assume that the coefficients $\omega_0, \ldots, \omega_{m-1}$ are in the Sobolev space and approximated well by B-splines. Ramsay leaves this concern aside in the hope of finding a useful working solution. Ramsay represents the coefficients $\omega_0(t), \cdots, \omega_{m-1}(t)$ in a B-spline basis, then estimates the smooth coefficients of the LDO by minimizing $\sum_{i=1}^n \|Lx_i\|^2 = \sum_{i=1}^n \int_0^T (Lx_i(t))^2 dt$ subject to a penalty term. Once the coefficients of the LDO are estimated, numerical methods from differential equations are used to construct a basis for the null space of the LDO. Ramsay (1996) uses the Runge-Kutta numerical method to find a basis for the null space of the LDO with coefficients estimated from the data when $\omega_m(t) = 1$. Dropping the assumption $\omega_m(t) = 1$ requires the use of a different set of numerical methods, namely, those that solve implicit systems of differential equations. Then, normalized basis functions $u_1(t), \ldots, u_m(t)$ are found by dividing each basis function by the square root of the norm. The lip data was registered with marks provided in the `fda` library. For the registered lip data, which will be used throughout this

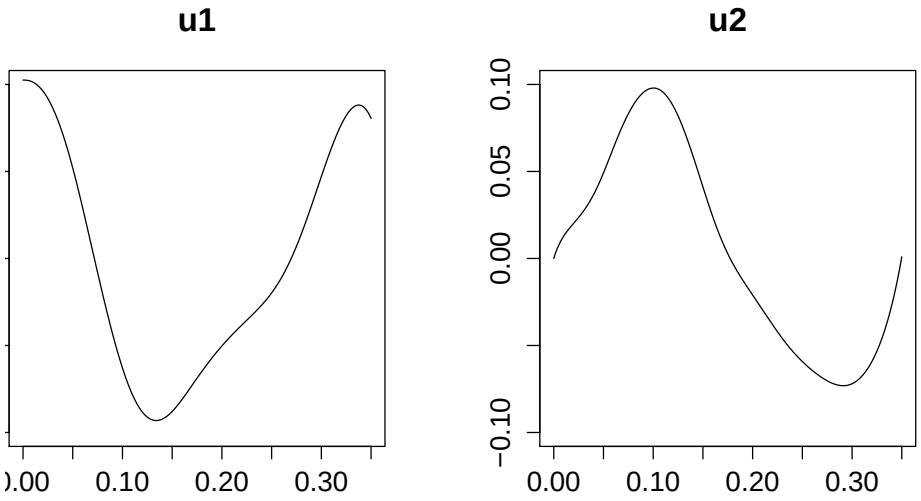


Figure 4.2: Normalized basis functions for a second-order LDO that annihilates the registered lip data, obtained with current PDA methodology.

dissertation, the normalized basis functions for a second-order LDO ($m = 2$) are provided in Figure 4.2. Finally, since each curve in the functional data set is believed to be an element of the null space of the LDO, a low-dimensional approximation is obtained by regressing each curve on the normalized basis functions $u_1(t), \ldots, u_m(t)$. For example, Figure 4.3 displays fits for the 20 curves in the registered lip data,

$$\hat{x}_i(t) = \hat{a}_{1_i}\hat{u}_1(t) + \hat{a}_{2_i}\hat{u}_2(t), \quad i = 1, \ldots, 20.$$

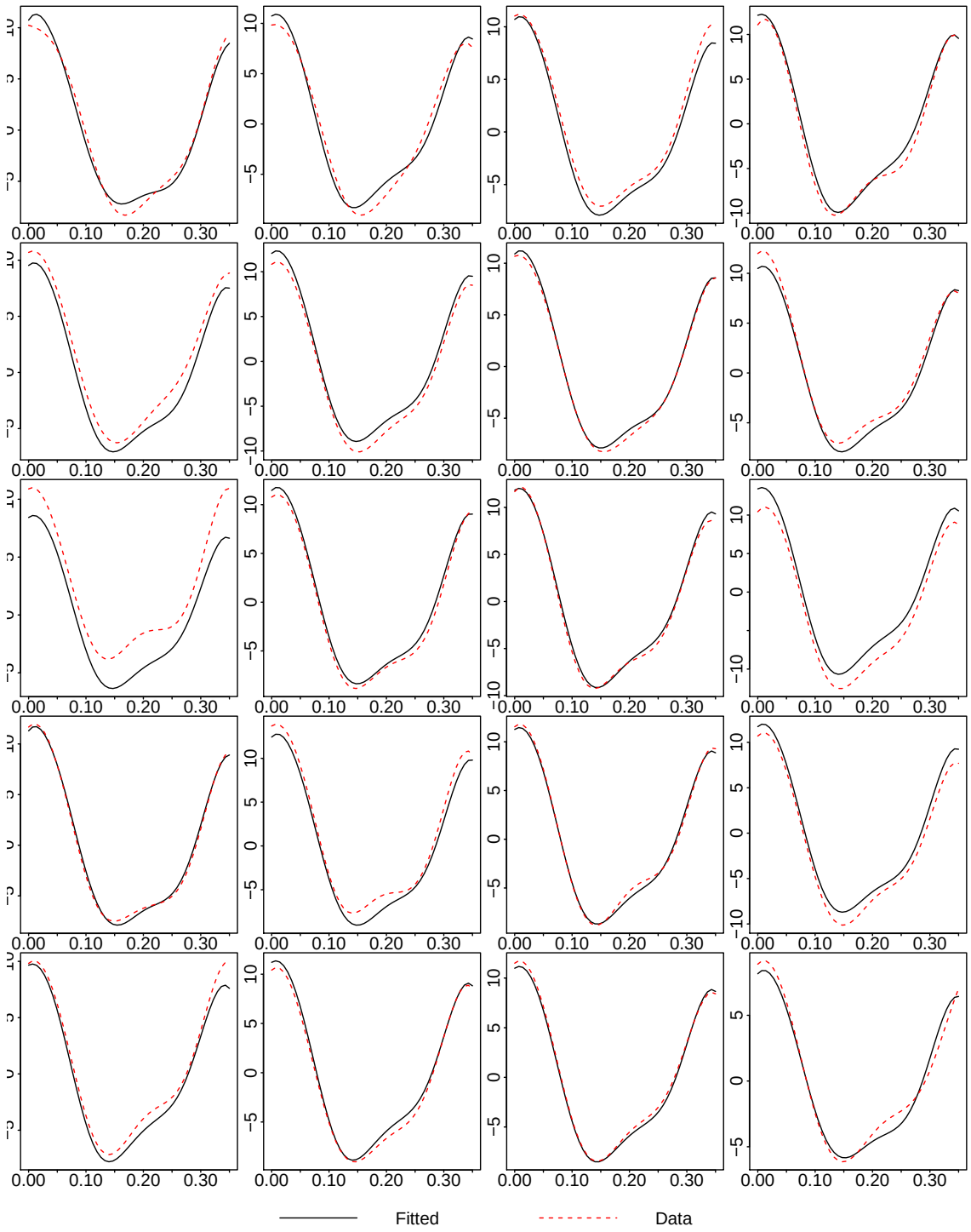


Figure 4.3: Fits of curves in registered lip data obtained with current PDA methodology ($\lambda = 0$, 21 knots, cubic B-spline).

Chapter 5

Resampling Methods

Consider a collection of observations (x_i, y_i), for $i = 1, \ldots, n$, satisfying $y_i = \beta_o + \beta_1 x_i + \epsilon_i$, where $\epsilon_1, \ldots, \epsilon_n$ are iid random variables with mean 0 and common variance σ^2. In Chapter 2, we discussed how the parameters in the regression model could be estimated by ordinary least squares. Frees (1991) investigated robust estimators of β_o, β_1 by pursuing the alternative characterization of the ordinary least squares estimator of the slope parameter. This is described next: compute all pairwise slopes

$$s_{ij} = \frac{y_j - y_i}{x_j - x_i},$$

and set

$$\beta_{Fr} = \frac{\sum w_i s_{ij}}{\sum w_i}.$$

Here, β_{Fr} is a linear combination of all $\binom{n}{2}$ pairwise slopes. It can be shown that choosing $w_i = (x_i - x_j)^2$ yields the least squares estimator

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n X_i(Y_i - \bar{Y})}{\sum_{i=1}^n X_i(X_i - \bar{X})}.$$

If instead, we first order the pairwise slopes, then trim unusually high or low slopes, an M-type estimator of the slope parameter is obtained.

The ideas in Frees (1991) inspired the resampling-based method for estimation of the coefficients $\omega_0, \ldots, \omega_m$ of the LDO in equation (4.1) presented in this dissertation. We compute the annihilating LDO for each collection of m curves sampled from the full collection of n curves. This is computationally intensive because there are $\binom{n}{m}$ sub-problems that must be solved for estimation of the LDO. Then

a linear combination of the $\binom{n}{m}$ LDO's is used as the final estimate $\hat{L}$ of the LDO in equation (4.1).

Theorem 2 below provides a justification for breaking up the problem into $\binom{n}{m}$ subproblems. In the case of an iid sample of size n, the collection of minimal sufficient statistics for all $\binom{n}{m}$ subproblems is sufficient for the joint distribution of the sample.

Let $\mathcal{F} = \{f_\theta(x), \theta \in \Omega\}$ denote a family of distributions with common support, and $X_1, \ldots, X_n$ a random sample that is iid f_θ. The joint distribution of the sample is given by $f_\theta(\mathbf{x}) = \prod_{i=1}^n f_\theta(x_i)$. Let $\mathbf{x}^* = (x_1^*, \ldots, x_m^*)$ denote a sample of size m selected without replacement from $\{x_1, \ldots, x_n\}$.

Theorem 2 *Suppose there is a k and $\theta_j \in \Omega$ for $j = 0, \ldots, k$ so that statistic*

$$T(\mathbf{x}) = (T_1(\mathbf{x}), \ldots, T_k(\mathbf{x})) = \left(\frac{f_1(\mathbf{x})}{f_0(\mathbf{x})}, \ldots, \frac{f_k(\mathbf{x})}{f_0(\mathbf{x})}\right)$$

is minimal sufficient for $\mathcal{F}$, where $f_j(\mathbf{x}) = f_{\theta_j}(\mathbf{x})$. Consider $T(\mathbf{x}^*) = (T_1(\mathbf{x}^*), \ldots, T_k(\mathbf{x}^*))$, where

$$T_j^* = T_j(\mathbf{x}^*) = \frac{\prod_{i=1}^m f_j(x_i^*)}{\prod_{i=1}^m f_0(x_i^*)}.$$

$T(\mathbf{x})$ can be recovered from the collection $\{T(\mathbf{x}^*)\}$ obtained by all possible resamples. Hence $\{T(\mathbf{x}^*)\}$ is sufficient for $\mathcal{F}$.

Proof Set $a_i = \dfrac{f_\theta(x_i)}{f_0(x_i)}, i = 1, \ldots, n$, suppressing the dependence on $\theta \in \{\theta_1, \ldots, \theta_k\}$. Then

$$\frac{\prod_{i=1}^n f_j(x_i)}{\prod_{i=1}^n f_0(x_i)} = a_1 \cdot \ldots \cdot a_n, \quad a_i > 0.$$

The product $a_1 a_2 \cdots a_n$ can be written as a product of terms of the form $a_{i_1} \cdots a_{i_m}$ as follows:

$$a_1 \cdots a_n = \left[\underbrace{(a_1 \cdots a_m)}_{\text{length } m}(a_2 \cdots a_m a_{m+1}) \cdots (a_{n-(m-1)} \cdots a_n) \cdots (a_n a_1 \cdots a_{m-1})\right]^{(1/m)}.$$

Hence the theorem is proven.

Condition 2 holds for many parametric families by an application of Theorem 6.6.5 in Casella and Berger (2002, p.309): Gaussian ($k = 2$), Binomial ($k = 1$), Logistic ($k = n$).

5.1 Resampling Solution

For each t, solve for the least squares estimators $\tilde{\omega}_{0,s}, \ldots, \tilde{\omega}_{(m-1),s}$ in

$$\underbrace{\begin{bmatrix} f_1 & Df_1 & \cdots & D^{m-1}f_1 \\ \vdots & \vdots & \vdots & \vdots \\ f_r & Df_r & \cdots & D^{m-1}f_r \end{bmatrix}}_{\mathbf{X}_s} \underbrace{\begin{bmatrix} \tilde{\omega}_{0,s} \\ \vdots \\ \tilde{\omega}_{(m-1),s} \end{bmatrix}}_{\tilde{\boldsymbol{\omega}}_s} = \underbrace{\begin{bmatrix} -D^m f_1 \\ \vdots \\ -D^m f_r \end{bmatrix}}_{\mathbf{z}_s}, \tag{5.1}$$

where $\{f_1, \ldots, f_r\} \subseteq \{x_1, \ldots, x_n\}, m \leq r \leq n$. Using notation in Ramsay (1996, Equation (5), p. 499), the least squares solution is

$$\tilde{\boldsymbol{\omega}}_s(t) = \left[\mathbf{X}_s^T(t)\mathbf{X}_s(t)\right]^{-1} \mathbf{X}_s^T(t)\mathbf{z}_s(t). \tag{5.2}$$

Note two special cases: (1) $\tilde{\boldsymbol{\omega}}_s(t) = \mathbf{X}_s^{-1}(t)\mathbf{z}_s(t),$ if $r = m,$ and (2) the full problem $\tilde{\boldsymbol{\omega}}(t) = \left[\mathbf{X}^T(t)\mathbf{X}(t)\right]^{-1} \mathbf{X}^T(t)\mathbf{z}(t)$, where $\mathbf{X}(t)$ is $\mathbf{X}_s(t)$ with $r = n$.

A "tilde" on the coefficients is used to indicate that the solution is obtained under $\omega_m = 1$. Wu (1986), stated as Theorem 3 below, gives us a way to write Ramsay's pointwise estimates of $\tilde{\boldsymbol{\omega}} = (\tilde{\omega}_0, \ldots, \tilde{\omega}_{m-1})^T$ as a weighted average of the coefficients $\tilde{\boldsymbol{\omega}}_{\mathbf{s}} = \left(\tilde{\omega}_{0,s}, \ldots, \tilde{\omega}_{(m-1),s}\right)^T$ obtained by resampling. Notice that the weights sum to 1.

Theorem 3 (Wu (1986), p. 1267) *The least squares solution $\tilde{\boldsymbol{\omega}}(t)$ satisfies*

$$\tilde{\boldsymbol{\omega}}(t) = \frac{\sum_r \det\left[\mathbf{X}_s^T(t)\mathbf{X}_s(t)\right] \tilde{\boldsymbol{\omega}}_s(t)}{\sum_r \det\left[\mathbf{X}_s^T(t)\mathbf{X}_s(t)\right]},$$

where r is the size of the subset, $r \geq m$ and $\sum_r$ is the sum over all subsets of size r.

Theorem 3 motivates an estimator for the LDO given in equation (4.1) without the constraint $\omega_m(t) = 1$. Set $r = m$, then the solution is taken to be $\hat{\boldsymbol{\omega}}(t) = \sum_r \boldsymbol{\omega}_s(t)$, where

$$\begin{aligned}\boldsymbol{\omega}_s(t) &= \det\left[\mathbf{X}_s^T(t)\mathbf{X}_s(t)\right]\begin{bmatrix}\tilde{\boldsymbol{\omega}}_s(t)\\ 1\end{bmatrix} \\ &= \left(\det\left[\mathbf{X}_s(t)\right]\right)^2\begin{bmatrix}\mathbf{X}_s^{-1}(t)\mathbf{z}_s(t)\\ 1\end{bmatrix} \\ &= \begin{bmatrix}\det\left[\mathbf{X}_s(t)\right]\text{cofactors}\left[\mathbf{X}_s(t)\right]\mathbf{z}_s(t)\\ \det\left[\mathbf{X}_s^T(t)\mathbf{X}_s(t)\right]\end{bmatrix}.\end{aligned} \tag{5.3}$$

Note that dividing $\hat{\boldsymbol{\omega}}(t)$ through by the leading coefficient $\hat{\omega}_m(t)$ recovers $\tilde{\boldsymbol{\omega}}(t)$ with $\tilde{\omega}_m = 1$. Recall that Ramsay's PDA solution estimates the smooth coefficients of the LDO by minimizing $\sum_{i=1}^n \|Lx_i\|^2 = \sum_{i=1}^n \int_0^T (Lx_i(t))^2 dt$ subject to a penalty term. 5 is a smooth version of $\tilde{\boldsymbol{\omega}}(t)$.

Another solution for $r = m$ instead uses $\hat{\boldsymbol{\omega}}^\gamma(t) = \sum_r \boldsymbol{\omega}_s^\gamma(t)$, where

$$\begin{aligned}\boldsymbol{\omega}_s^\gamma(t) &= \left(\det\left[\mathbf{X}_s^T(t)\mathbf{X}_s(t)\right]\right)^\gamma\begin{bmatrix}\tilde{\boldsymbol{\omega}}_s(t)\\ 1\end{bmatrix} \\ &= \left(\det\left[\mathbf{X}_s(t)\right]\right)^{2\gamma}\begin{bmatrix}\mathbf{X}_s^{-1}(t)\mathbf{z}_s(t)\\ 1\end{bmatrix} \\ &= \begin{bmatrix}\det\left[\mathbf{X}_s(t)\right]^{2\gamma-1}\text{cofactors}\left[\mathbf{X}_s(t)\right]\mathbf{z}_s(t)\\ \left(\det\left[\mathbf{X}_s^T(t)\mathbf{X}_s(t)\right]\right)^\gamma\end{bmatrix},\end{aligned} \tag{5.4}$$

with $1/2 \leq \gamma \leq 1$. This is a variation of an estimator suggested by Wu (1984) to guard against outliers in the multiple regression problem.

The estimator $\boldsymbol{\omega}^\gamma(t)$ implemented here consists of the steps enumerated below.

1. Take m distinct curves $x_{i_1}, \ldots, x_{i_m}$ at a time, compute $K = \binom{n}{m}$ sets of coefficients $\omega_{0k}^\gamma, \omega_{1k}^\gamma, \ldots, \omega_{mk}^\gamma, k = k(i_1, \ldots, i_m) = 1, \ldots, K$ using Equation (5.4).

2. Define the final estimators of the LDO coefficients $\omega_j^\gamma(t) = \sum_{k=1}^K \omega_{jk}^\gamma, j = 0, \ldots, m$. Two cases were considered in the implementation:

   - $\gamma = 1$ to obtain the least squares solution given by $\left(\det\left[\mathbf{X}_s^T(t)\mathbf{X}_s(t)\right]\right)^\gamma \begin{bmatrix} \tilde{\boldsymbol{\omega}}_s(t) \\ 1 \end{bmatrix}$, and
   - $\gamma = 1/2$, a robust version of the latter solution.

3. Smooth the estimated coefficients $\omega_j^\gamma(t)$ with a smoothing parameter denoted by λ_{cv}, chosen by inspection of the cross-validation function $CV(\lambda) = \sum_{i=1}^n \left\| L_\lambda^{(i)} x_i \right\|_2^2$, where $L_\lambda^{(i)} = \omega_{0,\lambda}^{(i)} + \omega_{1,\lambda}^{(i)} D + \cdots + \omega_{m-1,\lambda}^{(i)} D^{m-1} + \omega_{0,\lambda}^{(i)} D^m$ is the LDO computed by leaving the ith curve out. $\left\| L_\lambda^{(i)} x_i \right\|_2^2$ is calculated by trimming 10% from each boundary to minimize the effect of boundary on the choice of the smoothing parameter.

4. Solve the differential equation $\hat{L}x(t) = \omega_0^\gamma x(t) + \omega_1^\gamma Dx(t) + \cdots + \omega_{m-1}^\gamma D^{m-1}x(t) + \omega_m^\gamma D^m x(t) = 0$ to find the basis functions for the null space of the LDO.

5. Use the basis functions to find low-dimensional representations of the curves.

6. Display fits to the data.

Figure 5.1 shows in very general terms the steps in the proposed method, noting the sections in the next chapter where each which will be discussed in more detail, for $m = 2$.

5.2 Differential Equation Solvers

In both the proposed resampling method and in PDA methodology, the step that follows the estimation of the LDO coefficients is the solution of the differential equation $\hat{L}^\gamma x(t) = \omega_0^\gamma x(t) + \omega_1^\gamma Dx(t) + \cdots + \omega_{m-1}^\gamma D^{m-1}x(t) + \omega_m^\gamma D^m x(t) = 0$ to find

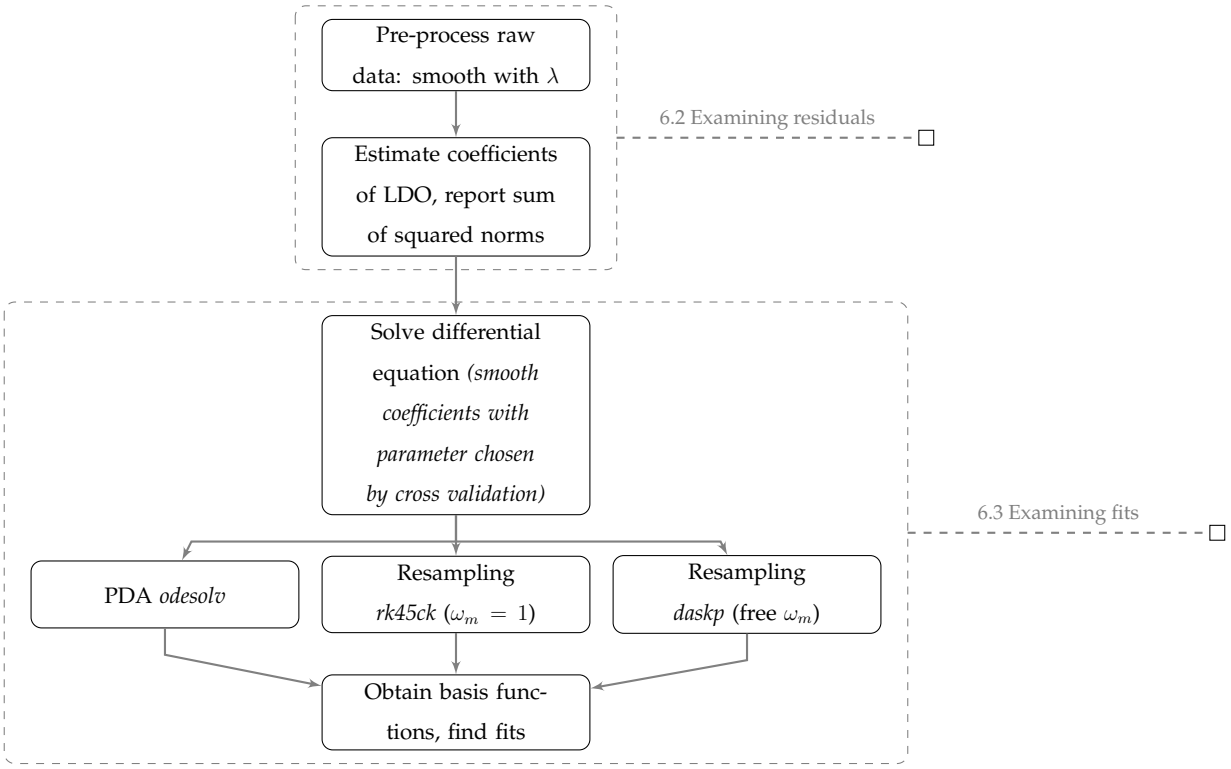


Figure 5.1: Schematic of proposed resampling method.

the basis functions for the null space of the LDO. But the two methods' differing assumptions about the value of ω_m lead to the application of distinct differential equation solvers. This section describes the `R` functions chosen in this implementation to numerically solve the differential equation.

Many natural phenomena, and science and engineering applications are best represented mathematically as (a system of) ordinary differential equations (ODEs)

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y}, \mathbf{z}), \text{ with } \mathbf{y} = (y_1(t), \ldots, y_m(t)) \tag{5.5}$$

where t is the independent variable ($t_0 \leq t \leq t_b$), $\mathbf{y}$ are the dependent differential variables, $\mathbf{z}$ are other variables, and $\mathbf{y}'$ are the first derivatives. To solve an ODE system, an initial condition is required. The initial condition may be an initial value (where values of the initial points $(t_0, \mathbf{y}(t_0))$ are given)

$$\mathbf{y}(t_0) = \mathbf{c},$$

or a boundary value

$$B(\mathbf{y}(t_0), \mathbf{y}(t_b)) = \mathbf{0}.$$

In its most general form, an ODE may be represented implicitly as

$$\mathbf{G}(t, \mathbf{y}, \mathbf{y}', \mathbf{z}) = 0. \tag{5.6}$$

If the variables in (5.8) have additional constraints described by

$$\mathbf{g}(t, \mathbf{y}, \mathbf{z}) = 0,$$

then combining $\mathbf{G}$ and $\mathbf{g}$ we obtain a system of differential algebraic equations (DAE)

$$\mathbf{H}(t, \mathbf{y}, \mathbf{y}', \mathbf{z}) = 0,$$

Most popular numerical techniques to solve initial value ODEs may be classified into Runge-Kutta or linear multistep methods. These methods are in general robust, highly efficient and accurate, but require that the ODE system be written

explicitly as in Equation (5.5). Higher order ordinary differential equation systems can be easily transformed into explicit ODEs by introducing new variables, as described next. Given an ordinary differential equation of order m

$$y^{(m)} = g(t, y, y', ..., y^{(m-1)},$$

introduce a new variable for each derivative, leading to the system (where $y_1 \equiv y$)

$$\begin{aligned} y_1' &= y_2 \\ &\vdots \\ y_{m-1}' &= y_m \\ y_m' &= f(t, y_1, y_2, \ldots, y_m). \end{aligned} \tag{5.7}$$

Now, recall our original notation where the functional data is represented by $x_i(t), i = 1 \ldots, n$. As stated earlier, to find the basis for the null space of the forcing function $Lx_i(t) = 0$, we need the solution to the differential equation

$$Lx(t) = \omega_0 x(t) + \omega_1 Dx(t) + \cdots + \omega_{m-1} D^{m-1} x(t) + \omega_m D^m x(t) = 0.$$

To illustrate the process of transforming the equation into a system of ordinary differential equations, let $m = 2$, and set $x_1 = x$ and $x_2 = x'$. The differential equation is transformed into the system

$$\begin{aligned} \omega_2(t)x_2'(t) + \omega_1(t)x_2(t) + \omega_0(t)x_1(t) &= 0 \\ x_1'(t) - x_2(t) &= 0, \end{aligned}$$

that may be written as an implicit system of ODE's,

$$\mathbf{G}(t, \mathbf{x}, \mathbf{x}') = 0. \tag{5.8}$$

Re-writing the system in matrix form as

$$\begin{pmatrix} 1 & 0 \\ 0 & \omega_2(t) \end{pmatrix} \begin{pmatrix} x_1'(t) \\ x_2'(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\omega_0(t) & -\omega_1(t) \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t), \end{pmatrix} \quad \text{or}$$

$$\mathbf{M(t)x'}(t) = \mathbf{F}(t, \mathbf{x}), \tag{5.9}$$

it can be transformed into an explicit ODE system

$$\mathbf{x'(t)} = \mathbf{M^{-1}}(t)\mathbf{F}(t, \mathbf{x}), \quad \text{whenever } \omega_2(t) \neq 0, \tag{5.10}$$

but we reiterate that we wish to maintain $\omega_2(t)$ as the leading coefficient. However, in Chapter 7 and for the purpose of comparing solvers and methodologies, we will consider both the implicit and explicit forms of the ODE system (equations 5.8 and 5.10, respectively).

Ramsay's PDA assumes that $\omega_m = 1$, so $Lx(t) = 0$ can be readily written as an explicit system of ODE's, and solved using Runge-Kutta methods. PDA software calls the function `odesolv` (in the package `fda`), which implements an embedded Runge-Kutta method using the Cash-Karp parameters and an adaptive stepsize based on the criteria described in Press et al. 1992, p. 717. In the `odesolv` documentation, users are encouraged to use the package `deSolve` in future applications.

The package `deSolve` includes a number of functions to solve differential equations. The wrapper `ode` with parameter `method=rk45ck` (to which we will refer as simply `rk45ck` in this document) implements the same embedded Runge-Kutta method as `odesolv`, except for the adaptive stepsize algorithm, which is based on Press et al. 2007, p. 914. This difference hindered the process of comparing the two methods, because `odesolv` and `rk45ck` can give slightly different results. Keeping in mind this caveat, `rk45ck` is applied in the resampling method only when we assume that $\omega_m = 1$.

The function `daspk` in `deSolve` solves a system of differential algebraic equations (DAE) as expressed in equation 5.8. In our case, the ODE system does not

include any additional algebraic constraints, which is specified in the call to the function. The initial conditions for both `daspk` and `rk45ck` were the mean of the curves and the mean of the derivatives at $t = 0$.

Chapter 6

Implementation of Resampling Method, m=2

6.1 Introduction

In this chapter we present results obtained by applying the resampling method and current PDA methodology to two data sets: simulated data and the lip data provided in the `R fda` library. Both data sets were further modified by replacing one curve with an obvious outlier. In this context, an outlier is any functional observation that is not in the span of the null space of the LDO.

