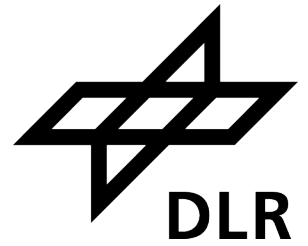




Ludwig-Maximilians-University of Munich
Department of Statistics

German Aerospace Center
Institute of Robotics and Mechatronics



Statistical Analysis of Joystick Trajectories

Master Thesis by Marius Wagner

Supervision:

Karan Sharma, M.Sc., Claudio Castellini, Ph.D.
Institute of Robotics and Mechatronics
German Aerospace Center (DLR)

Dr. Fabian Scheipl
Department of Statistics
Ludwig-Maximilians-University of Munich

Submission Date: 15th May 2017

Abstract The research field of affective computing aims to improve human-machine interaction. One of the main goals is to enable autonomous systems to recognize and adapt to human emotions. Machine learning is able to find an attribution between physiological reactions and underlying emotions. In order to provide labelled training data, human subjects annotate emotional stimuli in experimental studies. Three major challenges of the resulting continuous annotation data are: 1. Finding a suitable representation of this complex data, 2. Comparing the annotations of different subjects, 3. Combining the annotations to provide reliable ground truth for machine learning. Since previous research did not take into account the continuous nature of the annotation data, a functional data approach is introduced: Annotations are represented as smooth functions in a low-dimensional functional eigenspace. Comparison and ground truth estimation is then performed using simple statistical methods.

Contents

List of Figures	v
1 Introduction	1
1.1 The Role of Emotions in Human-Machine Interaction	1
1.2 Annotation Strategies for Emotions	2
1.3 Joystick-based Continuous Annotation of Emotion	4
1.4 Methodological Challenges of Continuous Annotations	9
1.5 Aim of the Thesis	12
2 From Discrete Measurements to Functions	15
2.1 Estimation of Functions	16
2.1.1 Preliminaries for a functional representation	17
2.1.2 Basis Representation	19
2.1.3 B-Splines	21
2.1.4 Penalized Splines	28
2.2 Estimating Multivariate Multi-Subject Annotation Functions using P-Splines	34
3 Statistical Analysis of Annotation Functions	41
3.1 Principal Component Analysis for Functional Data	41
3.1.1 PCA	42
3.1.2 FPCA	43
3.1.3 FPCA for Multivariate Functions	44
3.2 Applying multivariate FPCA on the annotation data	53
3.2.1 Number of functional principal components	53
3.2.2 The resulting eigenfunction space	54
4 Ground Truth through Characteristic Annotations	61
4.1 Number of components	61
4.2 Outlier removal	62
4.3 Characteristic Annotations	63
5 Discussion and Outlook	67
Bibliography	73

List of Figures

1.1	Annotation UI	6
1.2	Annotation process during the video stimuli	6
1.3	All annotations made by one subject	8
1.4	15 Annotations for different video stimuli	8
1.5	Possible domains for statistical modeling of observation data	11
1.6	Structure of the thesis	12
2.1	Sampling times	19
2.2	Single B-Spline basis functions	22
2.3	Complete B-Spline Basis	24
2.4	Components of a B-Spline representation	27
2.5	B-Spline fits for varying basis sizes	28
2.6	P-Spline fits for varying smoothing parameters	32
2.7	Relation between the GCV criterion and the smoothing level	35
2.8	Relationship between function and derivative in a P-Spline	37
2.9	SSE and computation time for different basis sizes	38
2.10	GCV Criterion for different values of the smoothing parameter	39
3.1	Scree plot of the MFPCA results	54
3.2	Illustration of multivariate fPCs	56
3.3	Heatmap of the similarity of the annotation functions	58
3.4	Discriminative power of the first four dimensions of the eigenfunction space	59
4.1	Scree plot for annotation functions seperately	62
4.2	Multivariate outlier detection and characteristic annotation	64
4.3	Characteristic Annotation	65
5.1	Possible domains for statistical modeling of observation data revised	71

Chapter 1

Introduction

1.1 The Role of Emotions in Human-Machine Interaction

The emergence of an increasing number of autonomous gadgets and electrical devices such as self-driving cars or robots that collaborate closely with humans, brings up numerous new challenges pertaining to effective user interaction and engagement. One of the most important ones is that machines should be able to adapt their behaviour to the users instant by instant and thus to provide a continuously engaging interaction experience. In this context, the possibility to determine and act according to the emotional state of the users is of great importance whenever human-machine interaction is required. For instance, in a multi-robot factory these emotion sensing machines would not only interact with workers, but also prompt them to take some rest when they feel tired, increase the complexity of tasks when they are bored, or initiate an emergency stop procedure as soon as they show signs of evident fear [1].

There has been a growing interest in developing systems and technologies that are able to recognise and interpret the emotional state of the user. This development was enhanced by the book "Affective Computing" [2] of MIT Media Lab Professor Rosalind Picard, who established this research domain two decades ago. The availability of cheap, light-weight and portable sensors further accelerates the progress of the field of research. Physiological reactions such as galvanic skin response, heart rate, respiration rate, muscular activity are used to improve the

human-machine interaction so that an adaption to the users emotions may soon be a common feature of autonomous devices.