Ramsay assumes that the lip data is best described by a second-order LDO ($m = 2$), and we adhere to that assumption. The simulated data, also described by a second-order LDO, was obtained as follows: given the LDO

$$L = 10I + 2D + D^2,$$

with analytical solutions

$$u_1(t) = e^{-t}\cos 3t \quad \text{and} \quad u_2(t) = e^{-t}\sin 3t,$$

$u_1(t)$ and $u_2(t)$ were used to generate 1000 independent sets of 20 curves

$$\begin{aligned} x_i &= a_{1,i}u_1(t) + a_{2,i}u_2(t) \\ &= e^{-t}(a_{1,i}\cos 3t + a_{2,i}\sin 3t) \end{aligned}$$

with randomly chosen coefficients

$$\begin{pmatrix} a_{1,i} \\ a_{2,i} \end{pmatrix} \sim N_2 \left(\begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix} \right)$$

for $i = 1, \ldots 20$ and 76 equally spaced points $t \in [0, 1.5]$.

Several modifications of this data set were considered:

1. Contaminate each curve with iid noise $\epsilon \sim N(0, \sigma^2)$, $\sigma = 0.3$:

$$x_i = e^{-t}(a_{1,i} \cos 3t + a_{2,i} \sin 3t + \sigma\epsilon, i = 1, \ldots 20.$$

2. Replace one of the curves in each set of 20 by an outlier, given by $x(t) = 6 \sin 10t$. In this case, when the sum of the squared norms of the forcing function $\sum_{i=1}^{n} \|Lx_i\|^2$ is reported, the outlier is not included.

3. Contaminate with noise and add an outlier as just described.

Figure 6.1 shows one realization of the simulated data without noise and the registered lip data, with and without outliers. The lip data was registered after introducing the outlier, using the same marks included in the `fda` R library. In the implementation of the resampling method for $m = 2$, the following will be explored: the effect of pre-processing of the data by smoothing, the effect of smoothing the estimated coefficients, the effect of noise and outliers on the basis functions and the fitted curves.

All computations were carried out in the R language and environment. We used several lines of code from the `lip.R` script provided in Ramsay's R library `fda`. The R libraries `MASS`, `deSolve` and `limSolve` are also required in our code. The script used to reproduce Ramsay's results was `lip.R` with a few modifications to adjust to the latest version of the `fda` library.

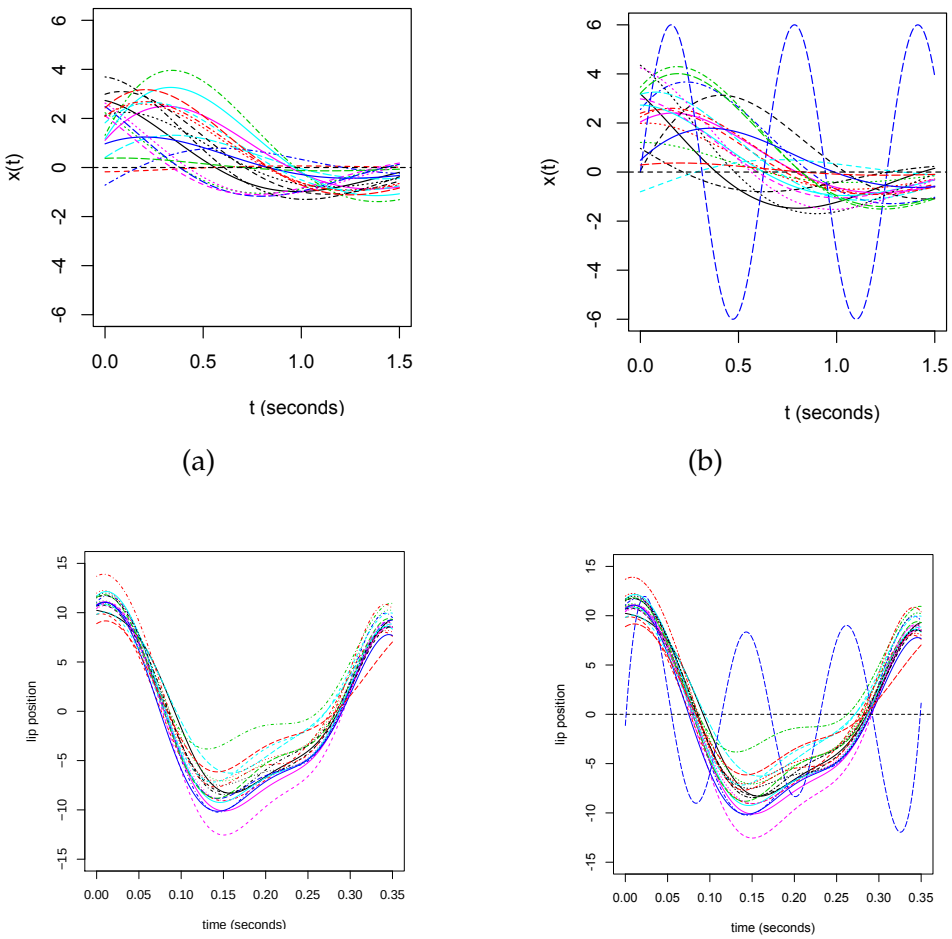


Figure 6.1: (a) One realization of simulated data, with noise $\sigma = 0$; (b) Same realization of simulated data with an outlier introduced. (c) The registered lip data; (d) The registered lip data with an outlier introduced.

6.2 Examining Forcing Functions

6.2.1 Simulated Data

The first step in the implementation is to preprocess the raw data by converting it into a functional data object, which implies a certain degree of smoothing (chosen by the user). Two levels of smoothing ($\lambda = 10^{-12}$ and $\lambda = 10^{-9}$) were applied to the simulated data during preprocessing; the sum of squared norms and fits are reported. Then, estimates of the coefficients of the LDO were obtained by applying the proposed resampling method and PDA methodology. The smoothing selected for implementation of the PDA methodology was chosen to be minimal in order to better compare with the proposed methodology. PDA is using a spline basis of dimension 150 (146 knots, i.e., 150 elements in the spline basis or order 4), whereas the proposed method does not impose any smoothness on the pointwise solution of the coefficients for the LDO. The sum of squared norms $\sum_{i=1}^{n} \|Lx_i\|^2$ was computed for each set of curves. The mean over 1000 simulations of $\sum_{i=1}^{n} \|Lx_i\|^2$ are reported in Tables 6.1a, 6.1b and 6.2. To compute the standard error, the 1000 simulations were split into 4 blocks of 250 sets, and the standard deviation of the 4 means was divided by $\sqrt{4} = 2$.

In Tables 6.1a and 6.1b, the PDA estimates of ω_0 and ω_1 where multiplied by ω_2^γ to compare the squared norms of the forcing functions on the same scale. Focus first on Tables 6.1a and 6.1b when $\sigma = 0$: the $\sum_{i=1}^{n} \|Lx_i\|^2$ estimated by simulation for the resampling method, regardless of the value of γ, is much smaller than that obtained for PDA. Our explanation for this was mentioned above, namely, the proposed resampling algorithm provides for a more flexible curve-fitting method because it is a pointwise least squares solution for the coefficients of the LDO, whereas the PDA method is a least squares solution constrained by the dimension of the B-spline basis. Notice that for the resampling method ($\sigma = 0$), $\sum_{i=1}^{n} \|Lx_i\|^2$ is not affected by the level of smoothing applied in the preprocessing step.

However, when noise is added to the data ($\sigma = 0.3$), the $\sum_{i=1}^{n} \|Lx_i\|^2$ estimated by simulation are very close for PDA and for the resampling method, regardless of the value of γ; see Tables 6.1a and 6.1b. This indicates that the regularization step incorporated into PDA, in the presence of noise, no longer results in a loss of information. Alternatively, in the presence of noise, the flexible resampling method based on a pointwise estimation procedure loses its advantage over the constrained fits provided by PDA. Note that in the presence of noise ($\sigma = 0.3$), robust resampling with $\gamma = 1/2$ suffers a slight loss of efficiency when compared to PDA. Notice that for the resampling method ($\sigma = 0.3$), $\sum_{i=1}^{n} \|Lx_i\|^2$ is reduced with a higher level of smoothing applied to the data in the preprocessing step.

Focus now on Table 6.2, which considers the effect of a single outlier on the resampling method with $\gamma = 1$ and $\gamma = 1/2$. Each set of coefficient estimates was multiplied by ω_2^{γ} from the other set of coefficients in order to compare the squared norms of the forcing functions on the same scale. Recall that the outlier curve is not included in the sum of the squared norm of the forcing functions. The $\sum_{i=1}^{n} \|Lx_i\|^2$ estimated by simulation with $\gamma = 1/2$ is smaller than for $\gamma = 1$. This is consistent with the idea that the estimates obtained with $\gamma = 1/2$ are less sensitive to outliers. In addition, we investigated the effect of adding another smoothing step to the proposed resampling method: smoothing the pointwise estimated coefficients of the LDO as implemented in `smooth.basisPar`. The estimated coefficients were lightly smoothed with 10^{-12}, and the simulation results were identical to those given in Table 6.2.

The smoothed coefficients are considered because the extra smoothing step helped the differential equation solvers converge, when using the non-smoothed pointwise estimates of the coefficients did not. In addition to accepting a user-provided value for the smoothing parameter, a cross-validation criteria was implemented to obtain a data-based parameter to smooth the pointwise estimated coefficients. Cross-validation results are illustrated in the next section.

Table 6.1a: Simulated data: Sum of squared norms of the forcing functions, proposed resampling method with $\gamma = 1$ and PDA. PDA estimates were multiplied by ω_2^γ to compare the squared norms of the forcing functions on the same scale. Standard error of the simulated results is reported in parenthesis.

	Sum of squared norms $\sum_{i=1}^{n} \|Lx_i\|^2$			
Method	Simulated data, $\sigma = 0$		Simulated data, $\sigma = 0.3$	
	(preprocess $\lambda = 10^{-12}$)	(preprocess $\lambda = 10^{-9}$)	(preprocess $\lambda = 10^{-12}$)	(preprocess $\lambda = 10^{-9}$)
PDA	$1.38 \times 10^{-7}(5.8 \times 10^{-10})$	$1.32 \times 10^{-8}(5.3 \times 10^{-11})$	$2.58 \times 10^{8}(1.88 \times 10^{6})$	$6.6 \times 10^{6}(7.1 \times 10^{4})$
Resampling ($\gamma = 1$)	$1.9 \times 10^{-18}(5.0 \times 10^{-20})$	$1.9 \times 10^{-18}(3.6 \times 10^{-20})$	$2.56 \times 10^{8}(1.92 \times 10^{6})$	$6.6 \times 10^{6}(7.1 \times 10^{4})$

Table 6.1b: Simulated data: Sum of squared norms of the forcing functions, proposed resampling method with $\gamma = 1/2$ and PDA. PDA estimates were multiplied by ω_2^γ to compare the squared norms of the forcing functions on the same scale. Standard error of the simulated results is reported in parenthesis.

	Sum of squared norms $\sum_{i=1}^{n} \|Lx_i\|^2$			
Method	Simulated data, $\sigma = 0$		Simulated data, $\sigma = 0.3$	
	(preprocess $\lambda = 10^{-12}$)	(preprocess $\lambda = 10^{-9}$)	(preprocess $\lambda = 10^{-12}$)	(preprocess $\lambda = 10^{-9}$)
PDA	$9.30 \times 10^{-8}(3.8 \times 10^{-10})$	$1.30 \times 10^{-8}(5.3 \times 10^{-11})$	$1.44 \times 10^{8}(1.08 \times 10^{6})$	$3.79 \times 10^{6}(3.6 \times 10^{4})$
Resampling ($\gamma = 1/2$)	$1.0 \times 10^{-18}(2.6 \times 10^{-20})$	$9.82 \times 10^{-19}(1.9 \times 10^{-20})$	$1.47 \times 10^{8}(1.15 \times 10^{6})$	$3.9 \times 10^{6}(3.7 \times 10^{4})$

Table 6.2: Simulated data: sum of squared norms of the forcing functions for the proposed resampling method when an outlier is included. Outlier curve is not included in the sum. Each set of coefficient estimates was multiplied by ω_2^γ from the other set of coefficients in order to compare the squared norms of the forcing functions on the same scale. Standard error of the simulated results is reported in parenthesis.

Sum of squared norms $\sum_{i=1}^{n} \|Lx_i\|^2$, resampling method		
Outlier included	Simulated data, $\sigma = 0$ preprocess $\lambda = 10^{-12}$	Simulated data, $\sigma = 0.3$ preprocess $\lambda = 10^{-9}$
$\gamma = 1$	$5.2 \times 10^5 (2.7 \times 10^3)$	$2.1 \times 10^8 (8.8 \times 10^5)$
$\gamma = 1/2$	$4.0 \times 10^5 (7.2 \times 10^2)$	$1.3 \times 10^8 (9.1 \times 10^5)$

6.2.2 Lip Data

The pre-processing smoothing applied to the data was $\lambda = 10^{-12}$, as done in Ramsay's original demonstration of PDA for the lip data. The ratio of the sum of squared norms $\sum_{i=1}^{n} \|Lx_i\|^2$ for Ramsay PDA (using $\lambda = 0$ and 146 knots so that we have 150 elements in the basis) to each of the resampling estimates of the LDO is reported in Table **??**. The term contributed by the outlier curve was not included in the sum of squared norms. The PDA estimates of ω_0 and ω_1 were multiplied by ω_2^γ to compare the squared norms on the same scale. Focusing on row 1, the fact that the ratio of the sum of squared norms is 1 supports Theorem 3 that states that using $\gamma = 1$ with the proposed resampling method yields the pointwise least squares solution $\tilde{\boldsymbol{\omega}}$. Focusing on column 1, an analysis of the original lip data suggests there is a small loss in efficiency when using a robust version of resampling ($\gamma = 1/2$). An analysis of the lip data with an outlier suggests that resampling with $\gamma = 1/2$ can lead to substantial gains in efficiency as compared to the current PDA methodology (ratio is 1.63).

Table 6.3: Ratio of sum of squared norms of the forcing functions: PDA/proposed. Outlier curve not included in sum. PDA estimates were multiplied by ω_2^γ to compare the squared norms of the forcing functions on the same scale.

Proposed Method	Ratio of sum of squared norms: PDA/proposed	
	Original lip data	with outlier
$\gamma = 1$	1.00	1.00
$\gamma = 1/2$	0.98	1.63

6.3 Examining Basis Functions and Fits

6.3.1 Simulated Data

Since the basis functions of the LDO that annihilates the simulated data curves are known ($u_1(t) = e^{-t}\cos 3t$ and $u_2(t) = e^{-t}\sin 3t$), first we verified that correct results were obtained by `daskp` and `rk45ck`. Given that the basis functions of an LDO are not unique, we checked whether the estimated $\{\hat{u}_1, \hat{u}_2\}$ and the known $\{u_1, u_2\}$ basis functions span the same space. The resampling method was applied to one set of 20 curves with no noise ($\sigma = 0$); each of the functions `daskp` and `rk45ck` returned a set of estimated basis functions. Then the known basis functions were fit to each of the sets of estimated basis functions using ordinary least squares regression. Figure 6.2 displays the regression residuals (note that they are virtually zero) and fits in the case of the estimated functions obtained by `daskp`; the estimated basis functions obtained by `rk45ck` produced identical fits and residuals. Regressing the basis functions in the reverse order (estimated to known), resulted in fits identical to those shown in Figure 6.2, and residuals that were also close to zero. In summary, we found that both sets of basis functions $\{u_1, u_2\}$ and $\{\hat{u}_1, \hat{u}_2\}$ span the same space when $\sigma = 0$, whether `daskp` or `rk45ck`

was used.

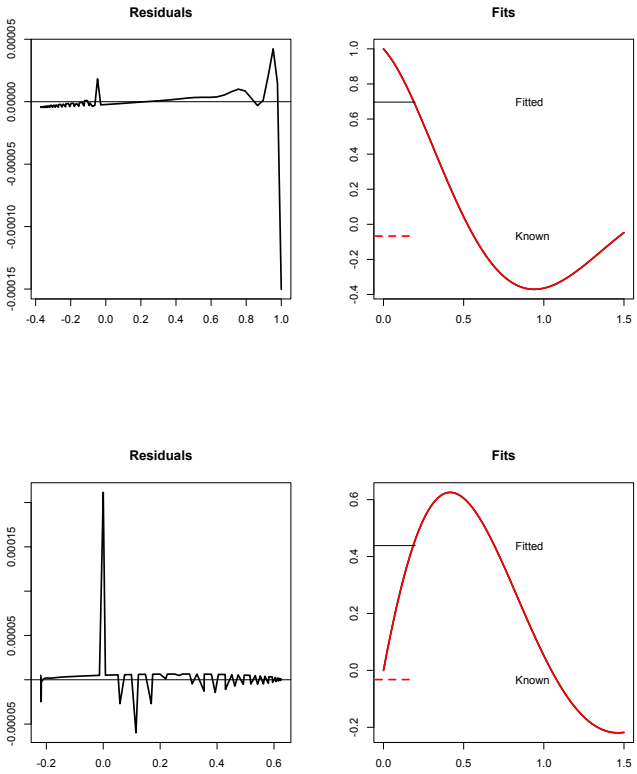


Figure 6.2: Residuals and fits obtained by regressing the known solutions $\{u_1, u_2\}$ on estimated basis functions $\{\hat{u_1}, \hat{u_2}\}$. u_1 on first row, u_2 on second row.

While exploring the simulated data with noise ($\sigma = 0.3$), we found that smoothing the pointwise estimated coefficients of the LDO was needed to achieve convergence of the differential equation solvers. A leave-one-out cross-validation criteria is used to identify values of the smoothing parameter worth trying for inspection of the fits. From this point forward in this document, the smoothing parameter λ_{cv} denotes the value of the smoothing parameter chosen after inspecting local and global minima of the cross-validation curve, as well as regions of plateaus. The cross-validation plots for one set of 20 simulated data curves pre-processed

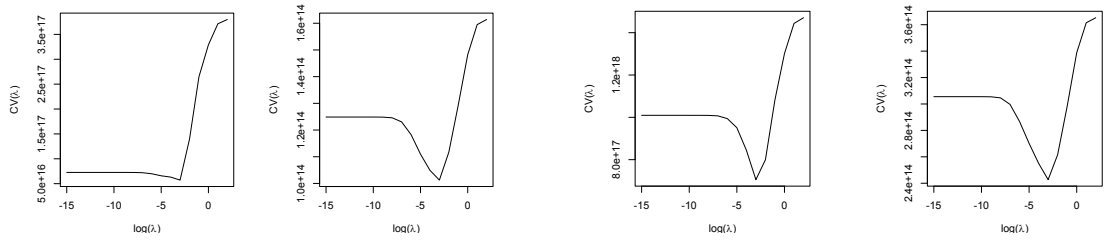
with smoothing $\lambda = 10^{-12}$ and $\lambda = 10^{-9}$ are displayed in Figures 6.3a and 6.3b, respectively.

When a preprocess smoothing of $\lambda = 10^{-12}$ was applied, several values of the smoothing parameter along the plateau on the interval $(10^{-15}, 10^{-7})$ and the global minimum 10^{-3} were examined. The smoothing parameter $\lambda_{cv} = 10^{-12}$ was chosen after visually inspecting the fits produced by the basis functions obtained under $\gamma = 1$ (in this case, the global minimum 10^{-3} produced very poor fits, due to over-smoothing of the coefficients). For $\gamma = 1/2$, the global minimum $\lambda_{cv} = 10^{-3}$ was selected. The same process was followed with the set of curves with an outlier introduced, obtaining the same selections for the smoothing parameter: $\lambda_{cv} = 10^{-12}$ for $\gamma = 1$, and $\lambda_{cv} = 10^{-3}$ for $\gamma = 1/2$.

When the preprocess smoothing applied was $\lambda = 10^{-9}$, the cross-validation minima produced the best fits. The values selected for the set of 20 curves were $\lambda_{cv} = 10^{-3}$ for $\gamma = 1$, and $\lambda_{cv} = 10^{-1}$ for $\gamma = 1/2$. For the set of curves with an outlier, $\lambda_{cv} = 10^{-3}$ was selected for both $\gamma = 1$ and $\gamma = 1/2$.

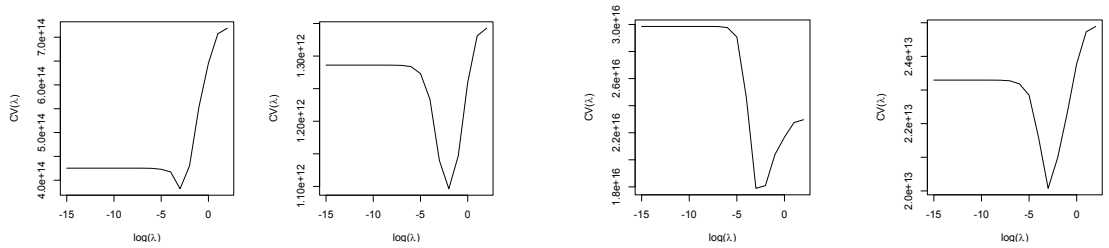
Fits to the data were found by performing ordinary least squares regression on the estimated basis functions obtained by `daspk`. Results identical to those described below were obtained regressing on the estimated basis functions obtained by `rk45ck`. Figure 6.4a and Figure 6.4b show one realization of the simulated data with noise ($\sigma = 0.3$), and fits obtained by PDA and the resampling method with $\gamma = 1/2$ and $\gamma = 1$.

In Figure 6.4a the data was preprocessed with $\lambda = 10^{-12}$ and the pointwise estimated coefficients of the LDO were smoothed with $\lambda_{cv} = 10^{-12}$ when $\gamma = 1$, and with $\lambda_{cv} = 10^{-3}$ when $\gamma = 1/2$. The fits produced by PDA follow the same trend as those produced by the proposed resampling method with $\gamma = 1$, but the PDA fits are smoother because a B-spline basis is used to estimate the coefficients, while the resampling method produces pointwise estimates for the coefficients of the LDO.



(a) $\gamma = 1$ (b) $\gamma = 1/2$ (c) $\gamma = 1$ (d) $\gamma = 1/2$

Figure 6.3a: (a) and (b) Cross-validation curves for one realization of simulated data, $\sigma = 0.3$, preprocessed with $\lambda = 10^{-12}$; (c) and (d) with an outlier.



(e) $\gamma = 1$ (f) $\gamma = 1/2$ (g) $\gamma = 1$ (h) $\gamma = 1/2$

Figure 6.3b: (e) and (f) Cross-validation curves for one realization of simulated data, $\sigma = 0.3$, preprocessed with $\lambda = 10^{-9}$; (g) and (h) with an outlier.

The data was preprocessed with $\lambda = 10^{-12}$ in Figure 6.4a, and with $\lambda = 10^{-9}$ Figure 6.4b; the pointwise estimated coefficients of the LDO were smoothed with λ_{cv} as described above. Comparing Figures 6.4a and 6.4b, we see that preprocessing the data yields much improved fits on noisy data; the fits in Figure 6.4b follow the data curves much tighter than the fits in Figure 6.4a. Figure 6.5a and Figure 6.5b are analogous to Figures 6.4a and 6.4b except that a curve has been replaced with an outlier. The estimated coefficients were smoothed with $\lambda_{cv} = 10^{-3}$. In Figure 6.5a, again the PDA fits follow the same trend as the resampling fits with $\gamma = 1$, but all fits are affected by the outlier. In Figure 6.5b the fits from the resampling method for $\gamma = 1$ and $\gamma = 1/2$ follow the same trend and track the curves closer than the PDA fits. In this case it seems that PDA is more sensitive to the outlier than the pointwise approach of resampling.

The effect of pre-processing the data with $\lambda = 10^{-9}$ as reported in Table 6.2 is more evident when comparing Figures 6.5a and 6.5b: smoothing with $\lambda = 10^{-12}$ in the pre-processing step results in very noisy fits, while pre-processing with $\lambda = 10^{-9}$ produces fits closer to the curves. The entries in Table 6.2 also suggest that resampling with $\gamma = 1/2$ should have slightly improved fits as compared with resampling with $\gamma = 1$. Examining Figure 6.5b, the estimates corresponding to $\gamma = 1/2$ are indeed tracking the curves slightly closer than the estimates corresponding to $\gamma = 1$, and show a larger improvement over the PDA estimates for all curves with the exception of the outlier.

6.3.2 Lip Data

The value of the smoothing parameter to smooth the estimated LDO coefficients was selected after examining a few values along the plateaus and the minima of the cross-validation curves in Figure 6.6. In each of the four cases, the global minimum produced better fits. $\lambda_{cv} = 10^{-8}$ was used to smooth the estimated

coefficients of the LDO for the registered lip data with $\gamma = 1$ and $\gamma = 1/2$, and for the registered lip data with an outlier and $\gamma = 1/2$. For the registered lip data with an outlier and $\gamma = 1$, $\lambda_{cv} = 10^{-9}$ was selected. Even though smoothing was not needed to help the differential equation solvers converge when $\gamma = 1$, the estimated coefficients of the LDO were smoothed with the cross-validation smoothing parameter λ_{cv}. The effect of applying this level of smoothing can be better observed in the case of the lip data with an outlier. Figure 6.7 shows that the estimates corresponding to $\lambda_{cv} = 10^{-9}$ reproduce the shape of the data curves better than the estimates corresponding to $\lambda = 0$, with the exception of the outlier.

The null basis functions found by the resampling method (with the solver `rk45ck`) and by PDA (with the solver `odesolv`) applied to the lip data with and without an outlier are shown in Figure 6.8. We see that the outlier had a larger effect on the basis functions found by PDA than in those found by the resampling method, but the two criteria of the adaptive step-size in the solvers may account for this difference.

Finally, we perform ordinary least squares regression on the estimated basis functions to find estimates for the 20 curves in the lip data. Figure 6.9a shows the lip data with fits obtained by PDA methodology, and by the resampling solution with $\gamma = 1/2$ and $\gamma = 1$. As suggested by Table **??**, the fits produced by PDA and resampling with $\gamma = 1/2$ and $\gamma = 1$ are quite close, although PDA fits are a little better for a few curves. The different implementation of the adaptive step-size algorithms in the differential equation solvers may account for the differences seen in the basis functions, and thus in the fits. The fits to the lip data with an outlier are shown on Figure 6.9b. In this case, the fits produced by the resampling method with $\gamma = 1$ and by PDA are similar, while the fits obtained by the resampling method with $\gamma = 1/2$ pull away from the other two and do not follow a downward trend on the boundaries.

6.4 Summary of Findings

The results in this chapter show that slight smoothing of the estimated coefficients of the LDO has a negligible effect on the sum of squared norms, but helps to achieve convergence of the differential equation solvers `daspk` and `rk45ck`. Even in the case when the differential solvers converge, smoothing of the data improves the fits obtained by the proposed method. Smoothing the estimates of the LDO coefficients is another regularization step that improves the fits. The smoothing paramater λ_{cv} is chosen by examining the local and global minima and the plateau regions of the cross-validation curves, and inspecting the fits produced by them. In most cases, smoothing the estimates of the LDO coefficients with parameter λ_{cv} yielded better fits in the presence of an outlier and noise. Tables 6.2 and **??**, as well as Figure 6.9b support the statement that the resampling method with $\gamma = 1/2$ provides better fits to the data sets with an outlier, although the improvement is modest.

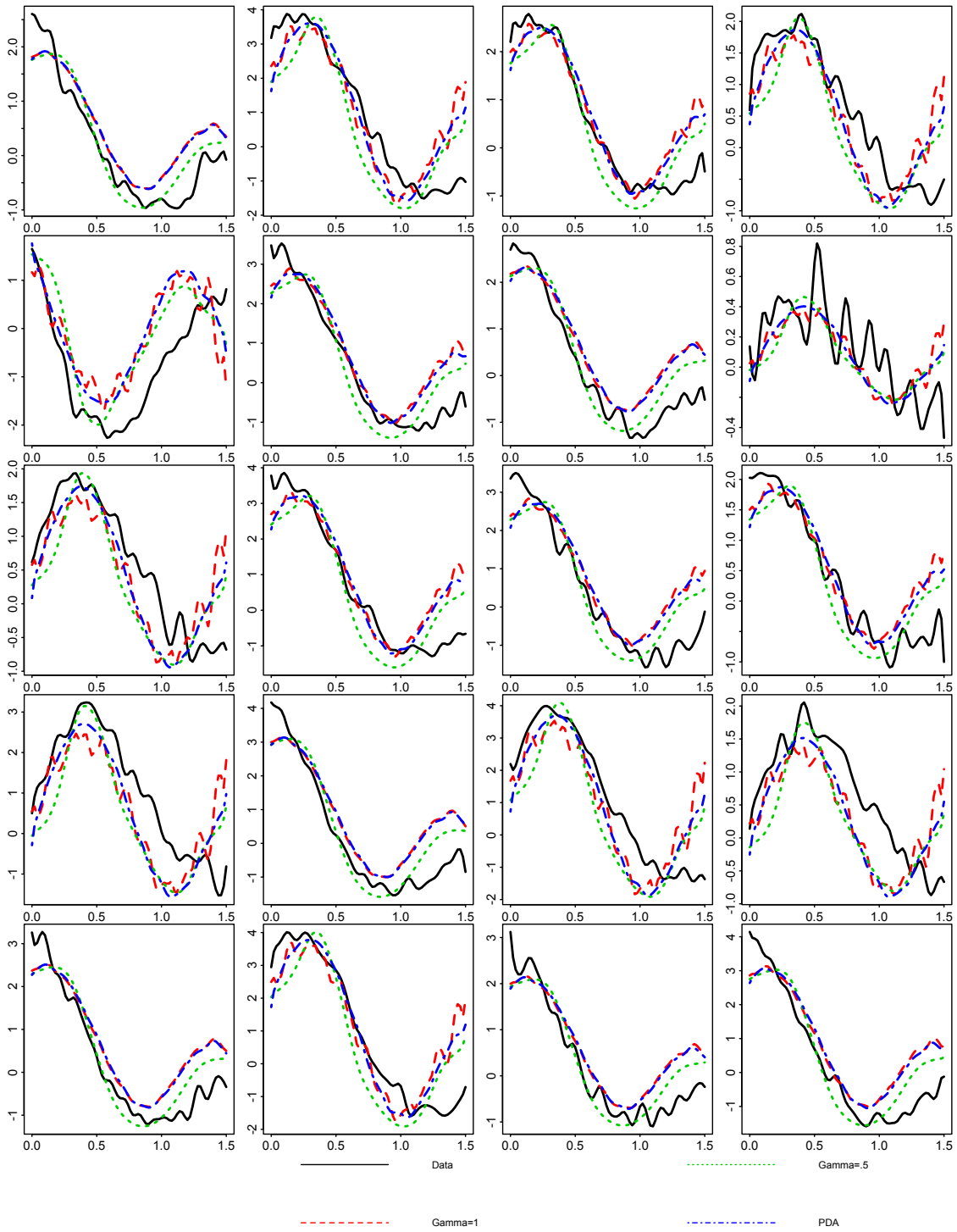


Figure 6.4a: Fits to one realization of simulated data, $\sigma = 0.3$, preprocessed with $\lambda = 10^{-12}$; estimated coefficients of the LDO smoothed with $\lambda_{cv} = 10^{-12}$ when $\gamma = 1$, and with $\lambda_{cv} = 10^{-3}$ when $\gamma = 1/2$.

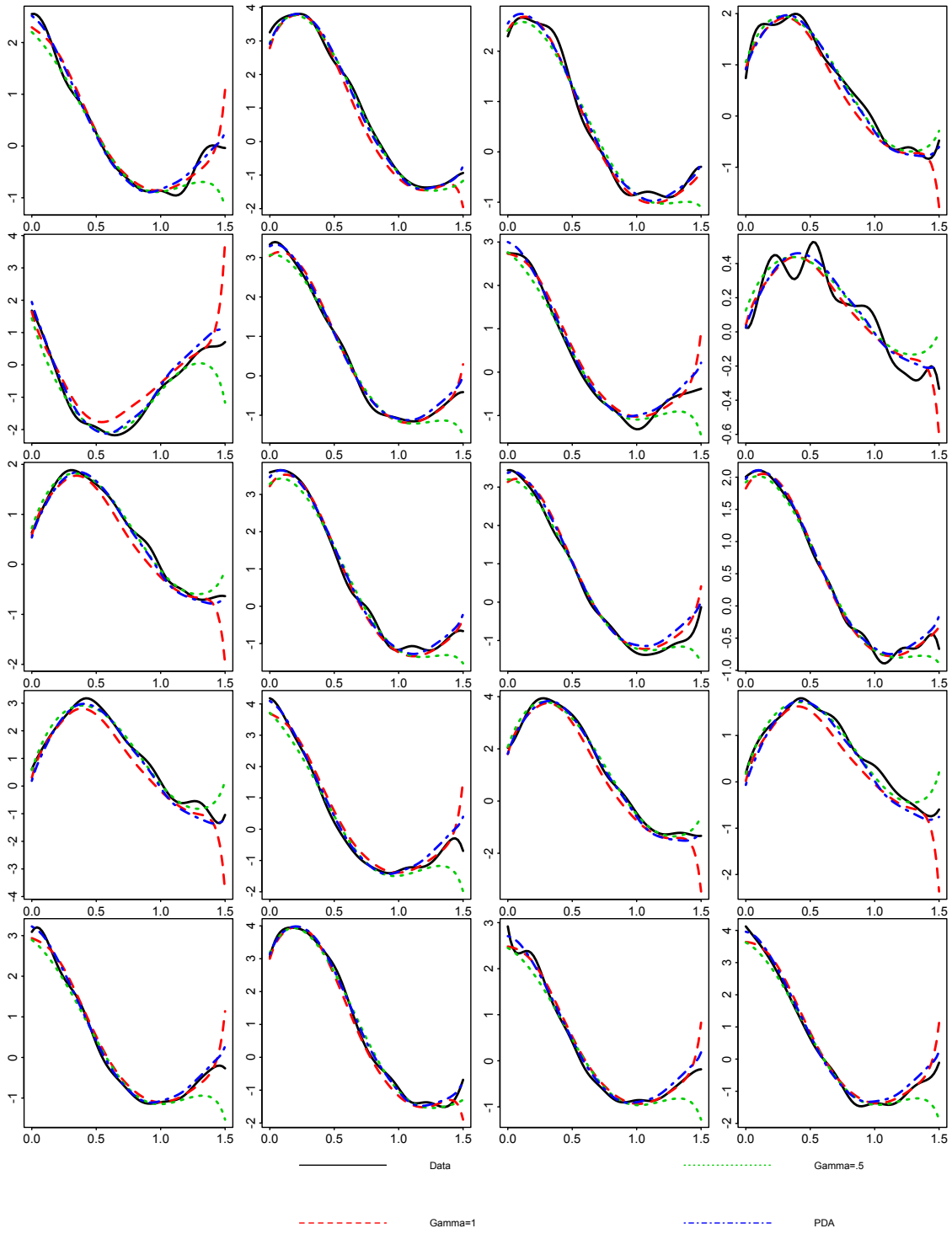


Figure 6.4b: Fits to one realization of simulated data, $\sigma = 0.3$, preprocessed with $\lambda = 10^{-9}$; estimated coefficients of the LDO smoothed with with $\lambda_{cv} = 10^{-3}$ when $\gamma = 1$ and with $\lambda_{cv} = 10^{-1}$ when $\gamma = 1/2$.

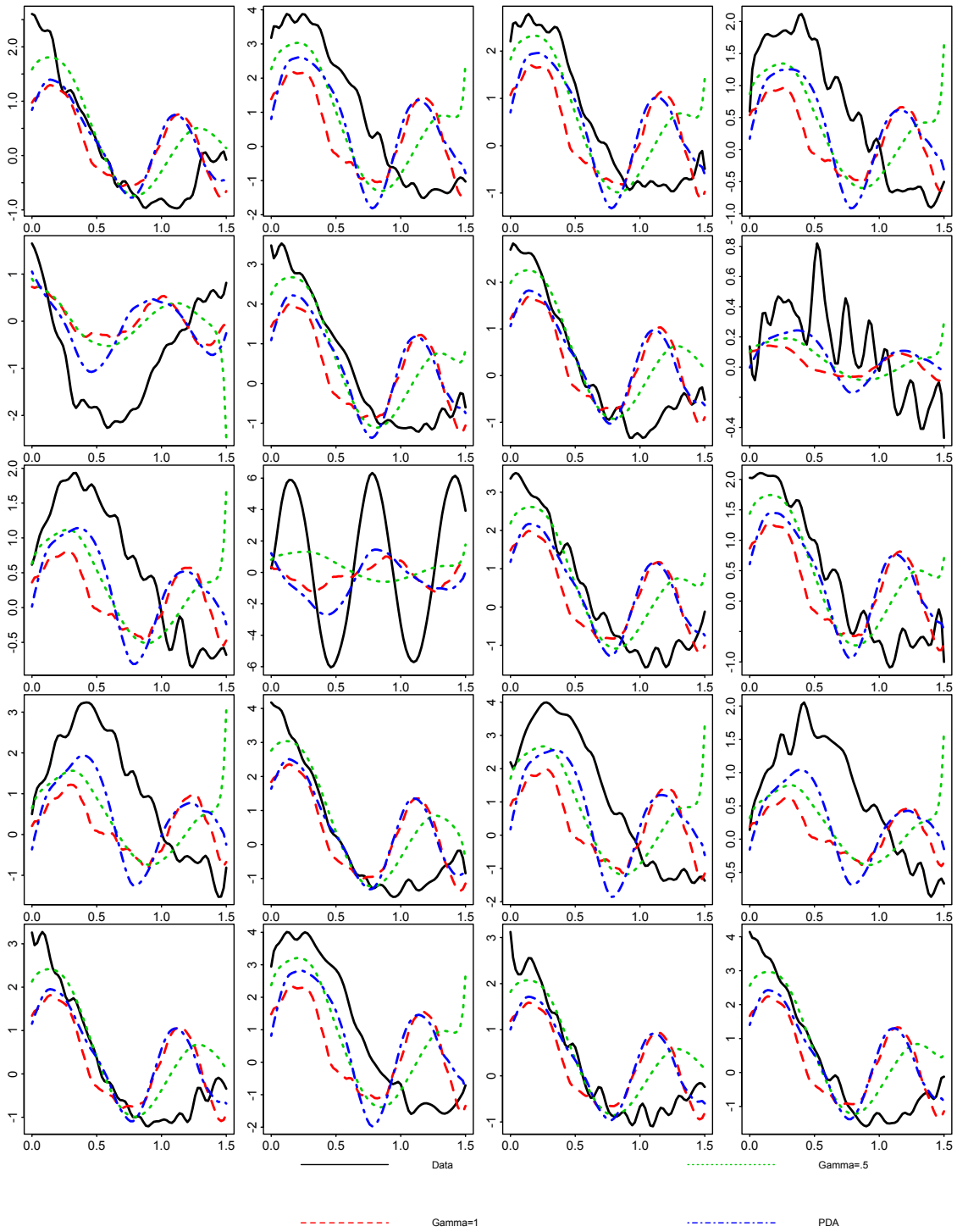


Figure 6.5a: Fits to one realization of simulated data with outlier, $\sigma = 0.3$, pre-processed with $\lambda = 10^{-12}$, estimated coefficients of the LDO smoothed with $\lambda_{cv} = 10^{-3}$.

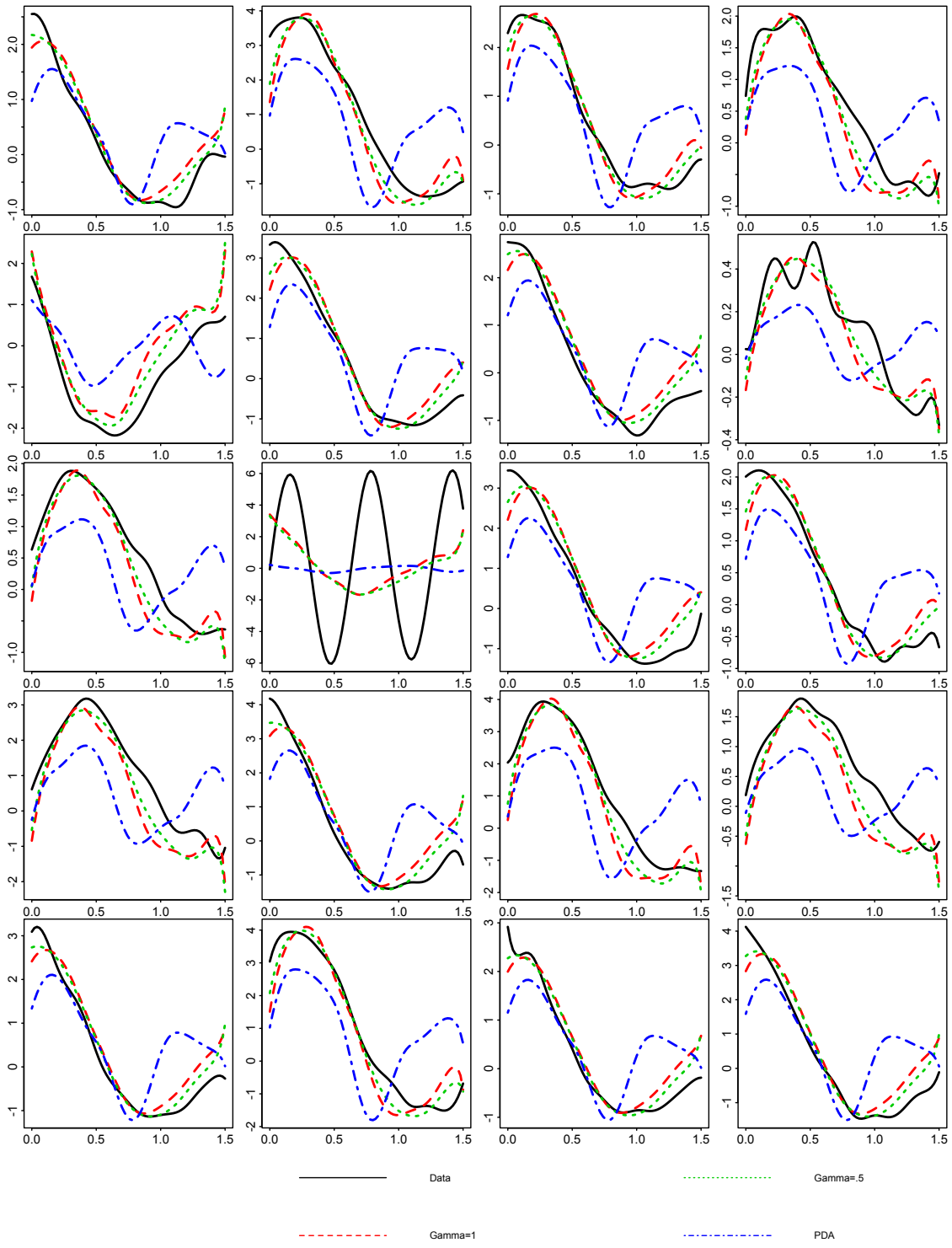
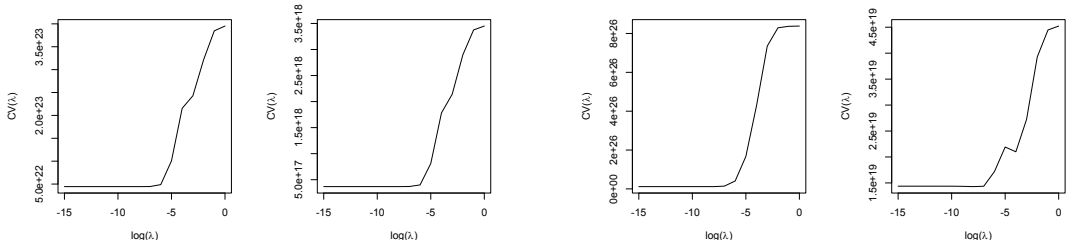


Figure 6.5b: Fits to one realization of simulated data with outlier, $\sigma = 0.3$, preprocessed with $\lambda = 10^{-9}$; estimated coefficients of the LDO smoothed with $\lambda_{cv} = 10^{-3}$.



(a) $\gamma = 1$ (b) $\gamma = 1/2$ (c) $\gamma = 1$ (d) $\gamma = 1/2$

Figure 6.6: (a) and (b) Cross-validation curves for lip data preprocessed with $\lambda = 10^{-12}$; (c) and (d) with an outlier.

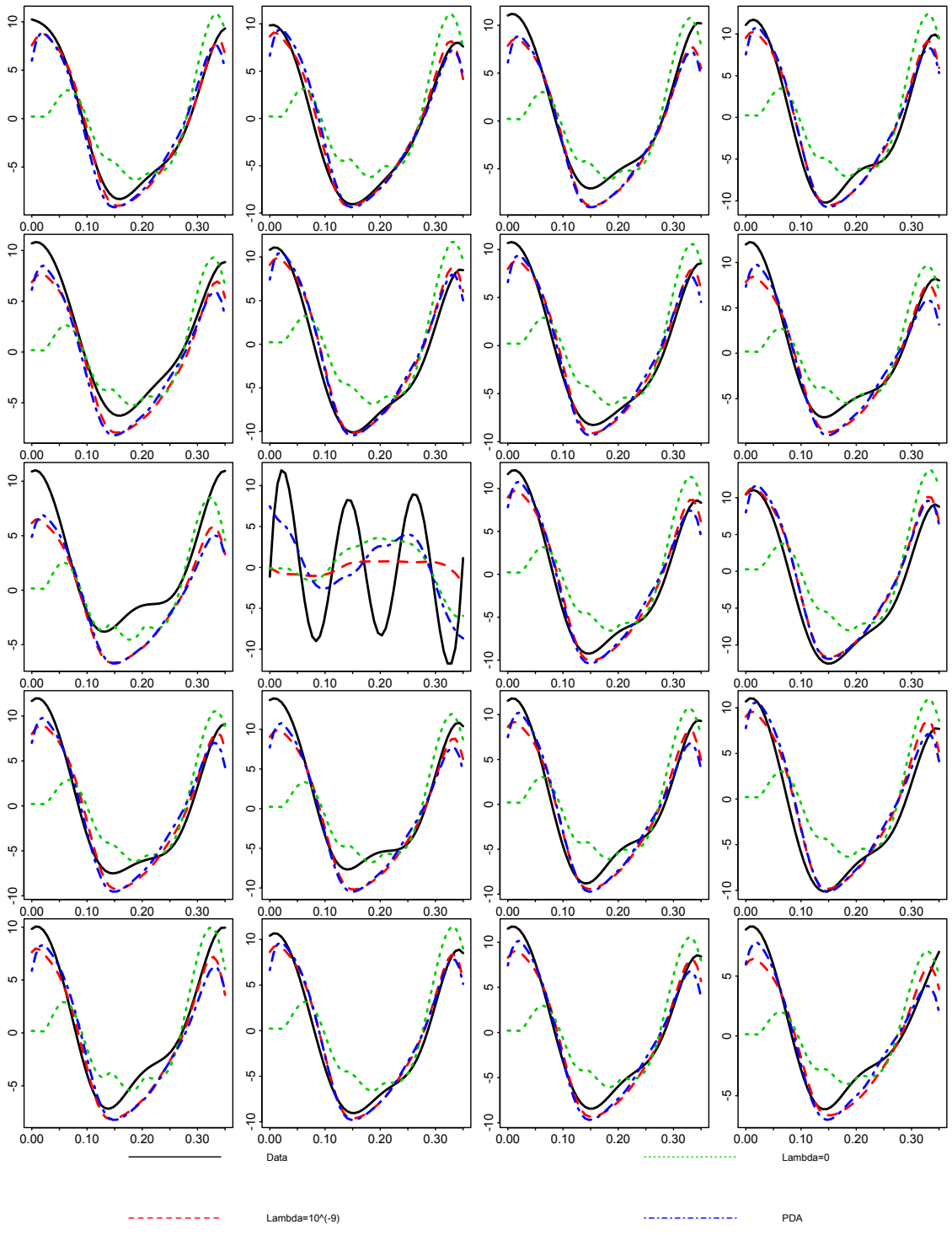


Figure 6.7: Fits of curves in lip data with outlier obtained with different levels of smoothing applied to estimated coefficients of the LDO before finding null basis functions ($\gamma = 1$).

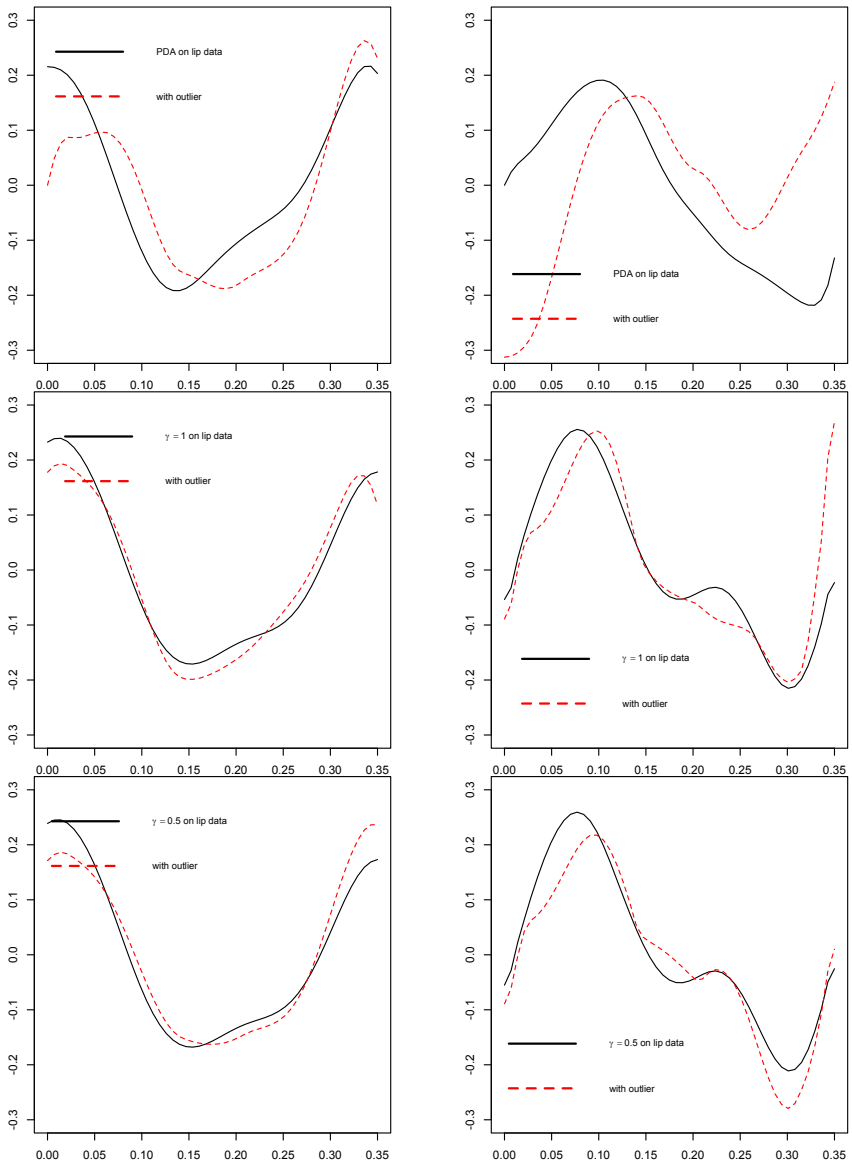


Figure 6.8: Comparison of null basis functions obtained by PDA methodology and by resampling method applied to the lip data with and without an outlier. Second row: estimated coefficients of the LDO were smoothed with $\lambda_{cv} = 10^{-9}$ for lip data with outlier; otherwise, smoothed with $\lambda_{cv} = 10^{-8}$. Third row: estimated coefficients of the LDO were smoothed with $\lambda_{cv} = 10^{-8}$. First column is $\hat{u}_1$, and second column is $\hat{u}_2$.

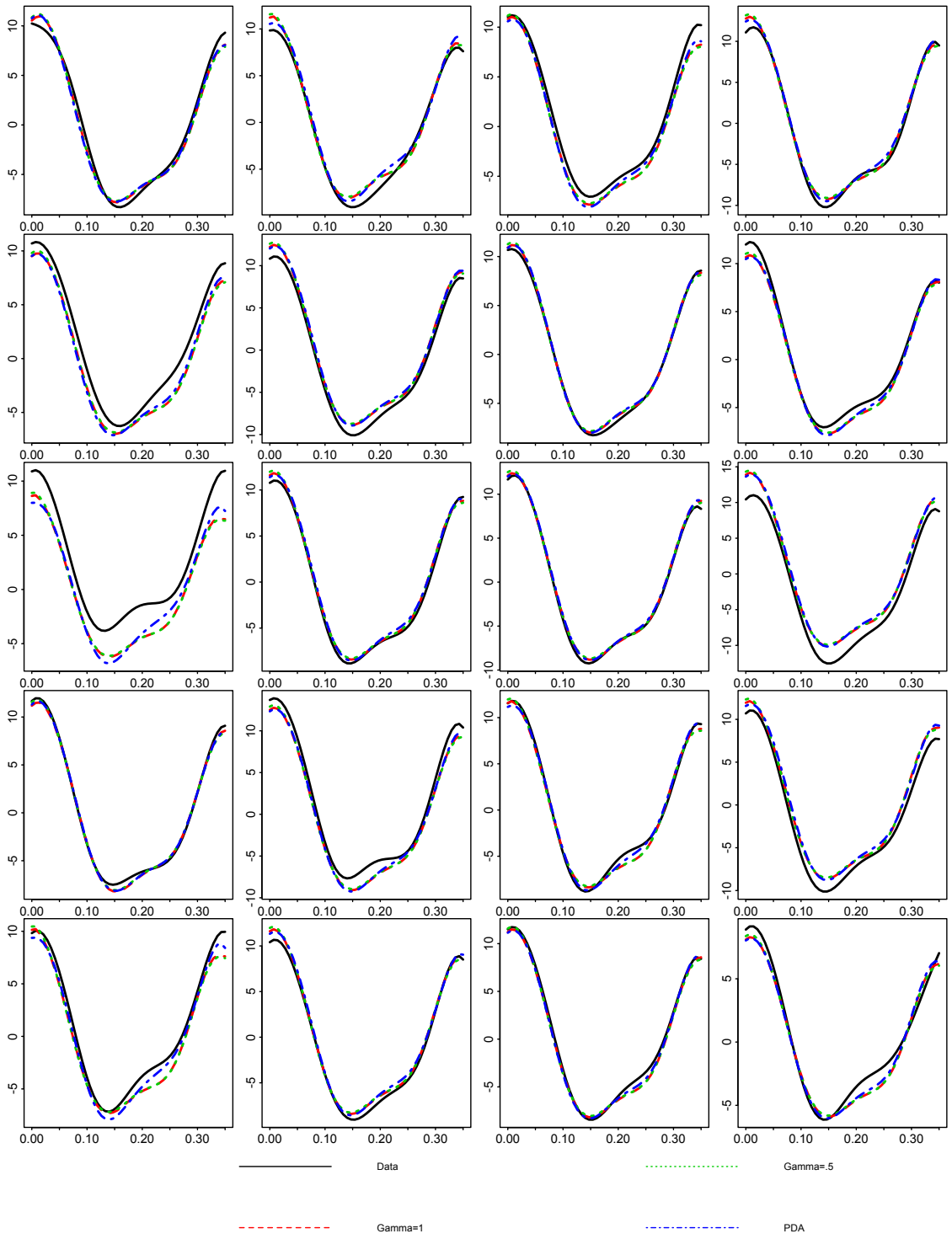


Figure 6.9a: Fits of curves in lip data obtained with PDA methodology and with resampling method ($\gamma = 1$ and $\gamma = 1/2$, estimated coefficients of the LDO smoothed with $\lambda_{cv} = 10^{-8}$).

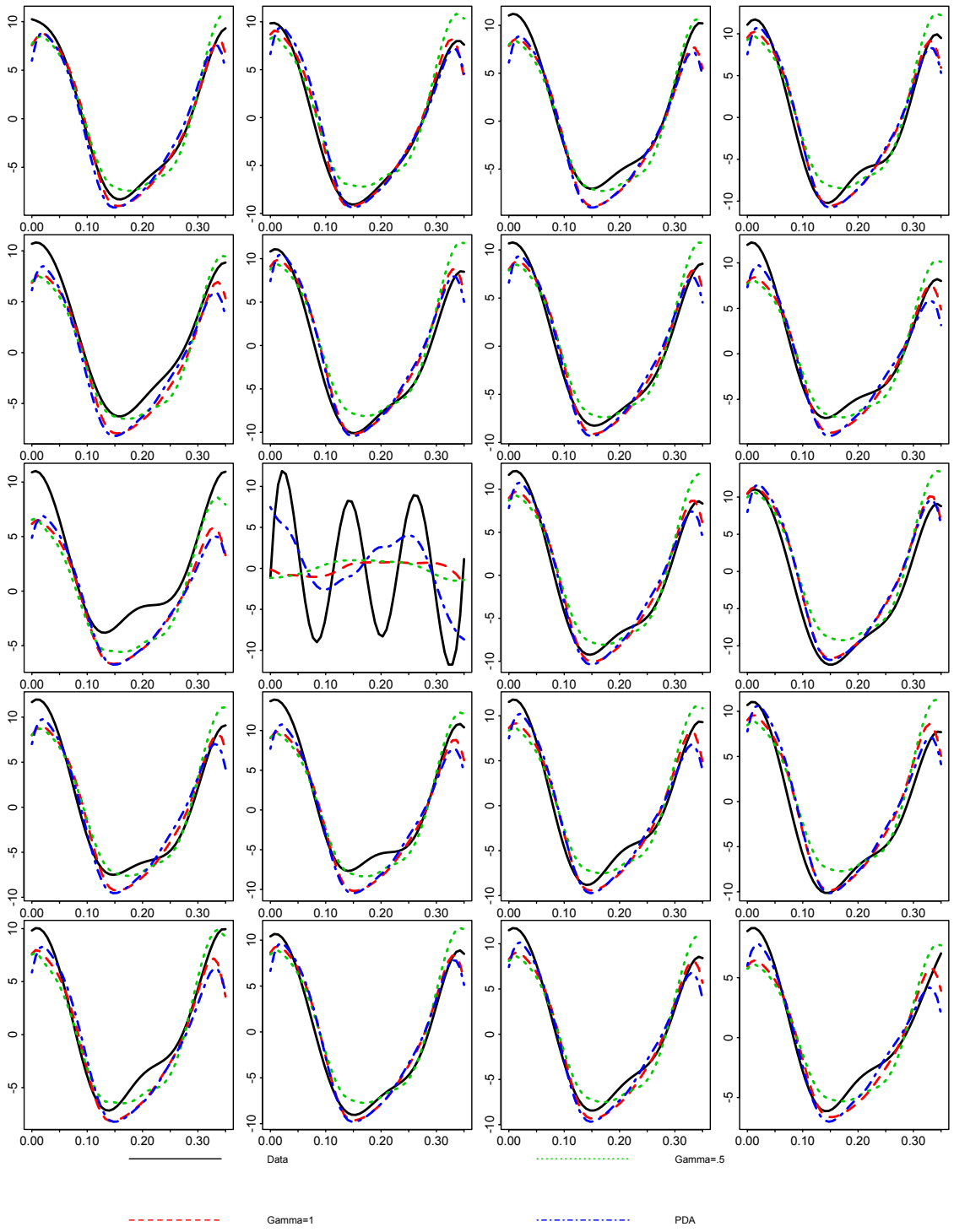


Figure 6.9b: Fits of curves in lip data with outlier obtained with PDA methodology and with resampling method (estimated coefficients of the LDO smoothed with $\lambda_{cv} = 10^{-9}$ for $\gamma = 1$, and with $\lambda_{cv} = 10^{-8}$ for $\gamma = 1/2$).

Chapter 7

Implementation of Resampling Method, m=4

7.1 Introduction

In this chapter we present results obtained by applying the resampling method to two data sets: simulated data with noise and the set of CAEP curves described in the introduction of this document. In both cases we will assume that the data is best described by a LDO of order 4.

The simulated data was obtained as follows: given the LDO

$$L = -I + D^4$$

with analytical solutions

$$u_1(t) = e^t, \quad u_2(t) = e^{-t}, \quad u_3(t) = \cos t \text{ and } u_4(t) = \sin t,$$

$u_1(t)$, $u_2(t)$, $u_3(t)$ and $u_4(t)$ are used to generate a set of 20 curves

$$\begin{aligned} x_i &= a_1 u_1(t) + a_2 u_2(t) + a_3 u_3(t) + a_4 u_4(t) \\ &= a_1 e^t + a_2 e^{-t} + a_3 \cos t + a_4 \sin t \end{aligned}$$

with randomly chosen coefficients

$$\begin{pmatrix} a_{1,i} \\ a_{2,i} \end{pmatrix} \sim N_2 \left(\begin{pmatrix} 5 \\ 5 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix} \right)$$

for $i = 1, \ldots 20$ and 63 equally spaced points $t \in [0, \pi]$. This data set was further modified by contaminating it with noise $\epsilon \sim N(0, \sigma^2)$, and $\sigma = 3$:

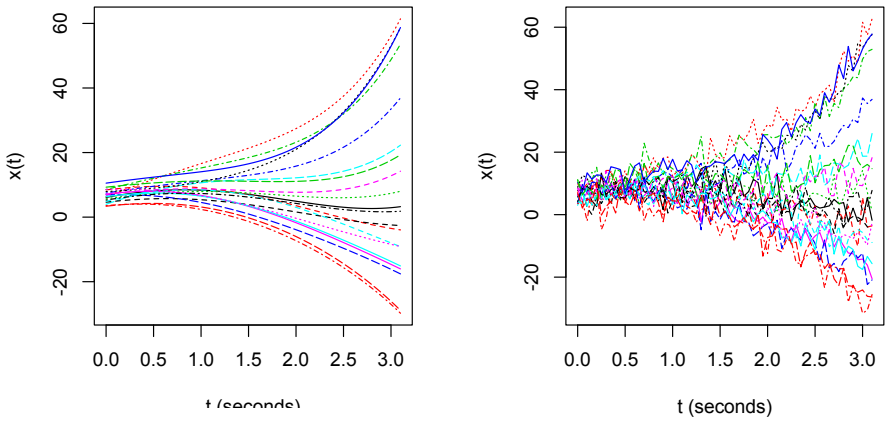
$$x_i = a_{1,i}e^t + a_{2,i}e^{-t} + a_{4,i}\cos t + a_{4,i}\sin t + \sigma\epsilon, \quad i = 1, \ldots, n$$

As described in the document introduction, the set of CAEP curves consists of $n = 74$ observations corresponding to children with normal hearing, with ages ranging from a few months to almost 7 years old. For this analysis of the CAEP curves, we will consider the interval [0,300], which delimits the P1 wave of the CAEP. The P1 wave is the first positive peak after presentation of the synthetic auditory stimulus. The location of the P1 wave in time, called the P1 latency, is used as a biomarker for maturation (plasticity) of central auditory pathways and has been reported to decrease with age (Sharma, Dorman, and Spahr 2002). Figure 7.1 displays the simulated data with and without noise, as well as the plot of CAEP curves over the intervals [0, 300].

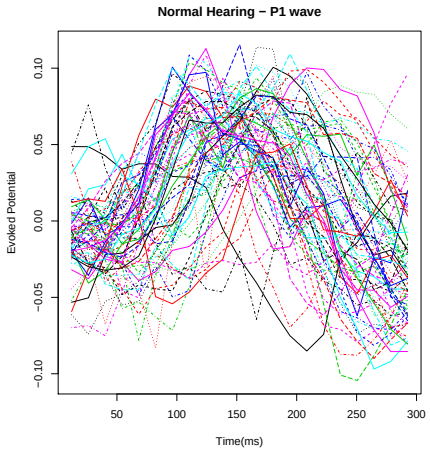
Results obtained for the simulated data without noise ($\sigma = 0$) will not be discussed here as the proposed method exactly reproduced the originating basis functions $u_1(t) = e^t$, $u_2(t) = e^{-t}$, $u_3(t) = \cos t$ and $u_4(t) = \sin t$, and yielded fits that matched exactly the set of simulated data curves. That is, applying the proposed resampling method to the simulated data with $\sigma = 0$ simply verified that the program gave correct results. The rest of this chapter illustrates the disastrous effects of noise that can be overcome by smoothing.

7.2 Simulated Data

The proposed method was applied to the simulated data with noise $\sigma = 3$. The entries on Table 7.1 show the $\sum_{i=1}^{n} \|Lx_i\|^2$ calculated after smoothing the estimates of the LDO coefficients with parameter λ_{cv}. The numbers in parenthesis are the $\sum_{i=1}^{n} \|Lx_i\|^2$ computed before smoothing the estimates of the LDO coefficients.



(a) (b)



(c)

Figure 7.1: (a) Simulated data, with noise $\sigma = 0$; (b) simulated data, with noise $\sigma = 3$; (c) P1 wave of CAEP curves of children with normal hearing, up to age 7.

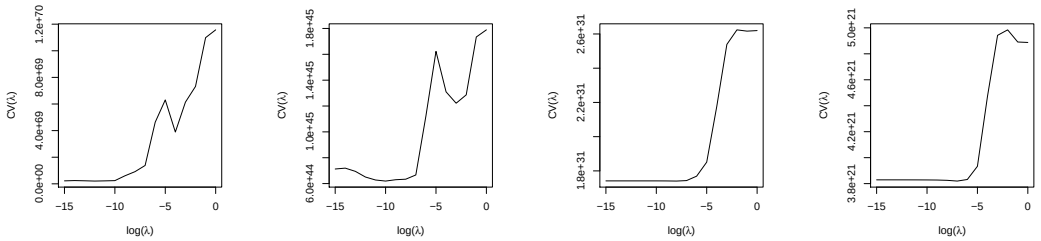
The global minima in the cross-validation curves (Figure 7.2) produced the best fits in each case, and were selected as λ_{cv}, also included in Table 7.1. For the data preprocessed with $\lambda = 10^{-12}$, $\lambda_{cv} = 10^{-12}$ when $\gamma = 1$, and $\lambda_{cv} = 10^{-10}$ when $\gamma = 1/2$. For the data preprocessed with $\lambda = 10^{-3}$, $\lambda_{cv} = 10^{-8}$ when $\gamma = 1$, and $\lambda_{cv} = 10^{-7}$ when $\gamma = 1/2$.

As stated in the previous chapter, smoothing the estimates of the LDO coefficients does not change the sum of squared norms of the forcing functions by much, which can be observed by comparing each of the entries on Table 7.1 to its corresponding entry in parenthesis. Also note that the heavier smoothing in the preprocessing step ($\lambda = 10^{-3}$) accounts for most of the sum of squared norms of the forcing functions, since the sums before and after smoothing the coefficients estimates are virtually the same.

Table 7.1: Simulated data: sum of squared norms $\sum_{i=1}^{n} \|Lx_i\|^2$ for the proposed resampling method before (in parenthesis) and after smoothing estimates of coefficients with λ_{cv}.

Sum of squared norms $\sum_{i=1}^{n} \|Lx_i\|^2$, resampling method		
	Simulated data, $\sigma = 3$	
	preprocess $\lambda = 10^{-12}$	preprocess $\lambda = 10^{-3}$
$\gamma = 1$	2.03×10^{14} (1.8×10^{14}) $\lambda_{cv} = 10^{-12}$	1.0×10^{5} (1.0×10^{5}) $\lambda_{cv} = 10^{-8}$
$\gamma = 1/2$	1.17×10^{15} (5.7×10^{14}) $\lambda_{cv} = 10^{-10}$	2.11×10^{5} (2.1×10^{5}) $\lambda_{cv} = 10^{-7}$

The estimated basis functions for the null space of the LDO for $\gamma = 1$ and $\gamma = 1/2$ are shown on Figures 7.3a and 7.3b, respectively. In both figures we can observe how much smoother were the estimated basis functions for the data preprocessed with $\lambda = 10^{-3}$ than those for the data preprocessed with $\lambda = 10^{-12}$.



(a) $\gamma = 1$ (b) $\gamma = 1/2$ (c) $\gamma = 1$ (d) $\gamma = 1/2$

Figure 7.2: Cross-validation curves for simulated data, $\sigma = 3$; (a) and (b) preprocessed with $\lambda = 10^{-12}$; (c) and (d) preprocessed with $\lambda = 10^{-3}$

Given the effect of data preprocessing on the estimated basis functions, it is not surprising to see the dramatic difference on the fits when comparing Figure 7.4a to Figure 7.5a, and Figure 7.4b to Figure 7.5b.

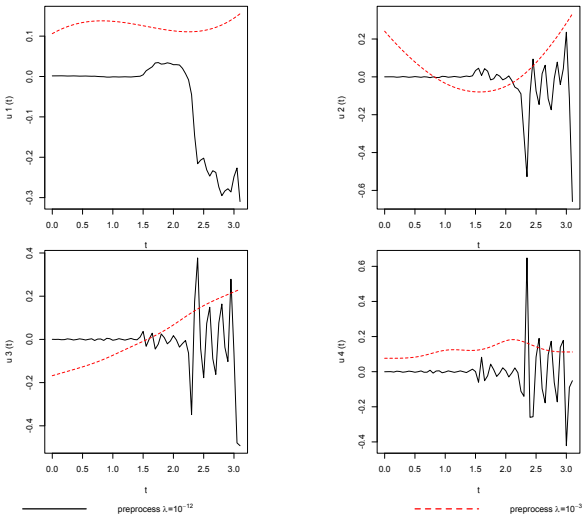


Figure 7.3a: Estimated basis functions of simulated data, $\sigma = 3$, $\gamma = 1$: $\hat{u}_1$ on first row, $\hat{u}_2$ on second row. Solid line corresponds to preproccessing with 10^{-12}; dashed line to preproccessing with 10^{-3}.

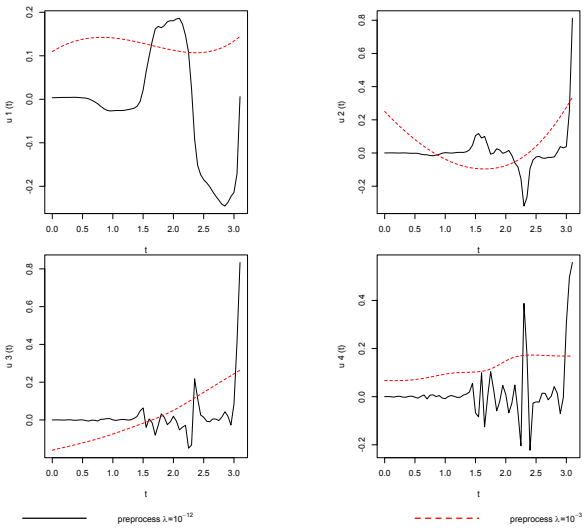


Figure 7.3b: Estimated basis functions of simulated data, $\sigma = 3$, $\gamma = 1/2$: $\hat{u}_1$ on first row, $\hat{u}_2$ on second row. Solid line corresponds to preproccessing with 10^{-12}; dashed line to preproccessing with 10^{-3}.

7.3 Normal Hearing CAEP Curves

The set of CAEP curves was preprocessed at different smoothing levels to explore the effect of preprocessing on this data, which is very noisy. The four preprocessing smoothing levels chosen were $\lambda = 10^{-3}$, $\lambda = 10^{3}$, $\lambda = 10^{7}$ and $\lambda = 10^{12}$. The entries on Table 7.2 show the $\sum_{i=1}^{n} \|Lx_i\|^2$ calculated after smoothing the estimates of the LDO coefficients with parameter λ_{cv}. As before, the local and global maxima, and the regions of plateau suggested by the cross-validation curves in Figure 7.6 were used to find the best fits, as determined by visual inspection. The values of λ_{cv} selected are listed in Table 7.3. Note that when the data was processed with $\lambda = 10^{-3}$ and $\gamma = 1/2$, the differential equation solver did not converge for any value of λ, up to the maximum that could be used ($\lambda = 10^{16}$) before producing an overflow error message. It was not possible to obtain fits to the CAEP curves in this case. Hence for Figure 7.7 ($\lambda = 10^{-3}$), there is no (a) and (b) since there are only results for $\gamma = 1$. For all other cases, the fits of selected CAEP curves are

shown in Figure 7.8 ($\lambda = 10^3$), Figure 7.9 ($\lambda = 10^7$), and Figure 7.10 ($\lambda = 10^{12}$). In each case, Figure (a) is for $\gamma = 1$, and (b) is for $\gamma = 1/2$.

Table 7.2: CAEP curves: sum of squared norms $\sum_{i=1}^{n} \|Lx_i\|^2$ for the proposed resampling method after smoothing estimates of coefficients with λ_{cv}.

Sum of squared norms $\sum_{i=1}^{n} \|Lx_i\|^2$, resampling method				
CAEP curves, normal hearing				
	preprocess $\lambda = 10^{-3}$	preprocess $\lambda = 10^{3}$	preprocess $\lambda = 10^{7}$	preprocess $\lambda = 10^{12}$
$\gamma = 1$	5.06×10^{-9}	4.58×10^{-10}	2.33×10^{-11}	4.37×10^{-17}
$\gamma = 1/2$	5.14×10^{-9}	6.31×10^{-10}	3.6×10^{-11}	1.03×10^{-16}

Table 7.3: Values of λ_{cv} selected for each preprocessing smoothing level of CAEP curves. $*$ Differential equation solver did not converge.

	Preprocessing smoothing parameter			
	10^{-3}	10^{3}	10^{7}	10^{12}
$\gamma = 1$	$\lambda_{cv} = 10^{-10}$	$\lambda_{cv} = 10^{-1}$	$\lambda_{cv} = 1$	$\lambda_{cv} = 10^{-3}$
$\gamma = 1/2$	$*$	$\lambda_{cv} = 10^{-12}$	$\lambda_{cv} = 1$	$\lambda_{cv} = 10^{-10}$

We can observe that as the curves are preprocessed with higher levels of smoothing, and the noise is reduced, the fits track the curves closer. At levels $\lambda = 10^{-3}$ and $\lambda = 10^3$ (Figures 7.7, 7.8a, and 7.8b), the curves look "noisy" and the estimated curves, although not as noisy, fail to capture some of the features of the data. When the data is preprocessed with $\lambda = 10^7$, the resulting curves are smoother, while retaining the features of the original curves. In this case, the fits follow the smoothed curves closer, specially when $\gamma = 1$ (Figure 7.9a). Finally, with a heavier preprocessing of $\lambda = 10^{12}$, the noise on the curves has been much reduced, but

many of the features have also been lost. We obtain very close fits, but the curves are oversmoothed.

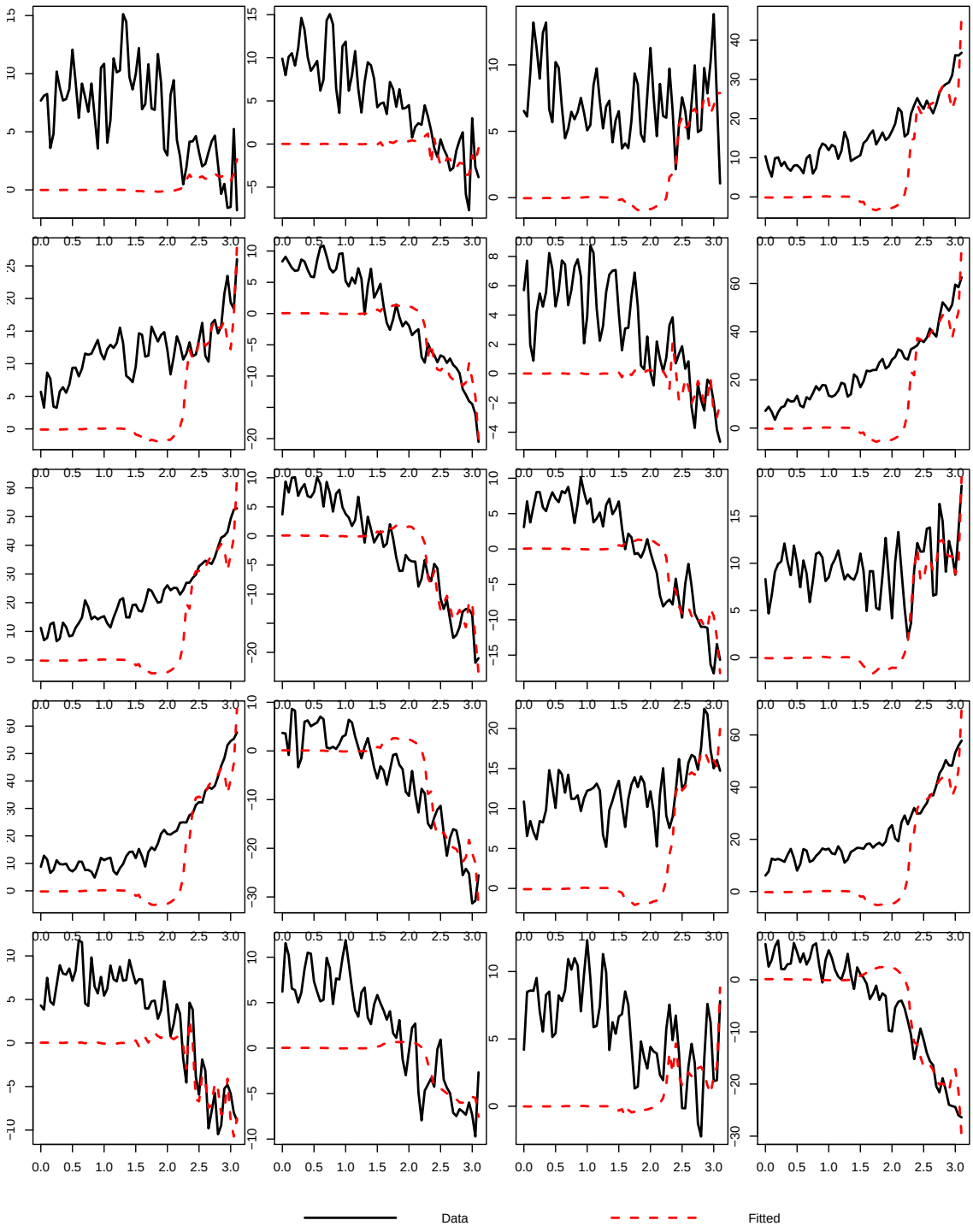


Figure 7.4a: Fits to simulated data, $\sigma = 3$, preprocessed with $\lambda = 10^{-12}$, $\gamma = 1$; estimated coefficients of the LDO smoothed with $\lambda_{cv} = 10^{-12}$. Example of poor fits due to undersmoothed data.

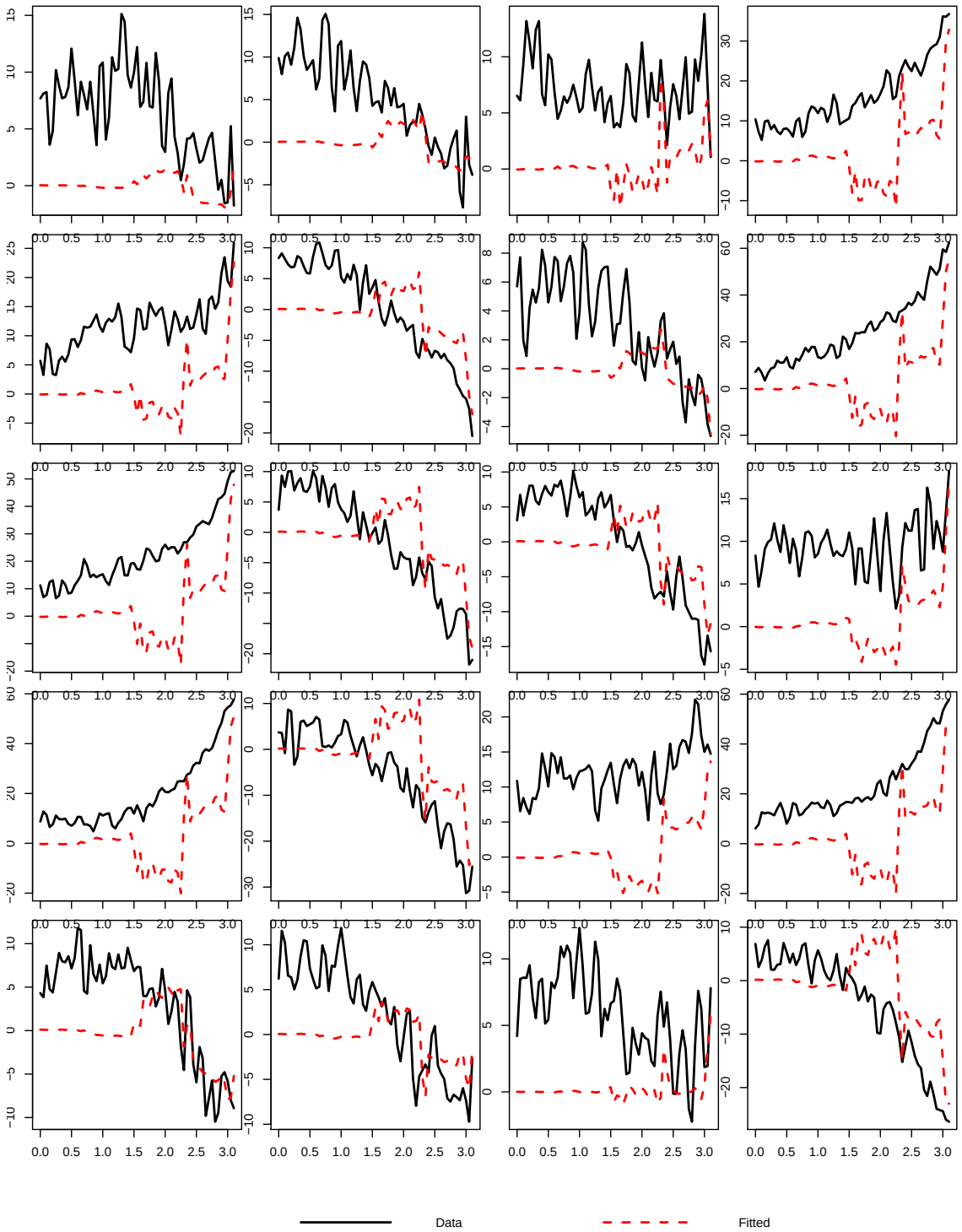


Figure 7.4b: Fits to simulated data, $\sigma = 3$, preprocessed with $\lambda = 10^{-12}$, $\gamma = 1/2$; estimated coefficients of the LDO smoothed with $\lambda_{cv} = 10^{-10}$. Example of poor fits due to undersmoothed data.

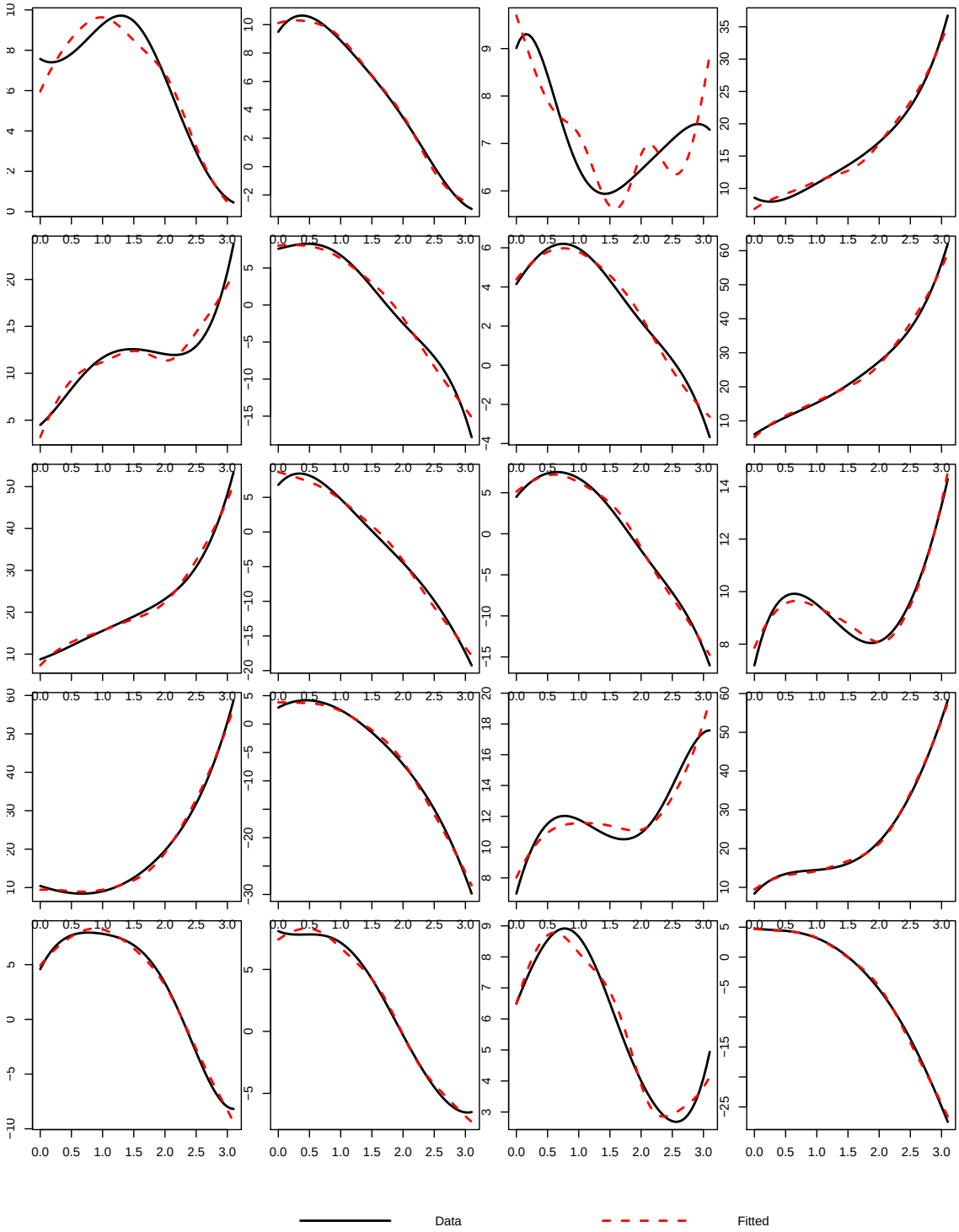


Figure 7.5a: Fits to simulated data, $\sigma = 3$, preprocessed with $\lambda = 10^{-3}$, $\gamma = 1$; estimated coefficients of the LDO smoothed with $\lambda_{cv} = 10^{-8}$. Improved fits after smoothing data.

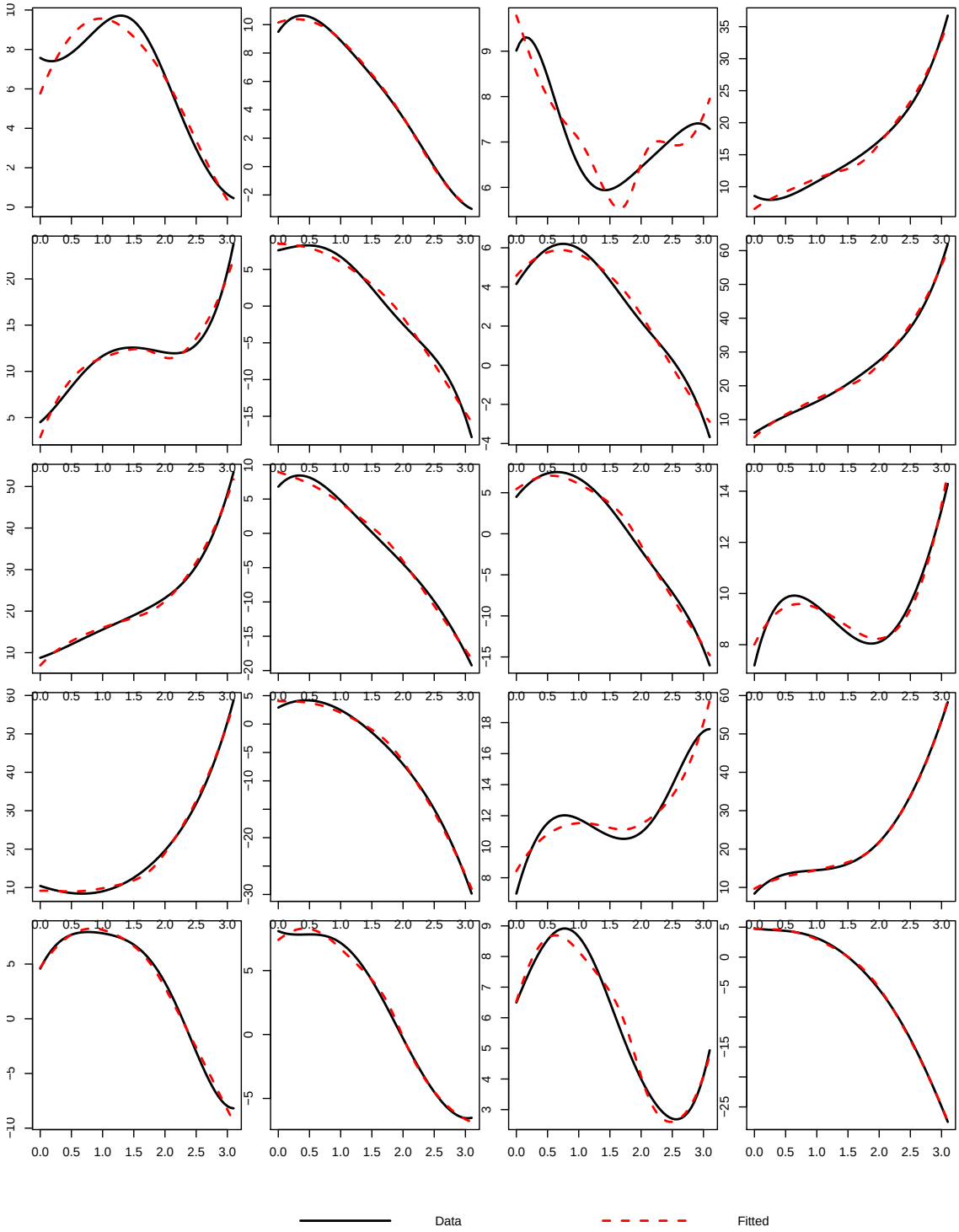
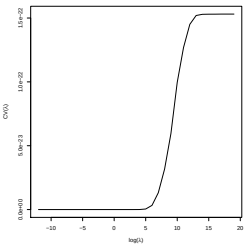
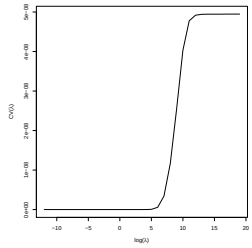


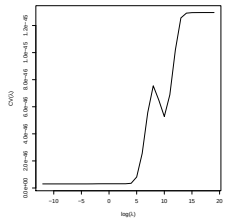
Figure 7.5b: Fits to simulated data, $\sigma = 3$, preprocessed with $\lambda = 10^{-3}$, $\gamma = 1/2$; estimated coefficients of the LDO smoothed with $\lambda_{cv} = 10^{-7}$. Improved fits after smoothing data.



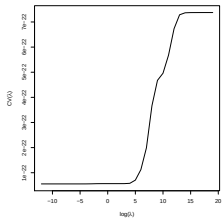
(a) $\lambda = 10^{-3}$, $\gamma = 1$



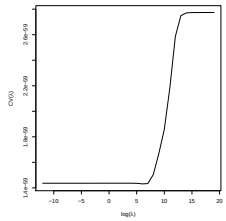
(b) $\lambda = 10^{-3}$, $\gamma = 1/2$



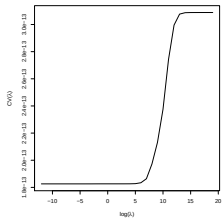
(c) $\lambda = 10^{3}$, $\gamma = 1$



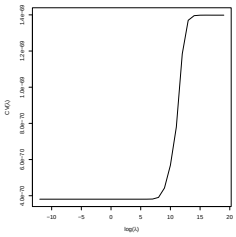
(d) $\lambda = 10^{3}$, $\gamma = 1/2$



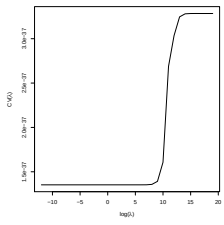
(e) $\lambda = 10^{7}$, $\gamma = 1$



(f) $\lambda = 10^{7}$, $\gamma = 1/2$



(g) $\lambda = 10^{12}$, $\gamma = 1$



(h) $\lambda = 10^{12}$, $\gamma = 1/2$

Figure 7.6: Cross-validation curves for CAEP curves; data preprocessed with smoothing parameter λ in caption.

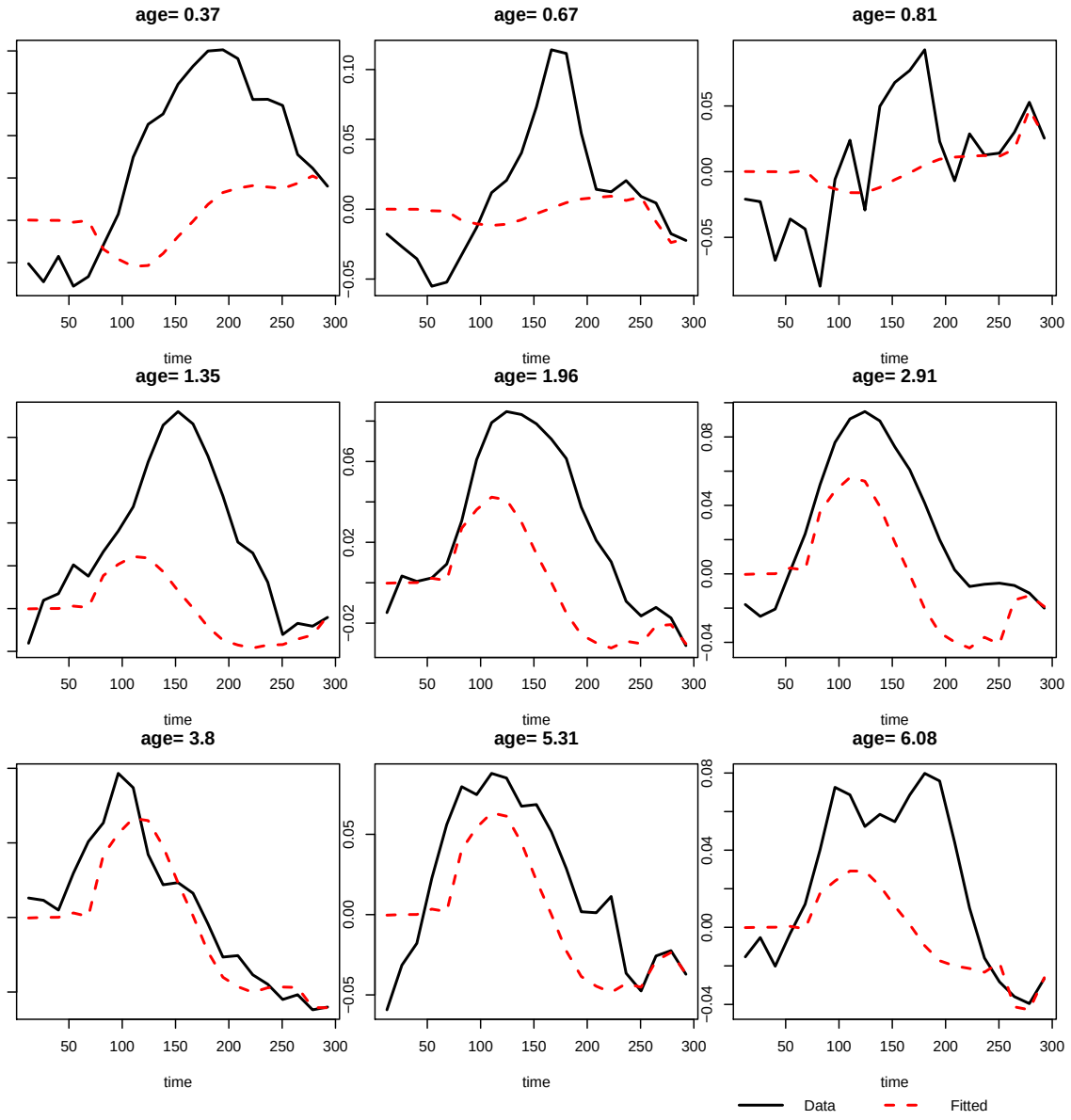


Figure 7.7: Fits to selected CAEP curves, preprocessed with $\lambda = 10^{-3}$, $\gamma = 1$; estimated coefficients of the LDO smoothed with $\lambda_{cv} = 10^{-10}$. Example of poor fits due to undersmoothed data.

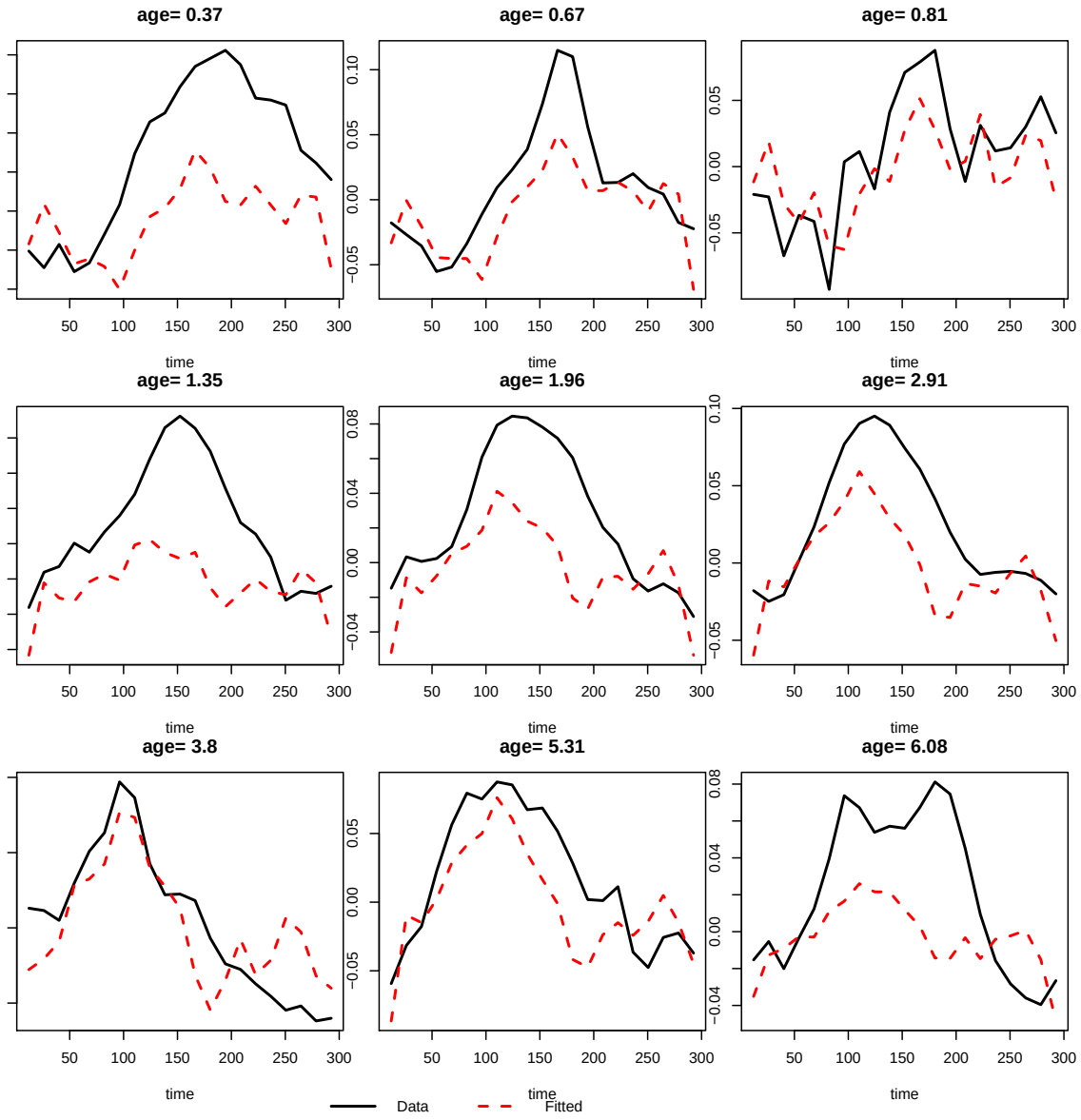


Figure 7.8a: Fits to selected CAEP curves, preprocessed with $\lambda = 10^3$, $\gamma = 1$; estimated coefficients of the LDO smoothed with $\lambda = 10^{-1}$. Example of poor fits due to undersmoothed data.

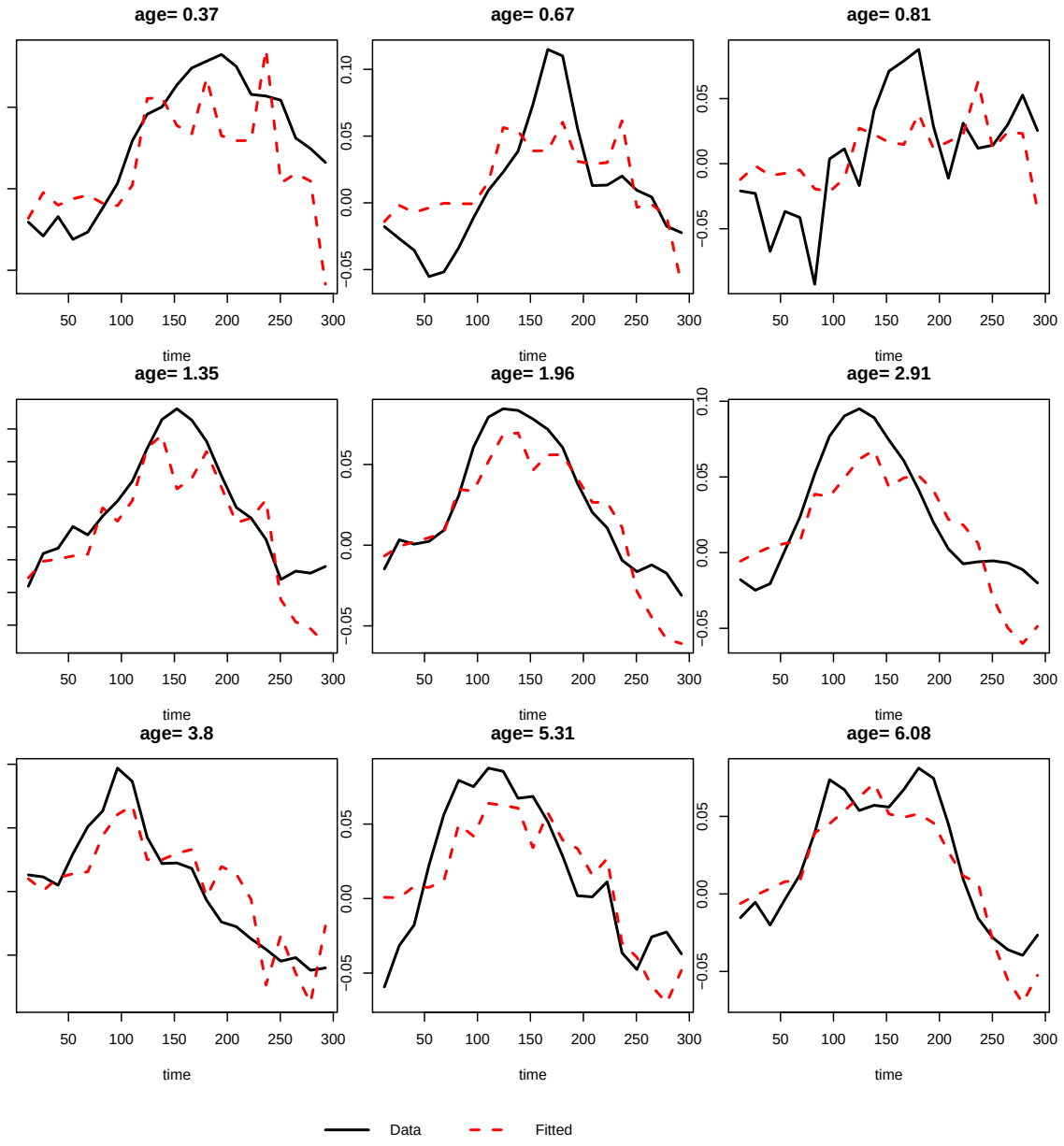


Figure 7.8b: Fits to selected CAEP curves, preprocessed with $\lambda = 10^3$, $\gamma = 1/2$; estimated coefficients of the LDO smoothed with $\lambda_{cv} = 10^{-12}$. Example of poor fits due to undersmoothed data.

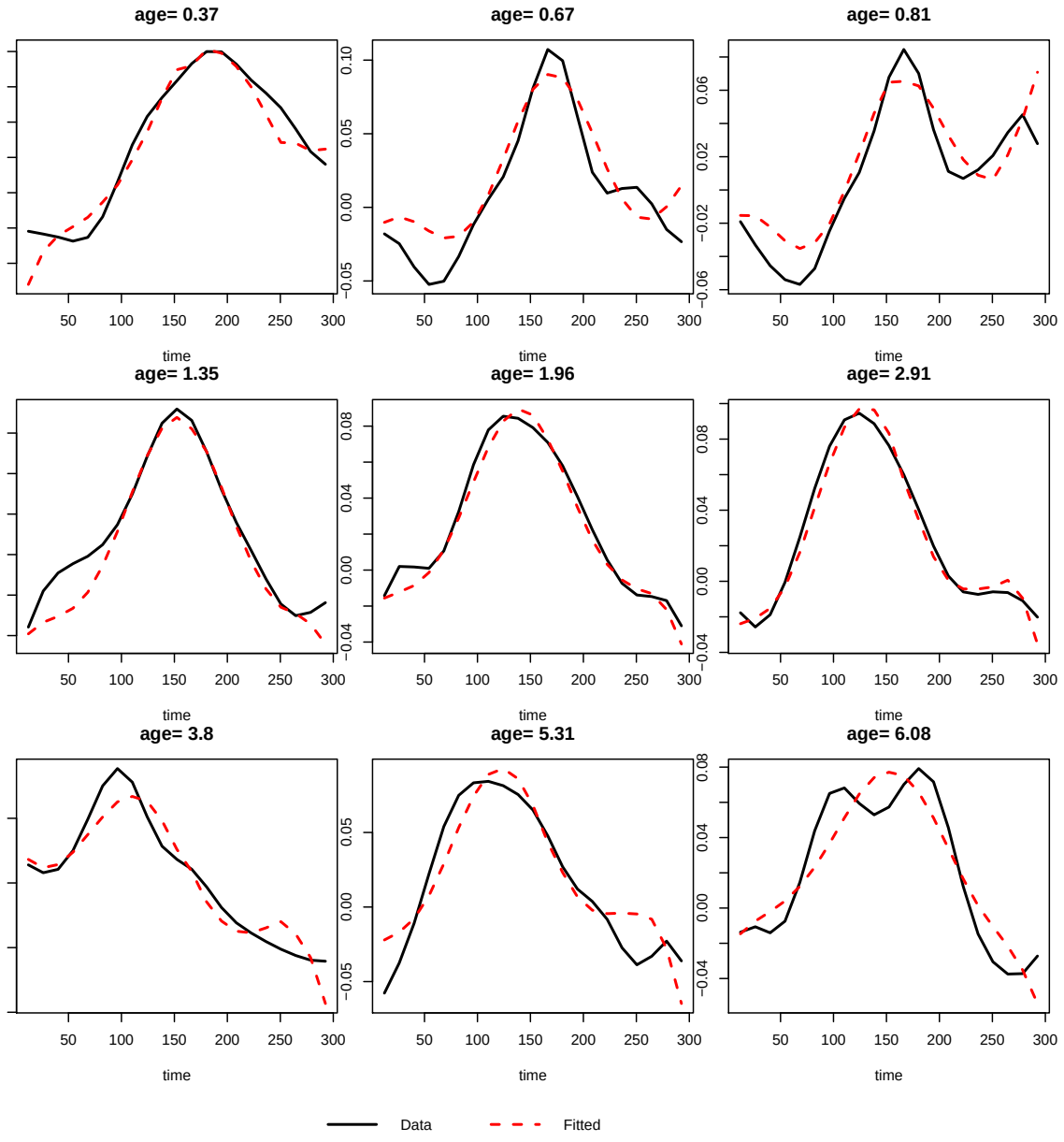


Figure 7.9a: Fits to selected CAEP curves, preprocessed with $\lambda = 10^7$, $\gamma = 1$; estimated coefficients of the LDO smoothed with $\lambda = 1$. Improved fits after smoothing curves without losing main features.

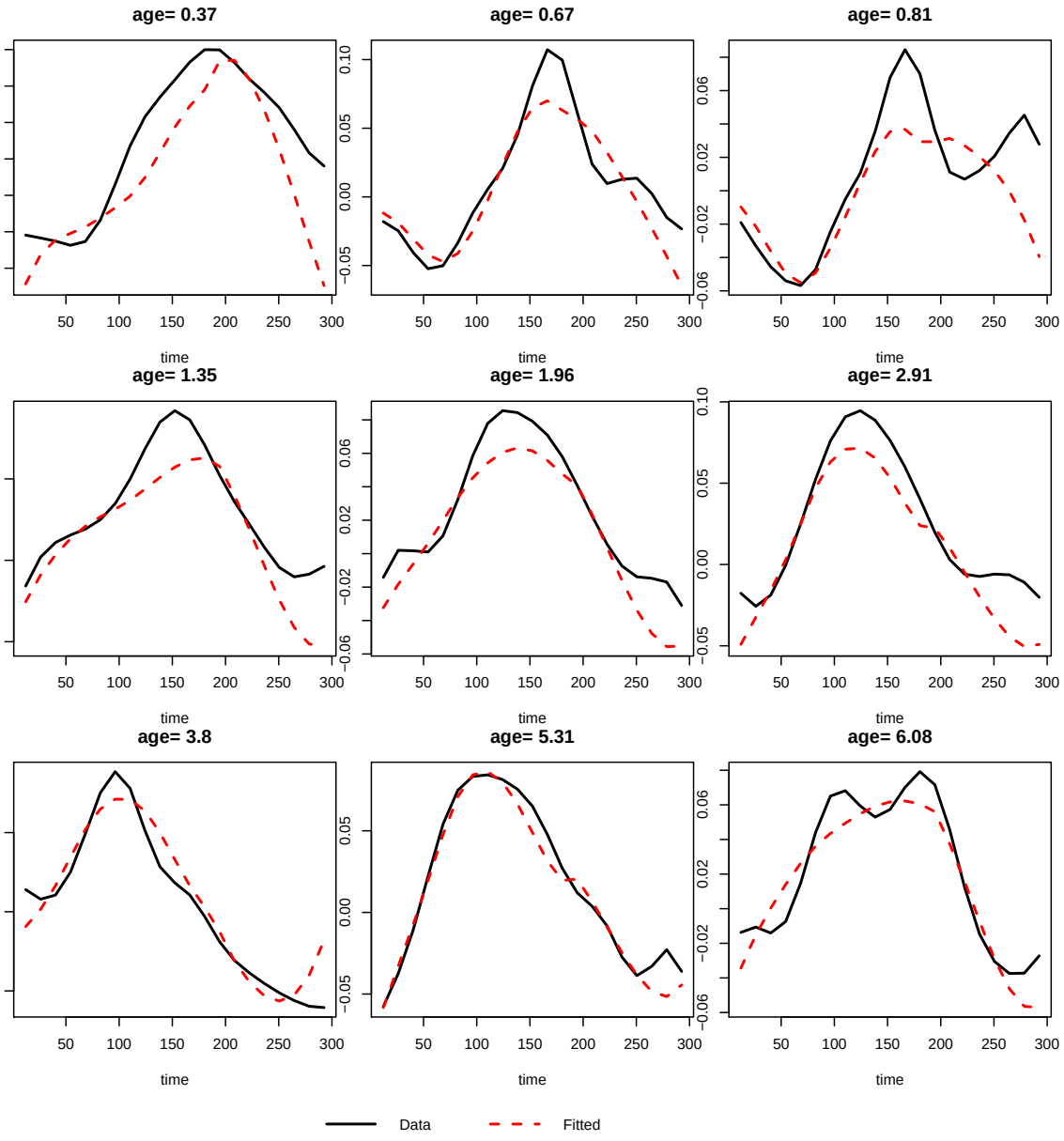


Figure 7.9b: Fits to selected CAEP curves, preprocessed with $\lambda = 10^7$, $\gamma = 1/2$; estimated coefficients of the LDO smoothed with $\lambda = 1$. Improved fits after smoothing curves without losing main features.

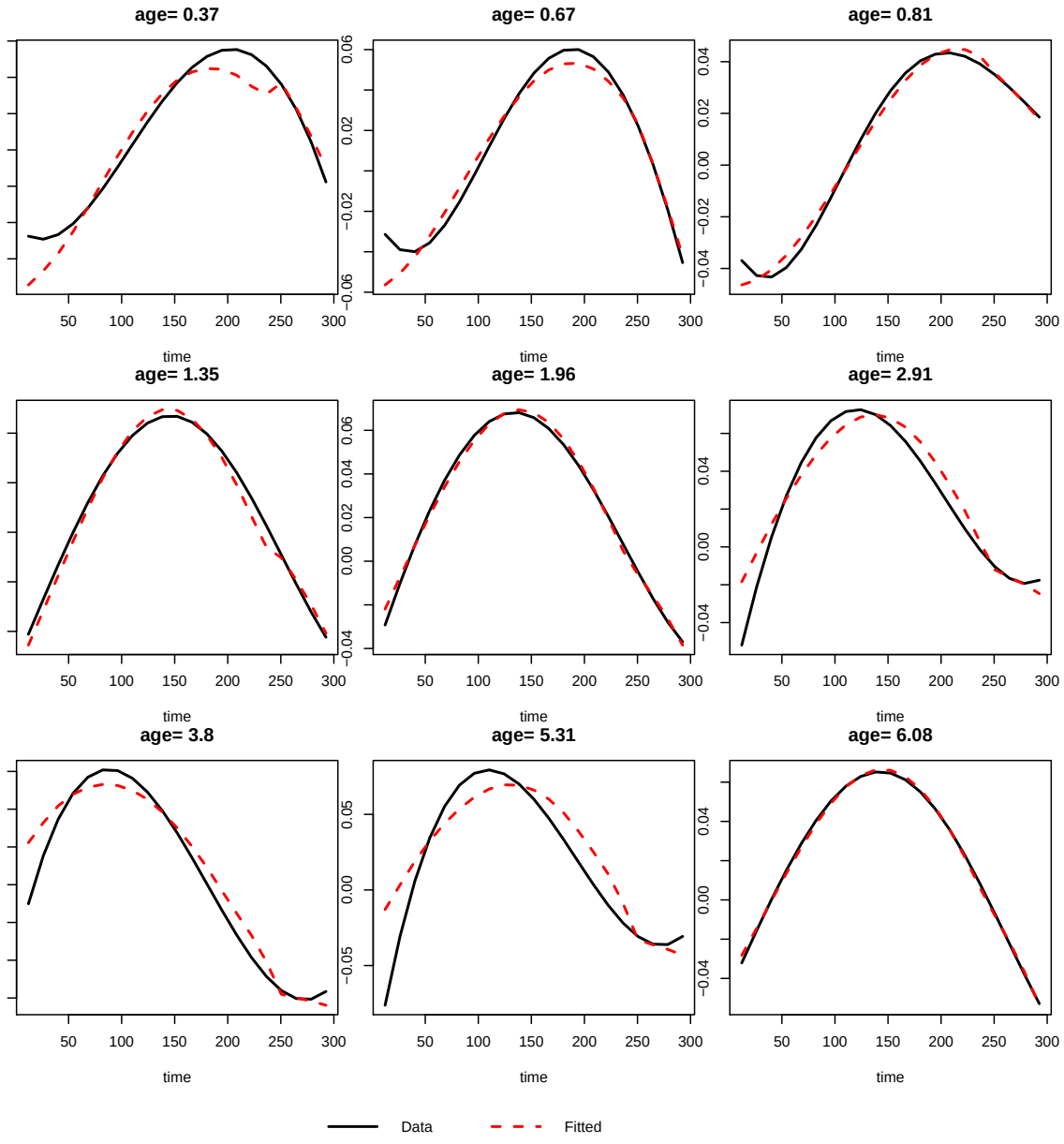


Figure 7.10a: Fits to selected CAEP curves, preprocessed with $\lambda = 10^{12}$, $\gamma = 1$; estimated coefficients of the LDO smoothed with $\lambda_{cv} = 10^{-3}$. Very close fits with oversmoothed curves.

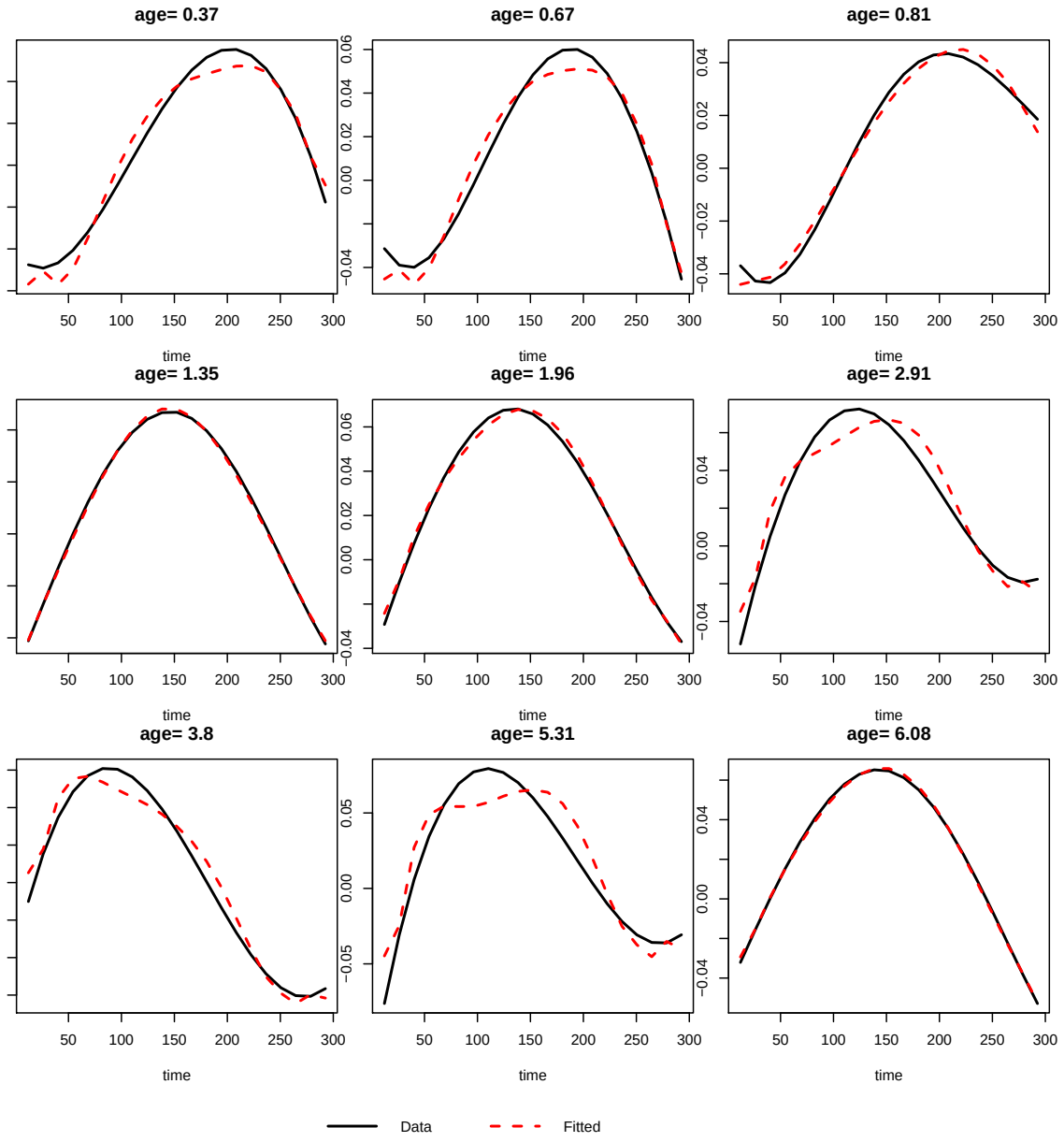


Figure 7.10b: Fits to selected CAEP curves, preprocessed with $\lambda = 10^{12}$, $\gamma = 1/2$; estimated coefficients of the LDO smoothed with $\lambda_{cv} = 10^{-10}$. Very close fits with oversmoothed curves.

Chapter 8

Analysis of CAEP curves by age subgroups

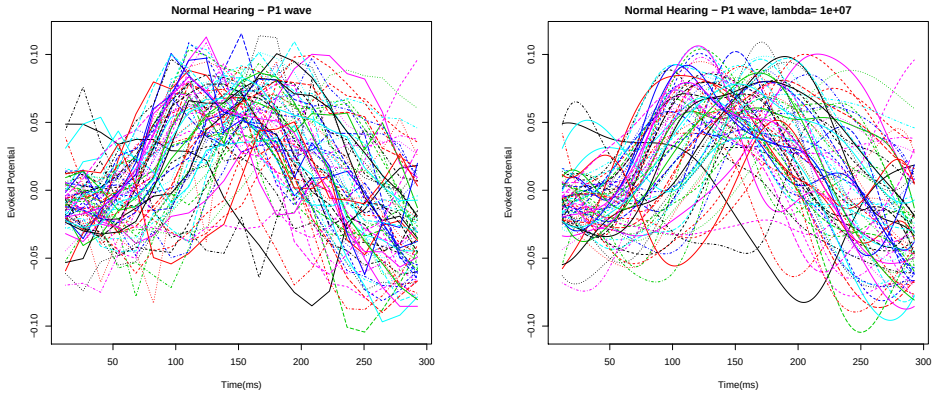
In this chapter we continue considering the P1 waves of the CAEP curves, pre-processed with $\lambda = 10^7$, assuming the data is best described by an LDO of order $m = 4$. Figure 8.1 displays the pre-smoothed and smoothed waves.

Recall that each of the curves x_i may be estimated as the linear combination of the estimated basis functions $\hat{u}_1, \ldots, \hat{u}_m$ of the null space of the LDO:

$$\hat{x}_i(t) = \sum_{j=1}^{m} a_{ij}\hat{u}_j(t), \quad i = 1, \ldots, n.$$

We examine the scatterplots of the coefficients a_{ij} for the 74 CAEP curves, when $m = 4$ in Figure 8.2: the coefficients follow clearer patterns for $\hat{u}_1$ and $\hat{u}_2$ than for $\hat{u}_3$ and $\hat{u}_4$. For $\hat{u}_1$, the data fluctuates over a wider range for the younger children, and over a narrower range around a nonzero constant as age increases. For $\hat{u}_2$, most of the coefficients are negative for children younger than 2 or 3 years old, and positive for the older children. As in the case of $\hat{u}_1$, the coefficients of $\hat{u}_2$, $\hat{u}_3$ and $\hat{u}_4$ also fluctuate less as age increases, but they seem to vary around zero. After looking for general patterns, and relying on general knowledge of child development, we classify the responses into four subgroups, also taking care that we have an adequate number of curves to conduct additional analysis. The age subgroups chosen are: ages 0 to 1 (25 curves), 1 to 2 (15 curves), 2 to 4 (21 curves) and 4 to 7 (13 curves).

As previously defined, λ_{cv} denotes the value of the smoothing parameter cho-



(a) (b)

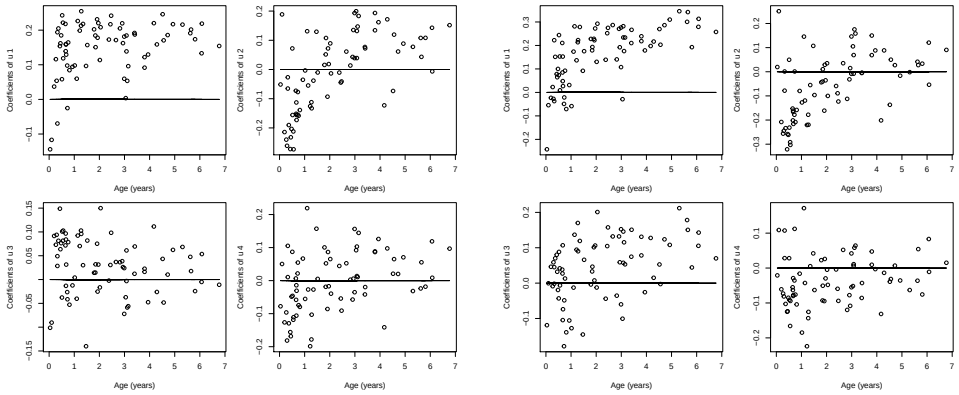
Figure 8.1: P1 waves in CAEP curves of children with normal hearing, ages 0 to 7 years old. (a) 74 original curves; (b) smoothed with smoothing parameter $\lambda = 10^7$.

sen after inspecting local and global minima of the cross-validation curve, as well as regions of plateaus. Table 8.1 presents the values tested after examining the cross-validation curves displayed in Figure 8.3, and the value of λ_{cv} selected for each age group. Results of a PDA analysis for each of the subgroups are described in the next subsections.

8.0.1 Ages 0 to 1

This subgroup consists of 25 curves, shown in Figure 8.4; several outliers can be observed. As recorded on Table 8.1, the estimates of the LDO coefficients were smoothed with $\lambda_{cv} = 10^{-3}$ when $\gamma = 1$, and with $\lambda_{cv} = 10^5$ when $\gamma = 1/2$.

The first plot of estimated basis functions in Figure 8.5 shows that $\hat{u}_1$ for both values of γ capture the P1 wave of the CAEP curves. The other estimated basis functions for this age subgroup look smoother for $\gamma = 1/2$ than for $\gamma = 1$. The fits for this age group (see Figure 8.6) are somewhat closer to the curves for $\gamma = 1/2$ than for $\gamma = 1$. This is consistent with previous results suggesting that the

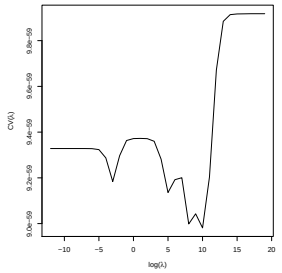


(a) $\gamma = 1$ (b) $\gamma = 1/2$

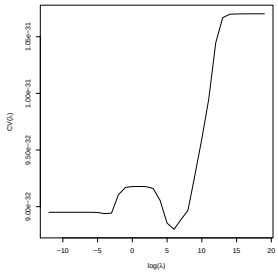
Figure 8.2: Scatterplots of coefficients of estimated basis functions: first row, for $\hat{u}_1$ and $\hat{u}_2$; second row, for $\hat{u}_3$ and $\hat{u}_4$. Data preprocessed with $\lambda = 10^7$; estimated coefficients of the LDO smoothed with $\lambda = 1$ for $\gamma = 1$ and $\gamma = 1/2$.

Table 8.1: Values of smoothing parameter tested and selected to smooth estimates of the LDO coefficients in each age group.

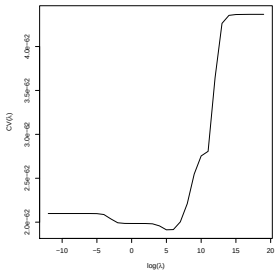
Age Subgroup	γ	λ_{cv} selected	Values tested
0 to 1	$\gamma = 1$	10^{-3}	$10^{-3}, 10^{5}, 10^{10}$
	$\gamma = 1/2$	10^{5}	$10^{-3}, 10^{5}, 10^{6}$
1 to 2	$\gamma = 1$	10^{-2}	$10^{-2}, 10^{-3}, 10^{5}$
	$\gamma = 1/2$	10^{6}	$10^{-2}, 10^{-3}, 10^{6}$
2 to 4	$\gamma = 1$	10^{5}	$10^{-2}, 10^{5}, 10^{13}$
	$\gamma = 1/2$	10^{5}	$10^{-2}, 10^{5}, 10^{11}$
4 to 7	$\gamma = 1$	10^{5}	$10^{-2}, 10^{5}, 10^{13}$
	$\gamma = 1/2$	10^{5}	$10^{-2}, 10^{4}, 10^{5}$



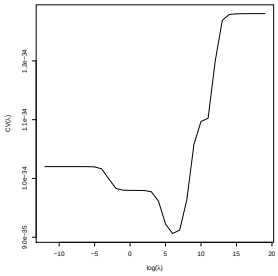
(a) Ages 0 to 1, $\gamma = 1$



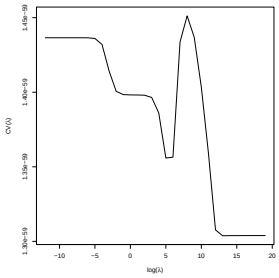
(b) Ages 0 to 1, $\gamma = 1/2$



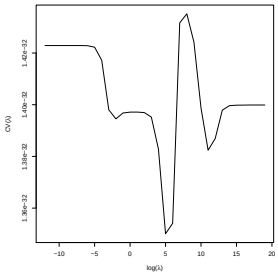
(c) Ages 1 to 2, $\gamma = 1$



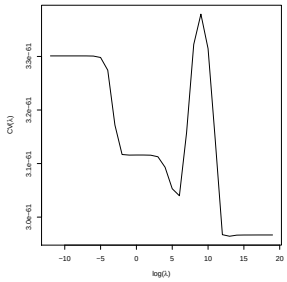
(d) Ages 1 to 2, $\gamma = 1/2$



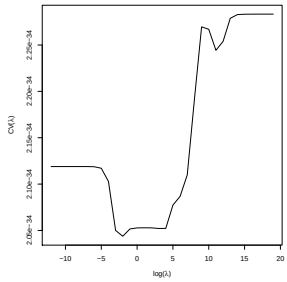
(e) Ages 2 to 4, $\gamma = 1$



(f) Ages 2 to 4, $\gamma = 1/2$

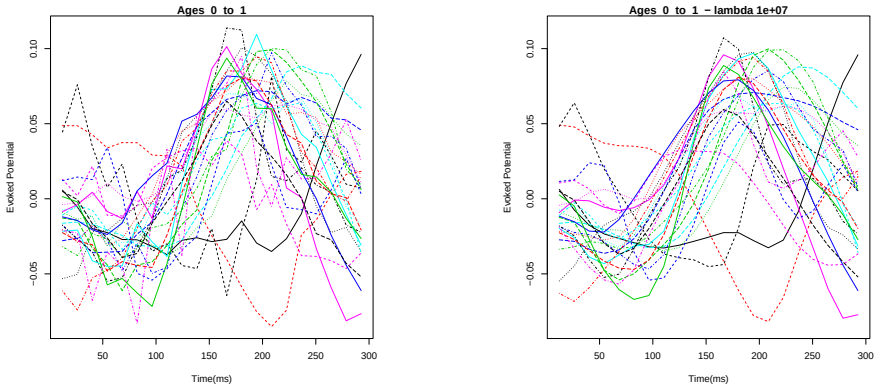


(g) Ages 4 to 7, $\gamma = 1$



(h) Ages 4 to 7, $\gamma = 1/2$

Figure 8.3: Cross-validation curves for P1 waves in CAEP curves for each age group. Each set of curves preprocessed with smoothing parameter $\lambda = 10^7$.



(a) (b)

Figure 8.4: P1 waves in CAEP curves of children with normal hearing, ages 0 to 1 years old. (a) 25 original curves; (b) smoothed with parameter $\lambda = 10^7$.

estimates under $\gamma = 1/2$ are more robust: several curves (corresponding to ages 0.04, 0.1, 0.33) could be considered outliers in this group. Finally, the coefficients of the estimated basis functions in Figure 8.7 show no obvious patterns for $\gamma = 1/2$, except that the coefficients of $\hat{u}_1$ vary around a nonzero constant, while the others seem to vary around zero. This suggests that $\hat{u}_2$, $\hat{u}_3$ and $\hat{u}_4$ are capturing variability in the shape of the P1 wave across children within this age group.

8.0.2 Ages 1 to 2

Figure 8.8 shows the 15 curves in this subgroup; note that there are several curves that could be considered outliers. The estimates of the LDO coefficients were smoothed with $\lambda_{cv} = 10^{-2}$ when $\gamma = 1$, and with $\lambda_{cv} = 10^6$ when $\gamma = 1/2$.

As in the previous subgroup, the first plot in Figure 8.9 shows that the estimated basis function $\hat{u}_1$ captures the P1 wave of the CAEP curves. The other estimated basis functions have varying shapes. In examining Figure 8.10, we observe that only a few of the fits are close to the curves, and it is not clear that either $\gamma = 1$

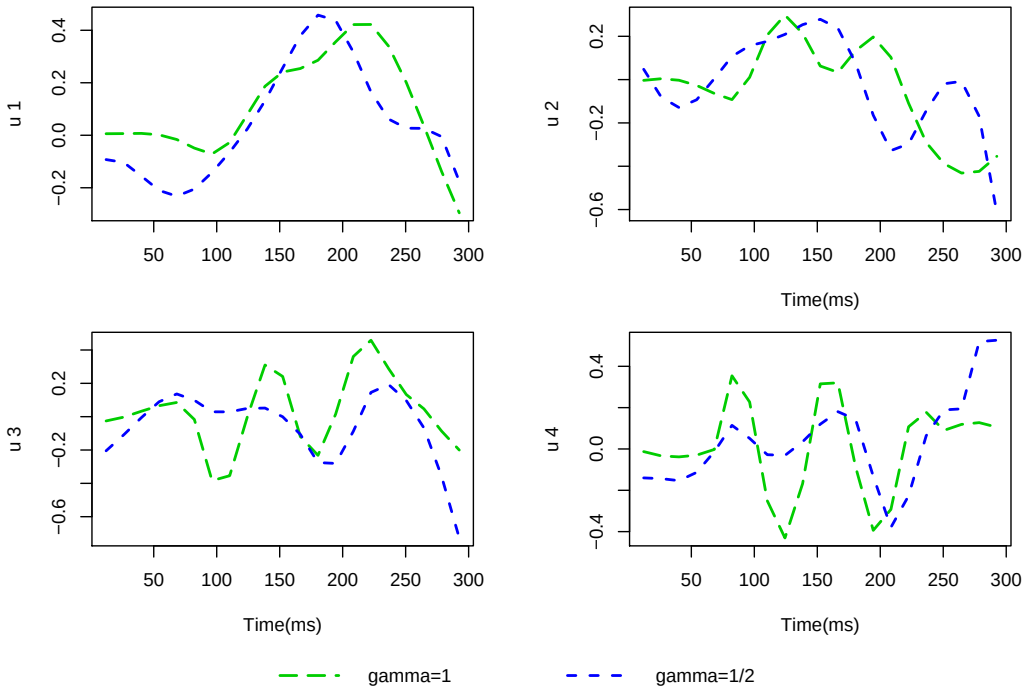


Figure 8.5: Estimated basis functions of LDO for P1 waves of children with normal hearing, ages 0 to 1 years old.

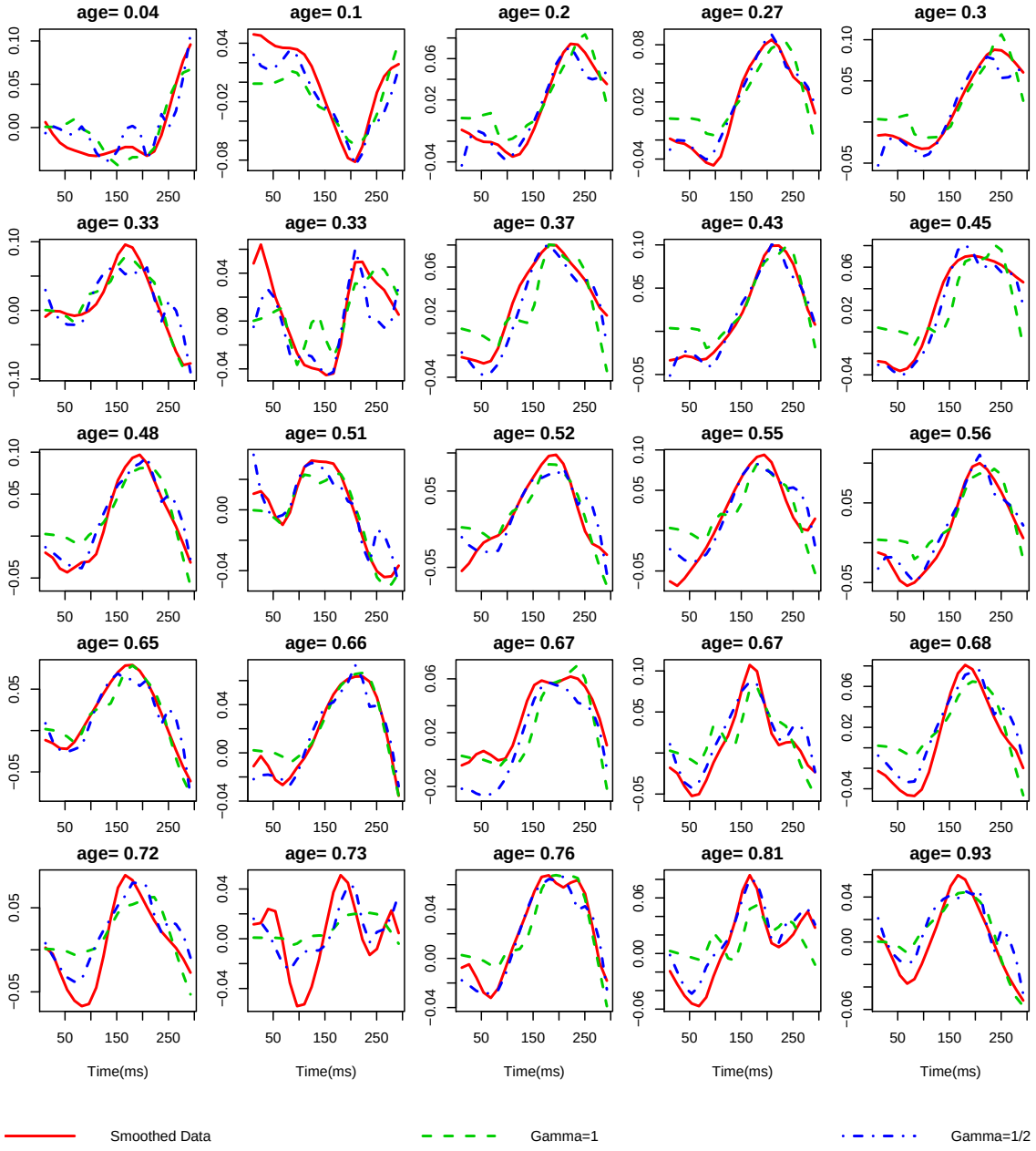
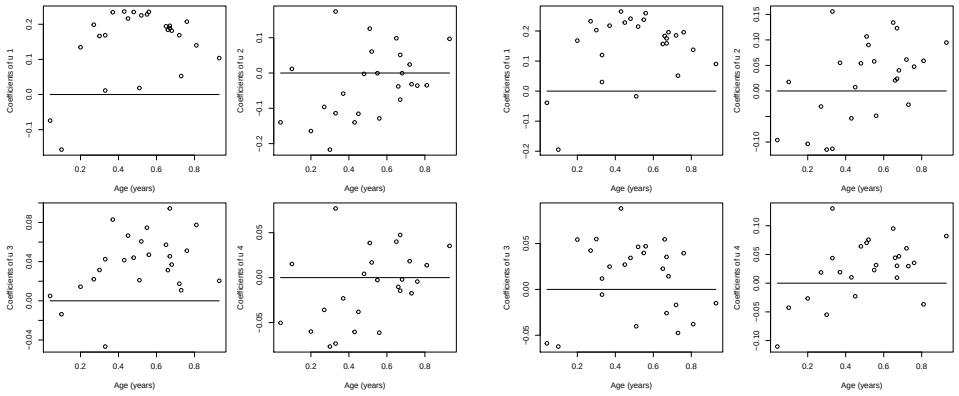


Figure 8.6: Fits of P1 waves of children with normal hearing, ages 0 to 1 years old. Data preprocessed with $\lambda = 10^7$; estimated coefficients of the LDO smoothed with $\lambda_{cv} = 10^{-3}$ for $\gamma = 1$, and with $\lambda_{cv} = 10^5$ for $\gamma = 1/2$.



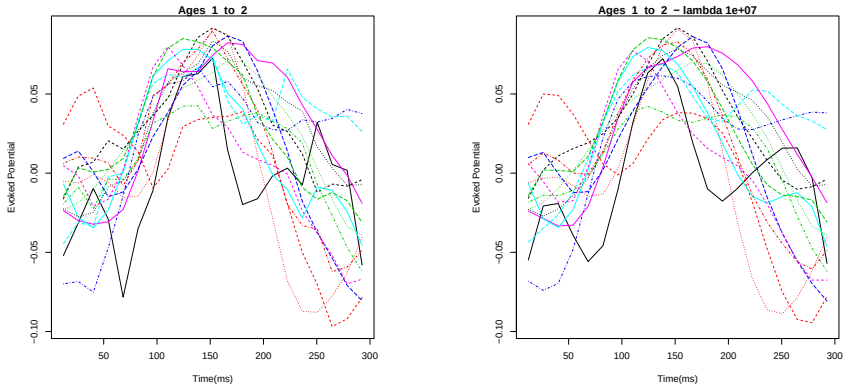
(a) $\gamma = 1$ (b) $\gamma = 1/2$

Figure 8.7: Coefficients of estimated basis functions for P1 waves of children with normal hearing between 0 and 1 years old.

or $\gamma = 1/2$ provide better fits overall. Finally, the coefficients of the estimated basis function in Figure 8.11 do not seem to follow a regular pattern, except that the coefficients of $\hat{u}_1$ vary around a nonzero constant, and the others seem to vary around zero. As in the previous age group, $\hat{u}_1$ is describing the P1 wave, and $\hat{u}_2$, $\hat{u}_3$ and $\hat{u}_4$ are capturing variability in the shape of the P1 wave across children within this age group.

8.0.3 Ages 2 to 4

This subgroup consists of 21 curves, shown in Figure 8.12; no gross outliers can be observed. The LDO coefficients were smoothed with $\lambda_{cv} = 10^5$ for both values of γ. In this age group the estimated basis functions for $\gamma = 1$ and $\gamma = 1/2$ are more similar to each other than those from the younger children subgroups (see Figure 8.13). However, just a few fits are close to the data, and again, it is not possible to state whether one set of fits (those corresponding to $\gamma = 1$ or $\gamma = 1/2$) is better. Also in this age group, we can observe that in the scatterplots of the



(a) (b)

Figure 8.8: P1 waves in CAEP curves of children with normal hearing, ages 1 to 2 years old. (a) 15 original curves; (b) smoothed with smoothing parameter $\lambda = 10^7$.

coefficients of the basis functions (Figure 8.15), the coefficients of $\hat{u}_1$ vary around a nonzero constant, while the others vary around zero. A few of the scatterplots show general patterns (for example, most of the coefficients of $\hat{u}_2$ and $\hat{u}_4$ in Figure 8.15a are negative for ages 2 to 3, and positive for ages 3 to 4), suggesting that a more detailed analysis is needed.

8.0.4 Ages 4 to 7

This subgroup consists of 13 curves, shown in Figure 8.16; one gross outlier can be readily noticed. The estimates of the LDO coefficients were smoothed with $\lambda_{cv} = 10^5$ for both values of γ. The estimated basis functions for $\gamma = 1$ and $\gamma = 1/2$ have shapes that are the most similar among all subgroups, as can be observed in Figure 8.17, suggesting that there are fewer outliers in this age group. The estimated basis corresponding to $\gamma = 1/2$ provide better fits than those corresponding to $\gamma = 1$; note that in Figure 8.19, the first curve (labeled "age=4.17") could be considered an outlier. In Figure 8.19 we are not able to detect any obvious patterns in the

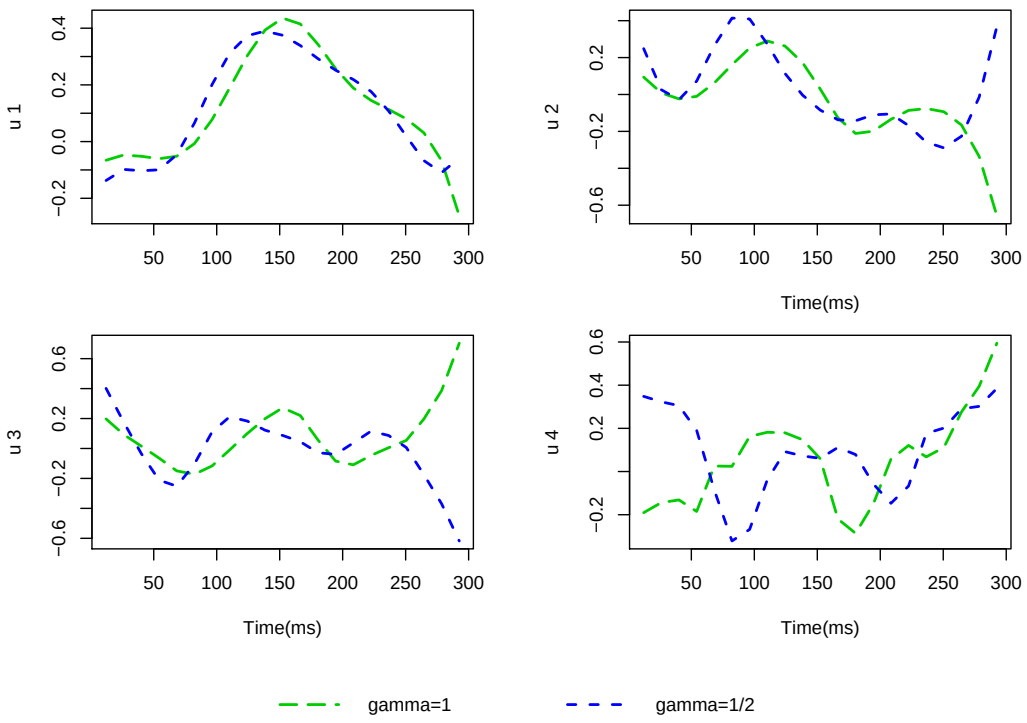


Figure 8.9: Estimated basis functions of LDO for P1 waves of children with normal hearing, ages 1 to 2 years old.

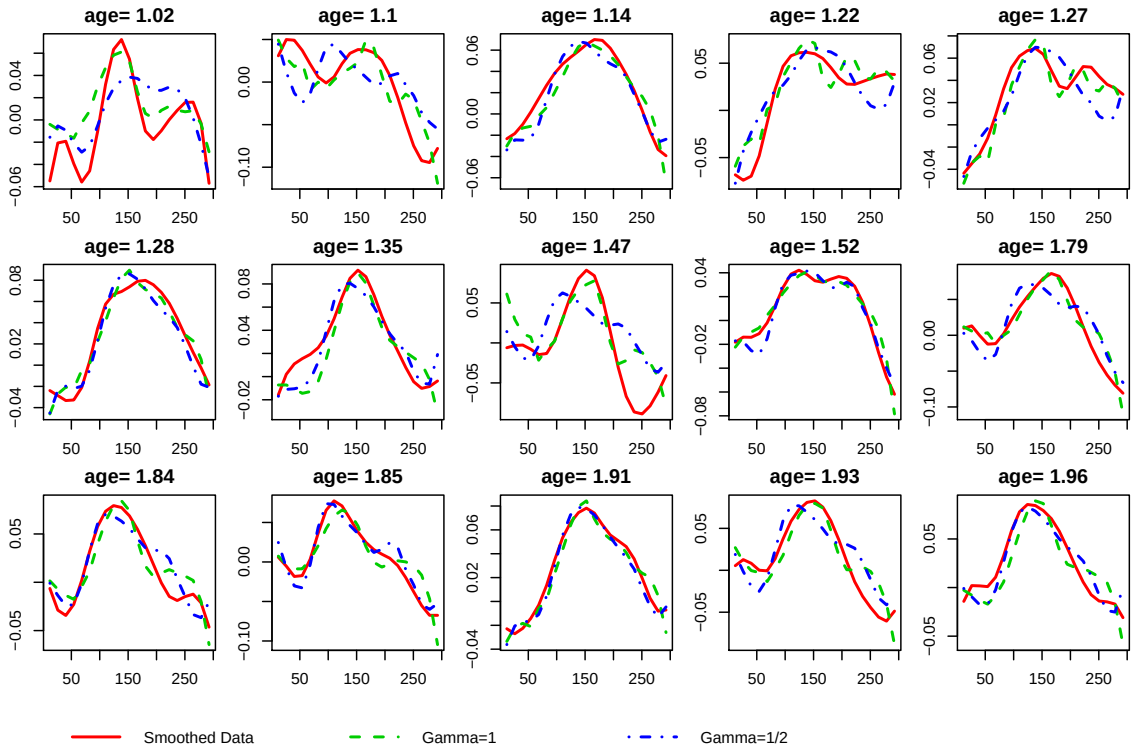
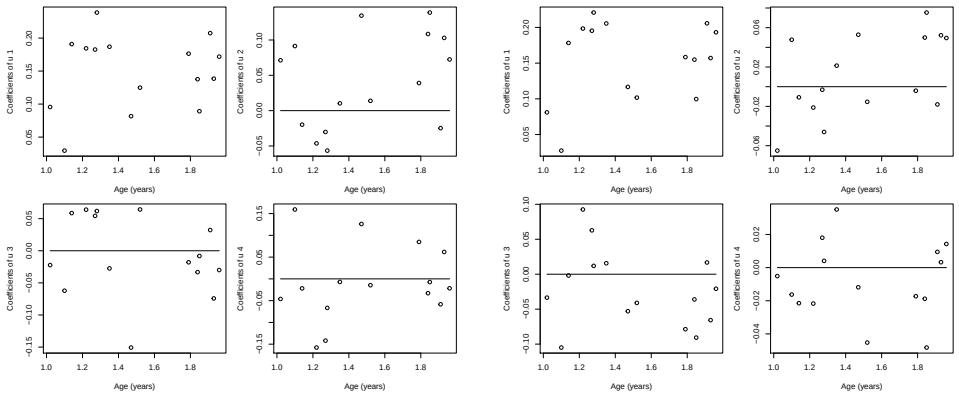
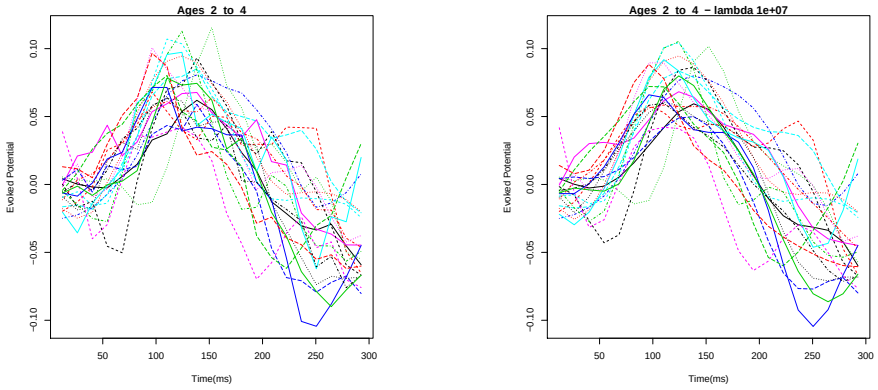


Figure 8.10: Fits of CAEP curves of children with normal hearing, ages 1 to 2 years old. Data was preprocessed with $\lambda = 10^7$; estimated coefficients of the LDO were smoothed with $\lambda_{cv} = 10^{-2}$ for $\gamma = 1$, and with $\lambda_{cv} = 10^6$ for $\gamma = 1/2$.



(a) $\gamma = 1$ (b) $\gamma = 1/2$

Figure 8.11: Coefficients of estimated basis functions for P1 waves of children with normal hearing between 1 and 2 years old.



(a) (b)

Figure 8.12: P1 waves in CAEP curves of children with normal hearing, ages 2 to 4 years old. (a) 21 original curves; (b) smoothed with smoothing parameter $\lambda = 10^7$.

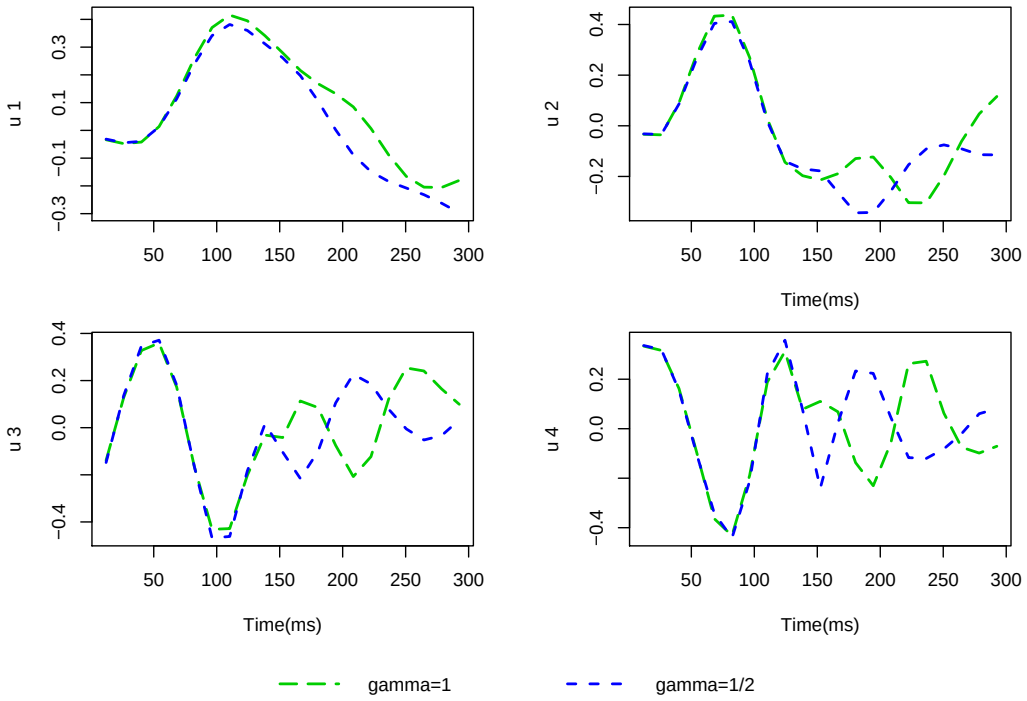


Figure 8.13: Estimated basis functions of LDO for P1 waves of children with normal hearing, ages 2 to 4 years old.

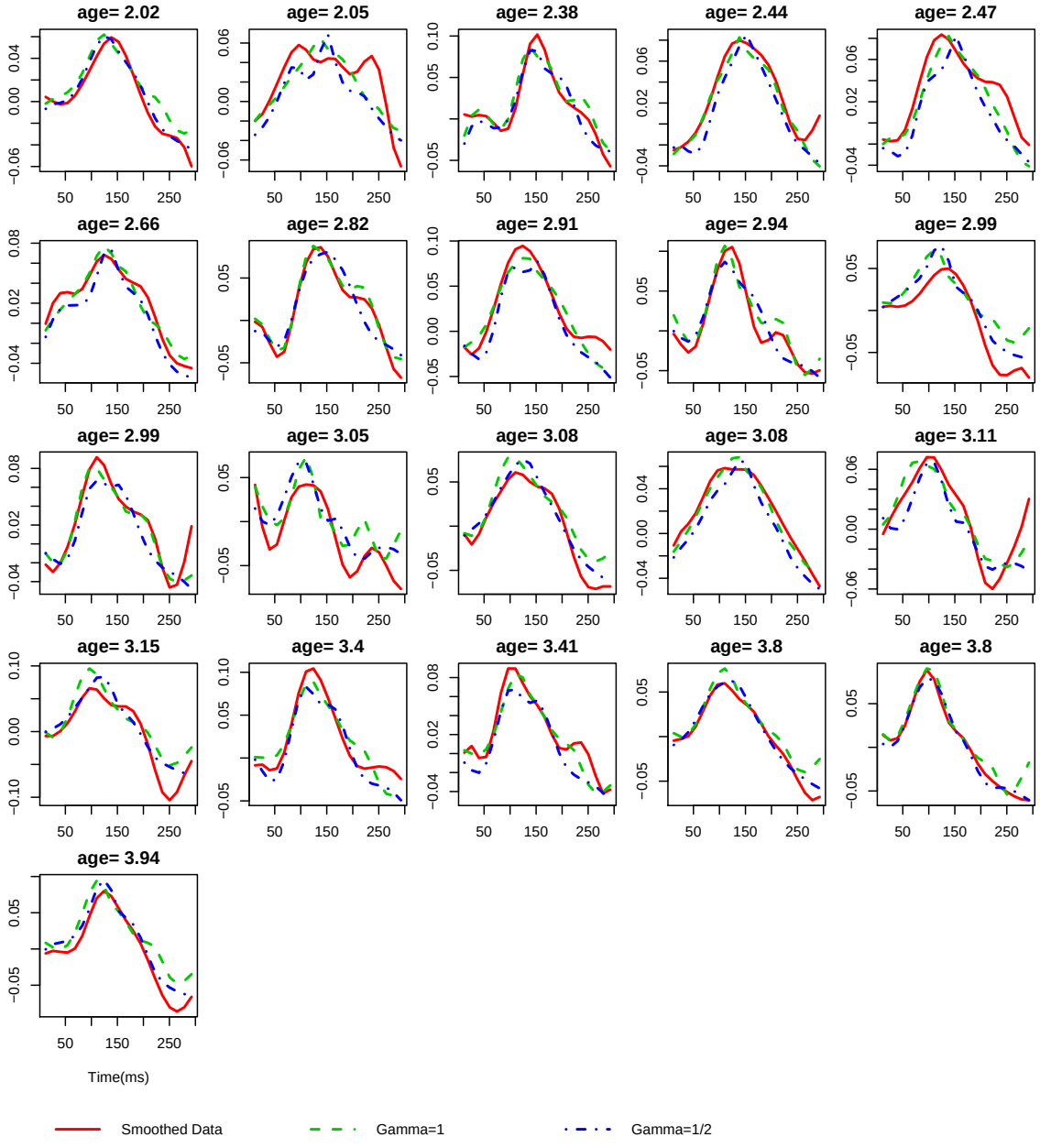
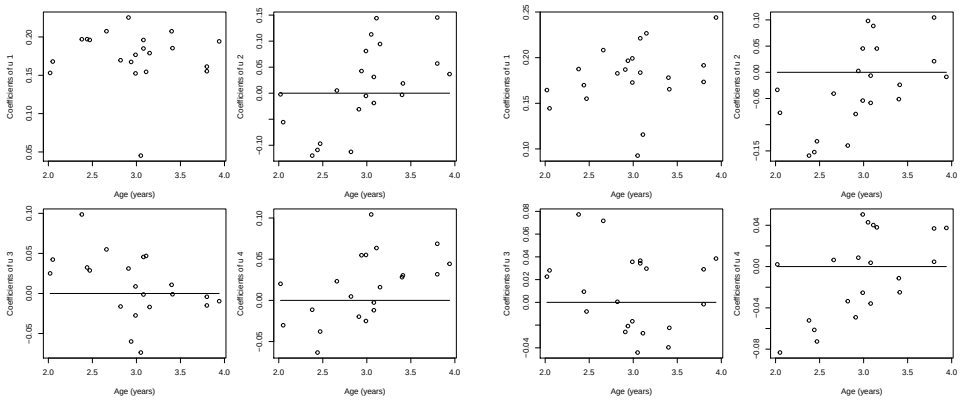


Figure 8.14: Fits of P1 waves of children with normal hearing, ages 2 to 4 years old. Data was preprocessed with $\lambda = 10^7$; estimated coefficients of the LDO were smoothed with $\lambda_{cv} = 10^5$ for $\gamma = 1$ and $\gamma = 1/2$.

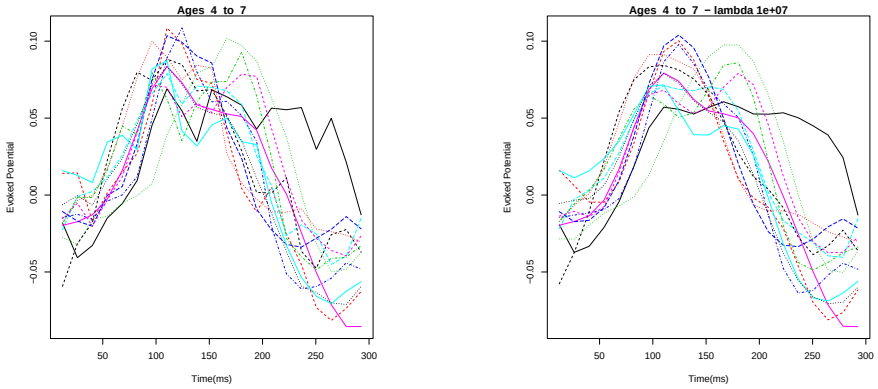


(a) $\gamma = 1$ (b) $\gamma = 1/2$

Figure 8.15: Coefficients of estimated basis functions for P1 waves of children between 2 and 4 years old.

scatterplots for the analysis using $\gamma = 1/2$, except that, as in all other subgroups , the coefficients of $\hat{u}_1$ vary around a nonzero constant, and the others vary around zero.

Figure 8.20 illustrates how the estimated basis functions change with age. Given that $\hat{u}_1$ captures the peak of the P1 curves, note that the responses for children younger than 2 are delayed with respect to the subgroups of children older than 2 years old. The basis functions corresponding to $\gamma = 1/2$ for the subgroups 2 to 4 and 4 to 7 years old look very similar, suggesting that these two age groups may be modeled with the same LDO. The behavior of the basis functions for the two subgroups 0 to 1, and 1 to 2 years old are quite different from the other subgroups; a local PDA analysis would be helpful in better understanding the changes in the curves that can be attributed to age.



(a) (b)

Figure 8.16: P1 waves in CAEP curves of children with normal hearing, ages 4 to 7 years old. (a) 13 original curves; (b) smoothed with smoothing parameter $\lambda = 10^7$.

8.1 Conclusions

A theorem from analysis establishes that the coefficients of the LDO are in the Sobolev space, and thus can be approximated by B-splines. Current PDA software used to estimate the LDO as given in Equation 4.1 assumes that $\omega_m = 1$, and approximates the LDO coefficients by B-splines, even though the Sobolev space is not closed under division. The method proposed in this document relaxes the condition that $\omega_m = 1$; it is inspired by results in linear regression (Frees, 1991 and Wu, 1986), and it has a statistical motivation and foundation in Theorem 3 (Wu (1986), p. 1267). The proposed method calculates the annihilating LDO for each sample of m curves out of the full collection of n curves, and estimates the coefficients of the LDO with the linear combination of the $\binom{n}{m}$ LDO's estimated for each sub-problem. Theorem 2 provides justification for breaking up the problem into $\binom{n}{m}$ sub-problems.

In applying the proposed method to several data sets, two estimates of the LDO were computed, one corresponding to a least-squares estimator (when $\gamma =$

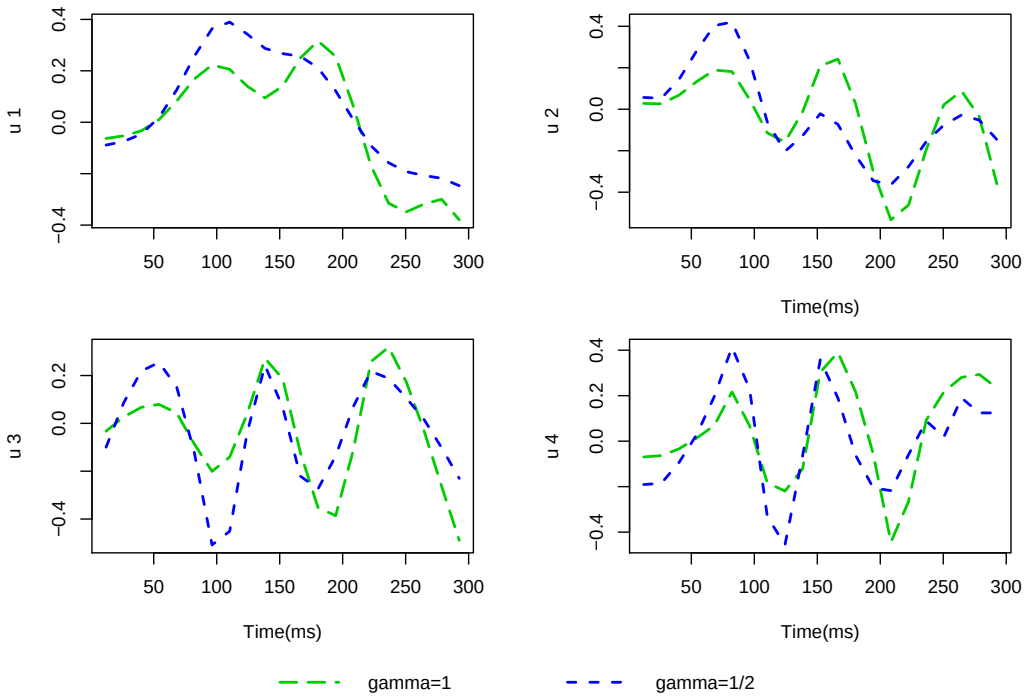


Figure 8.17: Estimated basis functions of LDO for P1 waves of children with normal hearing, ages 4 to 7 years old.

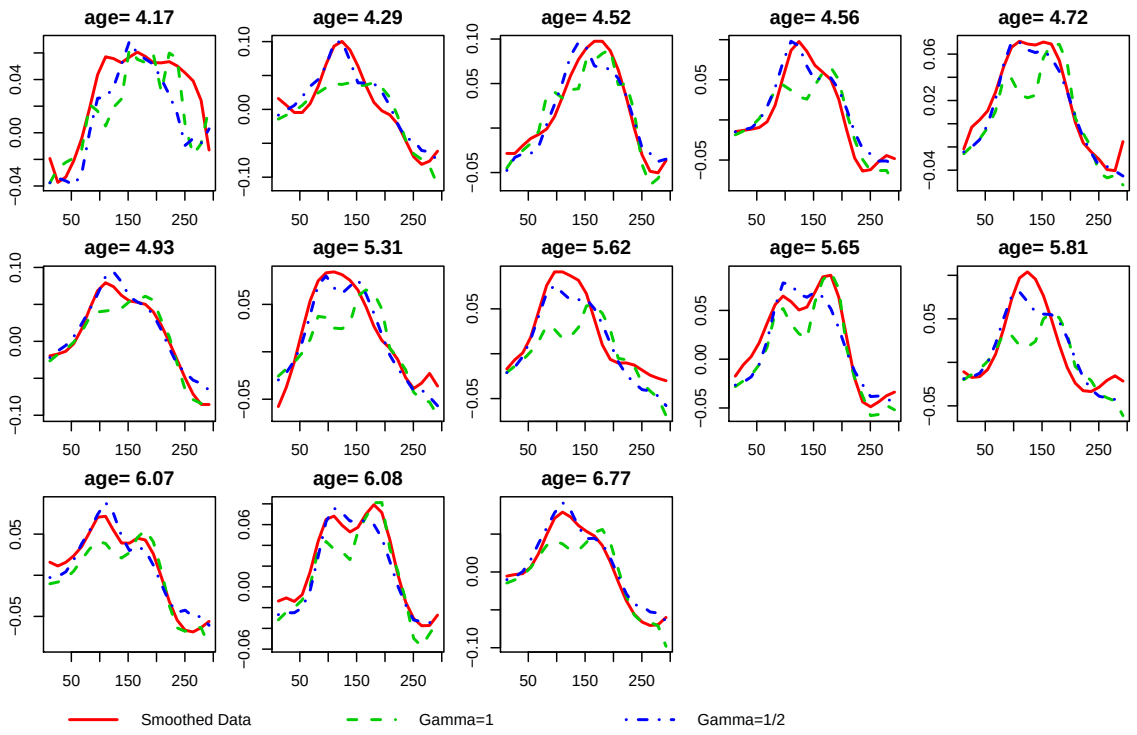
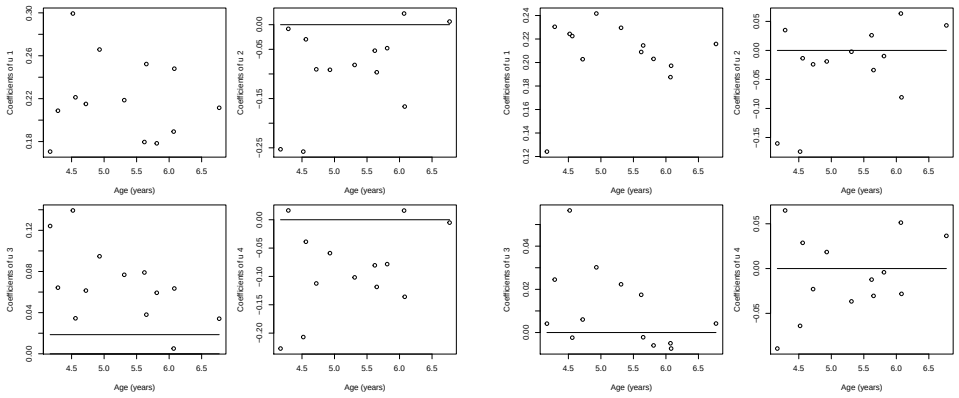


Figure 8.18: Fits of CAEP curves of children with normal hearing, ages 4 to 7 years old. Data was preprocessed with $\lambda = 10^7$; estimated coefficients of the LDO were smoothed with $\lambda_{cv} = 10^5$ for $\gamma = 1$ and $\gamma = 1/2$.



(a) $\gamma = 1$ (b) $\gamma = 1/2$

Figure 8.19: Coefficients of estimated basis functions for P1 waves of children between 4 and 7 years old.

1), and the second providing a robust estimator (when $\gamma = 1/2$) that consistently produced better fits to a collection of curves that included possible outliers. We reported the effects of smoothing the data in a pre-processing step, and of smoothing the estimated coefficients of the LDO. The PDA analysis of the P1 waves in the CAEP curves by subgroups gives some insights into the similarities and differences among age groups and suggests that a local PDA analysis would be helpful in studying differences. This local PDA is described under Future Directions.

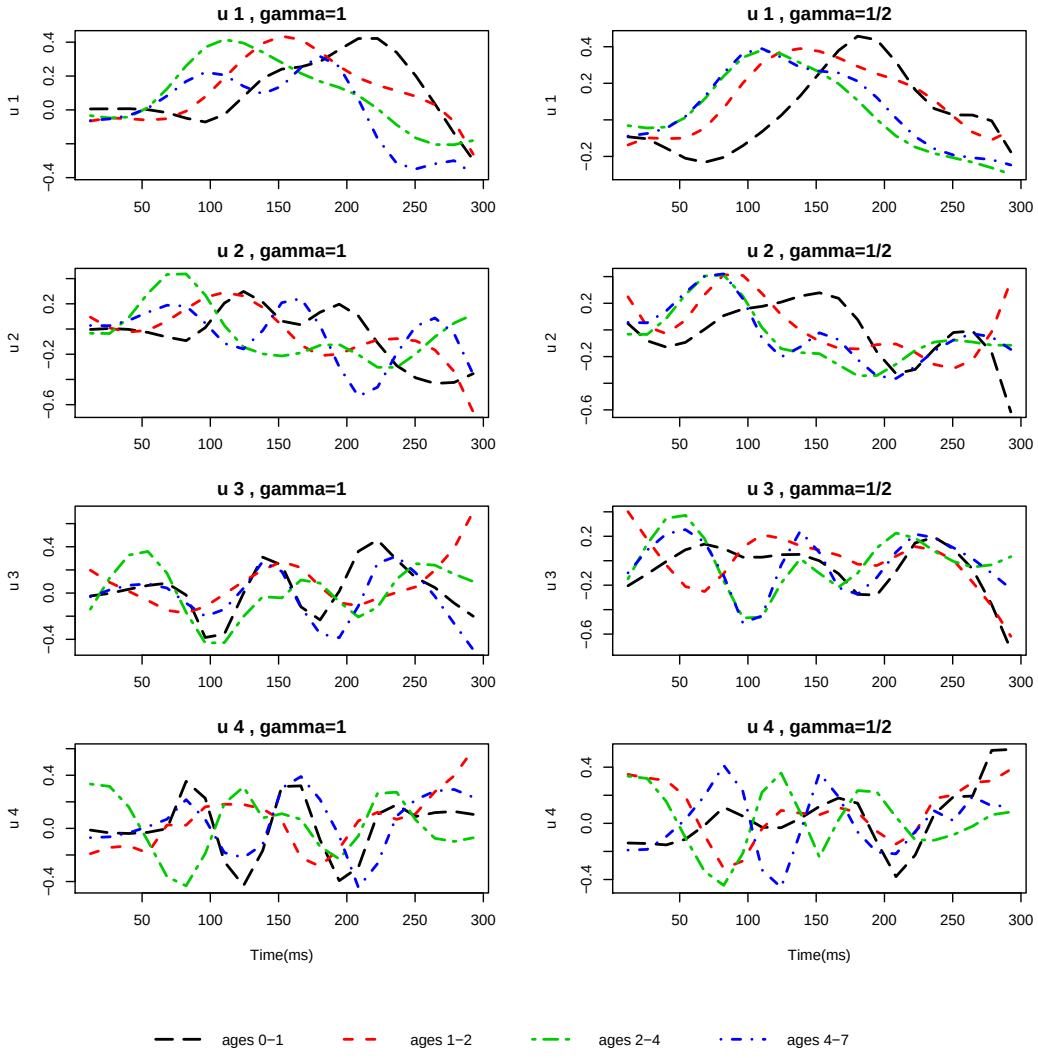


Figure 8.20: Estimated basis functions of LDO for P1 waves of children with normal hearing, by age subgroups.

8.2 Future Directions

Future work would consist of implementing strategies to alleviate the computational burden of $\binom{n}{m}$ sub-problems, and adding a local PDA estimate to study how the functional observations change with a covariate.

Subsamples

The computational burden generated by processing $\binom{n}{m}$ combinations may be addressed by considering $n_s (n_s << \binom{n}{m})$ samples of size m from the n curves, without replacement. This feature is already incorporated in the code of the proposed method, but no thorough analysis was conducted on a data set.

Local PDA

Including a weight function for local pda would essentially reduce the number of curves in the analysis and would assist in studying how the curves change with the covariate. Local PDA selects a subset of the n curves with covariance values $v_1, \ldots, v_n$ that are close in value to the targetted covariate value v_0. According to Wu (1986), p. 1267, the theorem can be extended to cover weighted least squares. Set $K = \text{diag}(k_i, \ldots, k_n)$, where $k_i = k(v_0, v_i; b)$ are kernel weights (Eubank (1999), p. 156) assigned for a local PDA using a subset of the curves with covariate values v_i near v_0. Let K_s denote the submatrix corresponding to set s. Then the local least squares solutions is given by

$$\tilde{\boldsymbol{\omega}}^{\circ}{}_s(t) = \left[\mathbf{X}_s^T(t) K_s \mathbf{X}_s(t)\right]^{-1} \mathbf{X}_s^T(t) K_s \mathbf{z}_s(t)$$

and

$$\tilde{\boldsymbol{\omega}}^{\circ}(t) = \frac{\sum_r \det\left[\mathbf{X}_s^T(t) K_s \mathbf{X}_s(t)\right] \tilde{\boldsymbol{\omega}}^{\circ}{}_s(t)}{\sum_r \det\left[\mathbf{X}_s^T(t) K_s \mathbf{X}_s(t)\right]}.$$

The robust estimator for the LDO without the constraint $\omega_m = 1$ is $\hat{\boldsymbol{\omega}}^{\circ}(t;\gamma) = \sum_r \boldsymbol{\omega}_s^{\circ}(t:\gamma)$, where

$$\boldsymbol{\omega}_s^{\circ}(t;\gamma) = \left(\det\left[\mathbf{X}_s^T(t) K_s \mathbf{X}_s(t)\right]\right)^{\gamma} \begin{bmatrix} \tilde{\boldsymbol{\omega}}^{\circ}{}_s(t) \\ 1 \end{bmatrix}$$

Parallel Computing

Whether all the combinations or subsets are used, the process of estimating the coefficients of the LDO and the cross-validation selection of the smoothing parameter are amenable to parallelism, and can be straightforwardly decomposed into tasks to be computed simultaneously. The resampling method code would need to be modified to incorporate parallel computing, taking advantage of existing R libraries.

Appendix A

Source Files

A.1 resampling.pda Demonstration Script

```
#
# ------------------------------------------------------------------------
#  Resampling PDA Demo
# ------------------------------------------------------------------------
#
#  This script illustrates the use of the resampling pda functions
#  using the growth data included in the fda R package, by Ramsay et al.
#
#  Principal Differential Analysis seeks to identify a Linear
#  Differential Operator L of order m:
#           L = w_0 I + w_1 I + ... + w_m D^m,
#  that satifies Lx=0 as closely as possible for each functional
#  observation x in a collection.
#
# ------------------------------------------------------------------------

##### Required R packages

  library(fda)
  library(MASS)
  library(limSolve)
  library(deSolve)

  source("resampling_pda_functions.r")

# ------------------------------------------------------------------------
#   Function arguments
# ------------------------------------------------------------------------
#
#  t - vector with values of independent variable
```

```
#  y - matrix with values of dependent variable
#  morder - order of the LDO
#
#  wgamma - exponent of estimator \hat{w}^wgamma;
#           value 1 selects the least squares estimator;
#           any other value selects the robust estimator
#
#  prelambda - smoothing parameter to smooth the data
#              before processing
#  postlambda - user-provided smoothing parameter to smooth
#               the estimated coefficients.
#
#  cv.postlambda - logical variable; if TRUE, leave-one-out
#                  cross-validation is performed. Produces
#                  plot of cross-validation curve.
#  cv.range - range over which to perform cross-validation
#  cv.trim - percentage to trim from each boundary to
#            minimize boundary effect on cross-validation
#  ------------------------------------------------------------

#  ------------------------------------------------------------
#   Initial values
#  ------------------------------------------------------------
t <- growth$age
ym <- growth$hgtm
yf <- growth$hgtf

morder <- 4
wgamma <- 1

prelambda <- 1e0      #these two values provide good fits for m=2 and m=4
postlambda <- 1e-7

cv.postlambda <- FALSE
cv.range <- (10^(seq(-15,5)))
cv.graph <- TRUE
cv.trim <- 0.1

#  ------------------------------------------------------------
#   The function resampling.pda  wraps all the steps
#   described in the script resampling_pda_demo_steps.r.
#   The function resampling.pda returns
```

```
#
#   y.list - list of y and its m derivatives
#   w.est - list of list of estimated coefficients and
#           cross-validation matrix
#   w.list - list of estimated coefficients for specified wgamma and
#           cross-validation matrix
#   w.sm.fd - functional object containing smoothed estimated coefficients
#   w.sm.grid - smoothed estimated coefficients evaluated at grid t
#   ssforce - sum of squared norm of forcing functions
#   u.umat - matrix with basis functions in columns
#
# ------------------------------------------------------------------------

   grm <- resampling.pda(t,ym,morder,wgamma,prelambda,postlambda,
                                        cv.postlambda,cv.range,cv.trim)

   grf <- resampling.pda(t,yf,morder,wgamma,prelambda,postlambda,
                                        cv.postlambda,cv.range,cv.trim)

# ------------------------------------------------------------------------
#   If daskp does not converge with minimum value of cross validation -
#   then smoothing parameter for estimated coefficients is too low.
#   Set lambda.coef after examining cv curve, and smooth the coefficients
# ------------------------------------------------------------------------

  w.list.m <- grm$w.est$w.l.g1
  w.list.f <- grf$w.est$w.l.g1

 # lambda.coeff <- 1e-7
  w.sm.m <- list()
  w.sm.f <- list()
  w.sm.m <- smooth.wm(t,w.list.m,lambda.coeff)
  w.sm.f <- smooth.wm(t,w.list.f,lambda.coeff)
  w.sm.fd.m <- w.sm.m$w.fd
  w.sm.fd.f <- w.sm.f$w.fd

# ------------------------------------------------------------------------
#   Last step - Find basis functions
# ------------------------------------------------------------------------

  u.m <- deSolve_m(t,grm$y.list,w.sm.fd.m)
  u.f <- deSolve_m(t,grf$y.list,w.sm.fd.f)
```

```
# ------------------------------------------------------------------------
#   Compare basis functions for males and females
# ------------------------------------------------------------------------

par(mfrow=c(2,2),pty="m",ask=FALSE,mar=c(3,5,2,2))
index <- 1:morder
for (i in index)
  matplot(t,cbind(u.m$umat[,i],u.f$umat[,i]), xlab=ifelse(i<3, "", "Age(years)"),
          ylab=paste("u",i),type="l",col=c(4,2),lty=c(5,2),lwd=c(2,2))
legend(-25,-.5, ncol=2, c("males","females"), col=c(4,2),lty=c(5,2),
       lwd=c(2,2), cex=1,bty="n")

# ------------------------------------------------------------------------
#   Obtain fits for curves corresponding to males and females
# ------------------------------------------------------------------------

par(mfrow=c(5,5),pty="s",ask=FALSE,mar=c(0.8,0.5,0.5,0.5),
    oma=c(7,.7,0,0),xpd=NA)
index <- 1:ncol(ym)
for (i in index) {
  xhat <- grm$y.list[[1]][,i] - lsfit(umat, grm$y.list[[1]][,i], int=FALSE)$residuals
   matplot(t, cbind(grm$y.list[[1]][,i],xhat), type="l", col=c(1,4),
           lty=c(1,2),lwd=c(2,2), xlab="", ylab="", main="")
 }
legend(-45,0, ncol=2, c("Data (M)","Fitted"), col=c(1,4),
       lty=c(1,2),lwd=c(2,2),cex=1,bty="n")
#
par(mfrow=c(5,5),pty="s",ask=FALSE,mar=c(0.8,0.5,0.5,0.5),
    oma=c(7,.7,0,0),xpd=NA)
index <- 1:ncol(yf)
for (i in index) {
  xhat <- grf$y.list[[1]][,i] - lsfit(umat, grf$y.list[[1]][,i],int=FALSE)$residuals
  matplot(t, cbind(grf$y.list[[1]][,i],xhat), type="l", lty=c(1,2),
          lwd=c(2,2), xlab="", ylab="", main="")
}
legend(-45,0, ncol=2, c("Data (F)","Fitted"), col=c(1,2),
       lty=c(1,2),lwd=c(2,2),cex=1,bty="n")

# ------------------------------------------------------------------------
#   Obtain boxplots for coefficients of basis functions
# ------------------------------------------------------------------------
```

```
ucoefm<-c()
index <- 1:ncol(ym)
for (i in index) {
  tempfit <- lsfit(grm$u.umat, grm$y.list[[1]][,i], int=FALSE)
  ucoefm <- rbind(ucoefm,tempfit$coef)
}

ucoeff<-c()
index <- 1:ncol(yf)
for (i in index) {
  tempfit <- lsfit(grf$u.umat, grf$y.list[[1]][,i], int=FALSE)
  ucoeff <- rbind(ucoeff,tempfit$coef)
}

colnames(ucoefm) <- c("u1","u2","u3","u4")
colnames(ucoeff) <- c("u1","u2","u3","u4")
par(mfrow=c(1,2),pty="m",ask=FALSE, oma=c(7,0,0,0),mar=c(4,4,1,1),xpd=NA)
  boxplot(ucoefm)
  title("Coefficients of u_i, males")
  boxplot(ucoeff)
  title("Coefficients of u_i, females")
```

resampling_pda_demo.r

A.2 resampling.pda Step-by-Step Demonstration Script

```
#
# ------------------------------------------------------------------
#  Resampling PDA Demo
# ------------------------------------------------------------------
#
#  This script illustrates the use of the resampling pda functions
#  using the growth data included in the fda R package, by Ramsay et al.
#  Refer to the file XXX.pdf for a more detailed description
#  of the functions and the method's motivation.
#
#  Principal Differential Analysis seeks to identify a Linear
#  Differential Operator L of order m:
#          L = w_0 I + w_1 I + ... + w_m D^m,
#  that satifies Lx=0 as closely as possible for each functional
#  observation x in a collection.
#
# ------------------------------------------------------------------

##### Required R packages

  library(fda)
  library(MASS)
  library(limSolve)
  library(deSolve)

  source("resampling_pda_functions.r")

# ------------------------------------------------------------------
#   Initial variables
# ------------------------------------------------------------------
#
#  t - vector with values of independent variable
#  y - matrix with values of dependent variable
#  morder - order of the LDO
#
#  wgamma - exponent of estimator \hat{w}^wgamma;
#           value 1 selects the least squares estimator;
#           any other value selects the robust estimator
#
#  prelambda - smoothing parameter to smooth the data
```

```
#               before processing
#  postlambda - user-provided smoothing parameter to smooth
#               the estimated coefficients.
#
#  cv.postlambda - logical variable; if TRUE, leave-one-out
#                  cross-validation is performed.
#  cv.range - range over which to perform cross-validation
#  cv.trim - percentage to trim from each boundary to
#            minimize boundary effect on cross-validation
# ------------------------------------------------------------------

t <- growth$age
y <- growth$hgtm

morder <- 4
wgamma <- 1

prelambda <- 1e-2      #these two values provide good fits for m=2 and m=4
postlambda <- 1e-7

cv.postlambda <- FALSE
cv.range <- (10^(seq(-15,5)))
cv.graph <- TRUE
cv.trim <- 0.1

# ------------------------------------------------------------------
#  Find y',...,y^m from functional object
# ------------------------------------------------------------------

y.list <- create.fobj(t,y,morder,prelambda)

# ------------------------------------------------------------------
#  Estimate LDO coefficients
# ------------------------------------------------------------------
#
# ------------------------------------------------------------------
#  Tip: if calculating w.est takes a long time,
#  user may want to save w.est, so future
#  analysis and calculations may be done
#  without starting from scratch.
#  For example: saveRDS(w.est,file="growth_m4_prel1e-12.RData")
# ------------------------------------------------------------------
```

```
w.est <- list()
w.est <- estimate.w(y.list,sampling=FALSE,n.samples)

if (wgamma==1) {
  w.list <- w.est$w.l.g1
  cv.w.list <- w.est$cv.w.l.g1
} else {
  w.list <- w.est$w.l.g5
  cv.w.list <- w.est$cv.w.l.g5
}

# ------------------------------------------------------------------
#   Cross-validation if cv.postlambda=TRUE
# ------------------------------------------------------------------

if (!cv.postlambda){
  lambda.coeff <- postlambda
} else {cv.out <- estimate.cv.lambda(y.list,w.list,cv.w.list,cv.range,
                                     cv.trim,cv.graph)
        lambda.coeff <- cv.out$lambda.cv
        cv.forcing.mat <- cv.out$cv.forcing
        }

# ------------------------------------------------------------------
#   Smooth estimated coefficients
#   either with user-provided postlambda,
#   or with global minimum of cross-validation
#   curve. User may want to examine the
#   cv curve and alternatively choose local minima and/or
#   values along region plateaus.
# ------------------------------------------------------------------

w.sm <- list()
w.sm <- smooth.wm(t,w.list,lambda.coeff)
w.sm.grid <- w.sm$w.grid
w.sm.fd <- w.sm$w.fd
ssforce <- ssnres(y.list,w.sm.grid)

# ------------------------------------------------------------------
#   Last step - Find basis functions
# ------------------------------------------------------------------
```

```
#   If daspk does not converge, the estimated coefficients should be
#   smoothed with a larger value of lambda.coeff, or the data should be
#   preprocessed with a larger smoothing parameter.
#

u <- deSolve_m(t,y.list,w.sm.fd)

# ------------------------------------------------------------------------
#   The function resampling.pda  wraps all the steps
#   described up to this point.
#   The function call for resampling.pda is
#
#   gr.demo <- resampling.pda(t,y,morder,wgamma,prelambda,postlambda,
#                             cv.postlambda,cv.range,cv.trim)
#
#  See the file resampling_pda_demo.r for more details
# ------------------------------------------------------------------------
#
# ------------------------------------------------------------------------
#   Once the basis functions are found, they may be used,
#   for example, to find fits of the curves.
# ------------------------------------------------------------------------

umat <- u$umat
n.curves <-ncol(y)

par(mfrow=c(5,4),pty="s",ask=FALSE,mar=c(0.8,0.5,0.5,0.5),
    oma=c(7,.7,0,0),xpd=NA)
index <- 1:n.curves
for (i in index) {
  xhat <- y.list[[1]][,i] - lsfit(umat, y.list[[1]][,i], int=FALSE)$residuals
  matplot(t, cbind(y.list[[1]][,i],xhat), type="l", lty=c(1,2),lwd=c(2,2),
          xlab="", ylab="", main="")
}
legend(-1.5,-10, ncol=2, c("Smoothed Data","Fitted"), col=c(1,2),
       lty=c(1,2),lwd=c(2,2),cex=1,bty="n")
```

resampling_pda_demo_steps.r

A.3 R language functions

```
create.fobj <- function(t, y, morder, prelambda){
  n.curves <- ncol(y)
  t.points <- length(t)
  t.range <- range(t)

  basisfd <- create.bspline.basis(t.range, nbasis=length(t), norder = (morder+2))
  y.fd <- smooth.basisPar(t, y, fdobj=basisfd, Lfdobj=int2Lfd(morder),
                          lambda=prelambda,method="qr")$fd

  par(mfrow=c(1,2),pty="m",ask=FALSE,mar=c(3,3,3,1))
  matplot(t,y, main="Data", type="l")
  plot(y.fd,t, main=paste("Data smoothed with lambda=",prelambda), xlab="x",
       ylab="y",cex=1.2,ylim=c(min(y),max(y)))

  grids.mat <- matrix(NA,t.points,n.curves)
  y.l <- rep(list(grids.mat),(morder+1))
  for(i in 1:(morder+1)){
    y.l[[i]] <- eval.fd(t,y.fd,Lfd=(i-1))
    }
  return(y.l)
}

ssnres <- function(y.list,lw.list)
{
  m <- length(y.list)
  n.curves <- ncol(y.list[[1]])
  t.points <- nrow(y.list[[1]])
   total.ss <- rep(NA,n.curves)
  forcing.mat <- matrix(0,nrow=t.points,ncol=n.curves)
  for(i in (1:n.curves))
  {
    temp.term <- rep(0,t.points)
    for (j in (1:m)){
      temp.term <- temp.term + (lw.list[[j]])*(y.list[[j]][,i])
      forcing.mat[,i] <- forcing.mat[,i] + (lw.list[[j]])*(y.list[[j]][,i]) }
    total.ss[i] <- sum(temp.term^2)
  }
  return(list(forcing.mat=forcing.mat,ss.total=sum(total.ss)))
}

```

```
smooth.wm <- function(t,w.list,lambda.coeff){
  morder <- length(w.list) - 1
  t.range <- range(t)
  basisfd <- create.bspline.basis(t.range, nbasis=length(t), norder = (morder+2))
  w.fd <- list()
  w.grid <- list()
  for (i in 1:(morder+1)) {
    w.fd[[i]] <- smooth.basisPar(t, w.list[[i]], fdobj=basisfd, Lfdobj=int2Lfd(morder),
                                 lambda=lambda.coeff,method="qr")$fd;
    w.grid[[i]] <- eval.fd(t,w.fd[[i]])
  }
  return(list(w.grid=w.grid,w.fd=w.fd))
}

deSolve_m <- function(t,y.list,w.fd){

  w.dae <- function(ti,w.fd){
    w.mat <- matrix(0,1)
    w.list.t <- rep(list(w.mat),(morder+1))

    for (i in 1:(morder+1)) {
      w.list.t[[i]] <- eval.fd(ti,w.fd[[i]])
    }
    return(w.list.t)
  }

  daefun <- function(ti,y,yprime,parms) {
    with (as.list(c(y, yprime,parms)),
          {
            w.temp <- w.dae(ti,w.fd)
            res1 <- c()
            res2 <- 0
            for (i in 1:(morder-1)) {
              res1.temp <- get(names(dyini[i])) - get(names(yini[(i+1)]))
              res1 <- c(res1,res1.temp)
              res2 <- res2 + w.temp[[i]]*get(names(yini[i]))
            }
            res2 <- res2 + w.temp[[morder]]*get(names(yini[morder])) +
                           w.temp[[(morder+1)]]*get(names(dyini[morder]))
            return(list(c(res1,res2)))
          })}
```

```
  morder <- length(y.list) - 1
  y.list.initial  <- sapply(lapply(y.list,'[',1,TRUE),mean)

  yini <- y.list.initial[1:morder]
  names(yini) <- paste0("y", seq_along(yini))
  #
  dyini <- y.list.initial[2:(morder+1)]
  names(dyini) <- paste0("dy", seq_along(dyini))

  DEQ <- ode(y = yini, times = t, func=NULL, dy=dyini, parms=NULL, method="daspk",
             res = daefun, maxord = 5, atol = 1e-6, rtol = 1e-6, nalg = 0, estini = 1,tcrit = max(t))

  xp <- DEQ[,1]
  yp <- DEQ[,2:(morder+1)]

  umat <- matrix(0,length(t),morder)

  for (i in 1:morder){
    umat[,i] <- approx(xp, t(yp[,i]), t)$y
    umat[,i] <- umat[,i]/sqrt(sum(umat[,i]^2))
  }
  return(list(xp=xp,yp=yp,umat=umat))
}

estimate.w <- function(y.list,sampling=FALSE,n.samples){

  xs.sample <- function(y.list,morder,t.index,combs.mat,s.index){
    xs <- matrix(y.list[[1]][t.index,combs.mat[1:morder,s.index]],nrow=morder,ncol=1)
    for (i in (2:morder))
    {
      xs <- cbind(xs,y.list[[i]][t.index,combs.mat[1:morder,s.index]])
    }
    return(xs)
  }

  morder <- length(y.list) - 1
  n.curves <- ncol(y.list[[1]])
  t.points <- nrow(y.list[[1]])

  if(!sampling){
    combs.mat <- combn(n.curves,morder)
  } else {
```

```
    combs.mat <- matrix(NA,morder,n.samples)
    for (i in 1:n.samples){
        combs.mat[,i] <- sample(c(1:n.curves),size=morder,replace=FALSE)
    }
}

w.mat <- matrix(0,t.points)
cv.mat <- matrix(0,t.points,n.curves)
w.l.g1 <- rep(list(w.mat),(morder+1))
w.l.g5 <- rep(list(w.mat),(morder+1))
cv.w.l.g1 <- rep(list(cv.mat),(morder+1))
cv.w.l.g5 <- rep(list(cv.mat),(morder+1))

for (t.index in 1:t.points){
  for (k in 1:ncol(combs.mat)) {
    xs.mat <- xs.sample(y.list,morder,t.index,combs.mat,k)
    zs <- -c(y.list[[morder+1]][t.index,combs.mat[1:morder,k]])
    M2.t <- abs(det(xs.mat))
    inv.t <- (Solve(xs.mat)%*%zs)
    ws.t.g1 <- inv.t*(M2.t^2)
    ws.t.g5 <- inv.t*M2.t
    w.l.g1[[morder+1]][t.index] <- w.l.g1[[morder+1]][t.index] + (M2.t)^(2)
    w.l.g5[[morder+1]][t.index] <- w.l.g5[[morder+1]][t.index] + (M2.t)
    cv.w.l.g1[[morder+1]][t.index,combs.mat[,k]] <-
                    cv.w.l.g1[[morder+1]][t.index,combs.mat[,k]] + (M2.t)^(2)
    cv.w.l.g5[[morder+1]][t.index,combs.mat[,k]] <-
                    cv.w.l.g5[[morder+1]][t.index,combs.mat[,k]] + (M2.t)
    for (i in 1:morder) {
      w.l.g1[[i]][t.index] <- w.l.g1[[i]][t.index] + ws.t.g1[i,1]
      w.l.g5[[i]][t.index] <- w.l.g5[[i]][t.index] + ws.t.g5[i,1]
      cv.w.l.g1[[i]][t.index,combs.mat[,k]] <-
                    cv.w.l.g1[[i]][t.index,combs.mat[,k]] + ws.t.g1[i,1]
      cv.w.l.g5[[i]][t.index,combs.mat[,k]] <-
                    cv.w.l.g5[[i]][t.index,combs.mat[,k]] + ws.t.g5[i,1]
    }
  }
}
wm.norm <- sqrt(sum((w.l.g1[[morder+1]])^2)/length(w.l.g1[[morder+1]]))
w.l.g1 <- mapply('/',w.l.g1,wm.norm,SIMPLIFY = FALSE)
wm.norm <- sqrt(sum((w.l.g5[[morder+1]])^2)/length(w.l.g5[[morder+1]]))
w.l.g5 <- mapply('/',w.l.g5,wm.norm,SIMPLIFY = FALSE)
return(list(w.l.g1=w.l.g1,w.l.g5=w.l.g5,cv.w.l.g1=cv.w.l.g1,cv.w.l.g5=cv.w.l.g5))
```

```
}

estimate.cv.lambda <- function(y.list,w.list,cv.w.list,cv.range,cv.trim,cv.graph){

  cv.ss <- function(cv.lambda,cv.trim,y.list,cv.loo.list)
  {
    morder <- length(y.list) - 1
    n.curves <- ncol(y.list[[1]])
    t.points <- nrow(y.list[[1]])
    cv.forcing.mat <- matrix(0,t.points,n.curves)
    for(i in 1:n.curves)
    {
      temp.list <- lapply(cv.loo.list,'[',TRUE,i)
      sm.temp.list <- smooth.wm(t,temp.list,cv.lambda)$w.grid
      for (k in (1:(morder+1))){cv.forcing.mat[,i] <- cv.forcing.mat[,i] +
                                  sm.temp.list[[k]]*(y.list[[k]][,i])}
     }
    return(cv.forcing.mat)
  }

  morder <- length(y.list) - 1
  n.curves <- ncol(y.list[[1]])
  t.points <- nrow(y.list[[1]])
  cv.mat <- matrix(0,t.points,n.curves)
  cv.loo.list <- rep(list(cv.mat),(morder+1))
  cv.forcing <- array(NA, dim=c(t.points,n.curves,length(cv.range)))
  for(i in 1:(morder+1)) {cv.loo.list[[i]] <- as.vector(w.list[[i]])-cv.w.list[[i]]}

  # Cross Validation along cv.range
  cv.smooth <- c()
  for(i in 1:length(cv.range))
  {
    cv.temp <- cv.ss(cv.range[i],cv.trim,y.list,cv.loo.list)
    cv <- apply((cv.temp[round(cv.trim*t.points):round((1-cv.trim)*t.points),])^2,2,sum)
    cv.smooth <- c(cv.smooth,sum(cv))
    cv.forcing[,,i] <- cv.temp
  }

  lambda.cv <- cv.range[which.min(cv.smooth)]
  print(paste("cv global minimum:", lambda.cv))

  par(mfrow=c(1,1),pty="s")
```

```
  plot(log10(cv.range),cv.smooth,type="l",xlab=expression(paste("log(",lambda,")")),
       ylab=expression(paste("CV(",lambda,")")))

  return(list(lambda.cv=lambda.cv,cv.forcing=cv.forcing,cv.smooth=cv.smooth))
}
resampling.pda <- function(t,y,morder,wgamma=1,prelambda,postlambda,cv.postlambda=TRUE,
                           cv.range=(10^(seq(-15,12))),cv.trim=0.1){

  n.curves <- ncol(y)
  n.combs <- choose(n.curves,morder)
  t.points <- length(t)

  y.list <- create.fobj(t,y,morder,prelambda)

  w.est <- list()
  w.est <- estimate.w(y.list,sampling=FALSE,n.samples)
  if (wgamma==1) {
    w.list <- w.est$w.l.g1
    cv.w.list <- w.est$cv.w.l.g1
  } else {
    w.list <- w.est$w.l.g5
    cv.w.list <- w.est$cv.w.l.g5
  }

  if (!cv.postlambda){
    lambda.coeff <- postlambda
  } else {cv.out <- estimate.cv.lambda(y.list,w.list,cv.w.list,cv.range,cv.trim,cv.graph)
  lambda.coeff <- cv.out$lambda.cv
  cv.forcing.mat <- cv.out$cv.forcing}

  w.sm <- list()
  w.sm <- smooth.wm(t,w.list,lambda.coeff)
  w.sm.grid <- w.sm$w.grid
  w.sm.fd <- w.sm$w.fd
  ssres <- ssnres(y.list,w.sm.grid)

  u<-deSolve_m(t,y.list,w.sm.fd)

  return(list(y.list=y.list,w.est=w.est,w.list=w.list,w.sm.fd=w.sm.fd,w.sm.grid=w.sm.grid,
              ssforce=ssres$sstotal,u.xp=u$xp,u.yp=u$yp,u.umat=u$umat))
}
```

resampling_pda_functions.r

References

Boor, C. de (1978). *A Practical Guide to Splines*. New York: Springer-Verlag.

Casella, A. and R. L. Berger (2002). *Statistical Inference*. Duxbury Advanced Series. Australia; Pacific Grove, California: Thomson Learning.

Coddington, A. and N. Levinson (1955). *Theory of Ordinary Differential Equations*. International series in pure and applied mathematics. New York: Tata McGraw-Hill.

Dau, Torsten et al. (2000). "Auditory brainstem responses with optimized chirp signals compensating basilar-membrane dispersion". In: *The Journal of the Acoustical Society of America* 107.3, pp. 1530–1540.

Eilers, Paul H. C. and Brian D. Marx (1996). "Flexible Smoothing with B-splines and Penalties". In: *Statistical Science* 11.2, pp. 89–121.

Eubank, R.L. (1999). *Nonparametric Regression and Spline Smoothing, Second Edition*. Statistics: A Series of Textbooks and Monographs. New York: Taylor & Francis.

Frees, Edward W. (1991). "Trimmed slope estimates for simple linear regression." In: *Journal of Statistical Planning and Inference* 27, pp. 203–221.

Jin, S., J.G. Staniswalis, and I. Mallawaarachchi (2013). "Principal Differential Analysis with a Continuous Covariate: Low Dimensional Approximations for Functional Data." In: *Journal of Statistical Computation and Simulation* 83.10, pp. 1964–1980.

Messer, K. (1991). "A comparison of a spline estimate to its equivalent kernel estimate". In: *The Annals of Statistics* 19, pp. 817–829.

Press, William H. et al. (1992). *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge, MA: Cambridge University Press.

— (2007). *Numerical Recipes: The Art of Scientific Computing*. Cambridge, MA: Cambridge University Press.

Ramsay, J. (1996). “Principal Differential Analysis: Data Reduction by Differential Operators”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58.3, pp. 495–508.

Ramsay, J. and B.W. Silverman (1997). *Functional Data Analysis*. Springer Series in Statistics. New York: Springer.

Sharma, A., M.F. Dorman, and A.J. Spahr (2002). “A sensitive period for the development of the central auditory system in children with cochlear implants: implications for age of implantation.” In: *Ear and Hearing* 23.6, pp. 532–539.

Silverman, B. (1984). “Spline smoothing: the equivalent variable method.” In: *The Annals of Statistics* 12, pp. 898–916.

Staniswalis, J.G., C. Dodoo, and A. Sharma (2015). “Local Principal Differential Analysis: Graphical Methods for Functional Data with Covariates.” In: *Communications in Statistics Simulation and Computation*.

Wahba, G. (1990). *Spline Models for Observational Data.* CBMS-NSF Regional Conference Series in Applied Mathematics, 59. Philadelphia, PA: SIAM.

Wu, C. F. J. (1984). “Jackknife and bootstrap inference in regression and a class of representations for the LSE”. In: *Wisconsin Univ-Madison, Mathematics Research Center* MSR-TSR-2675.

— (1986). “Jackknife, Bootstrap and Other Resampling Methods in Regression Analysis”. In: *The Annals of Statistics* 14.4, pp. 1261–1295.

Curriculum Vitae

María Ivette Barraza was born in Mexico City on October 14, 1963. A few years after her family moved to Chihuahua, she entered into The University of Texas at El Paso and completed a B.S degree in Mathematics, and a M.S. degree in Statistics. In 1987 she became the first director of the Math Computer Learning Laboratory, funded by a MSIP grant, whose PI was Dr. Eugene F. Schuster.

Since then María has worked in different capacities in the Department of Mathematical Sciences and in the College of Science. She is a UTEP-certified translator from/to English to/from Spanish, and has taught the Introduction to Translation class a few times.

María enrolled in the Computational Science Ph.D. program in the fall of 2009, while keeping her current full-time position as Systems Administrator in the Department of Mathematical Sciences. Her research was done under the supervision of Dr. Joan G. Staniswalis.

Contact Information: mbrios@utep.edu

This dissertation was typed by María Ivette Barraza.