But these sensor data need to be attributed to internal human emotions, and this is still a largely unsolved problem: Emotions are latent and inaccessible manifestations of the human nature. Whenever we want to attribute physiological reactions to them, we need to understand which stimuli triggers certain emotions and how they manifest themselves on the physiological level. In order to study this relation scientifically, a reliable annotation of emotional stimuli is required that are presented to human subjects. The attribution between emotion and physiological response can then be established using supervised learning strategies.

1.2 Annotation Strategies for Emotions

The aforementioned association problem led to its own experimental study design. In these studies, human subjects are exposed to emotional stimuli and not only their physiological responses are measured but also their emotional interpretation of the stimuli. These annotations then can be used as labels for the underlying emotions that were evoked due to the stimuli. The labels serve as the ground truth pertaining to the affective experiences. This facilitates the use of statistical learning methods that attribute the data from the physiological response to an emotion [3].

The modalities of the emotional stimuli correspond to the human senses. Researchers often provide stimuli in form of videos [4], music [5] or photos [6] to the participants. The physiological responses of the subjects are recorded using modalities such as biosignals (e.g. heart rate, respiration rate, muscle activity) [7, 1], speech signals [8] and/or computer vision based approaches [9].

In most of the studies, the acquired annotations are in a discrete and time-independent form. That is, the subjects report their affective experience on questionnaires using psychometric

rating scales after they have been exposed to the video stimuli [10, 11]. In this case a participant provides one single rating for the complete duration of the stimuli. In studies, where continuous stimuli (e.g. video/audio clips) are used, a more continuous form of ratings is desirable [12]. This is especially true for video stimuli due to their dynamic nature. They are known to evoke specific emotional responses in a relatively short period of time [13, 12] and the affect-evoking content of the videos can differ for different segments of the clip. To provide annotation tools for continuous stimuli, several rating interfaces that allow for time-continuous self-reporting have been developed in the past decade. The most prominent are FEELTRACE [14], Gtrace [15], Emotion Slider [16], Affect Rating Dial [17] and the EMuJoy [18] frameworks.

When using annotation tools for both discrete and continuous stimuli, an operationalisation of emotion is essential. Despite the lack of consensus on a holistic model of emotion [6], researchers in psychology make use either of the two following models: Discrete emotion classes or dimensional models of emotion [19, 20]. One commonly used continuous model is the two-dimensional circumplex model of affect, also called the valence-arousal model [21]. Whereby, as per to this model, the conscious experience of the raw emotion at any given moment is defined as composed by two main dimensions: A horizontal dimension called valence that ranges from displeasure to pleasure, and a vertical dimension called arousal that ranges from sleepiness to high tension. Hence, by using this model, commonly occurring emotional states, such as scared, pleased, relaxed, bored etc., can be expressed in terms of coordinates of valence and arousal in a two-dimensional space [22]. Therefore, the valence-arousal model is widely adopted by the affective computing community, where researchers use tools based on this model to acquire annotations/ratings from participants about their affective experiences during a study [3].

Irrespective of the plethora of tools that exist for annotations, several problems still exist. Using annotation tools that use a one-dimensional operationalisation (Affect Rating Dial), the subject is not able to report valence and arousal simultaneously [17]. This problem is easily

reduced by two-dimensional annotation tools but they also have deficiencies. For instance, both FEELTRACE and Gtrace are based on a computer-mouse interface and require the subject to continuously press the mouse button to perform annotations. The same applies for EMuJoy, which requires the users to press a mouse button to report their climax experiences [18]. This requirement of continuously pressing a button can be adverse to the usability of these systems, since it increases the physical and cognitive load for the users [23, 3]. Based on these and other usability issues associated with the use of computer-mouse interfaces, several works propose the use of joystick based systems [18, 3].

1.3 Joystick-based Continuous Annotation of Emotion

To overcome the shortcomings mentioned, a joystick-based annotation system was developed at the Institute of Robotics and Mechatronics of the German Aerospace Center [24]. The users continuously annotate their affect state by moving or holding the joystick in the different regions of a user-interface (UI). The different regions of the UI imply different affect states that are characterised by distinct valence and arousal levels. The UI design is based on the aforementioned valence-arousal affect model [25] (see Figure 1.1, left). One difference between the original valence-arousal model and the UI used in [18], is an UI extension through the use of self-assessment-manikins (SAM) [10] to the coordinate axes of the interface. The manikin figures depict different valence (on x-axis) and arousal (on y-axis) levels and serve as useful guides for the participants while they report their affect state [10]. Also, the SAMs are non-language dependent indicators of valence and arousal levels, since their addition to the interface is an improvement over tools where users are guided using english based description of valence and arousal states e.g. the label "scary" might be interpreted differently by people with different language and cultural backgrounds [6]. The left panel of Figure 1.1 shows the resulting annotation UI.

The usage of a joystick over other input peripherals (e.g a computer-mouse or a keyboard) is motivated by the following factors:

- In comparison to previously mentioned mouse-based systems, the users do not have to continuously press any button to report their affect state.
- The joystick features a return spring that automatically realigns the joystick when the user has stopped reporting [18]. The return spring also provides a form of simple force-feedback to the user.
- A joystick is by design more ergonomic than a standard computer mouse, as both the hand and wrist positions are at a neutral angle when holding a joystick.
- The location and movement of the pointer in a UI for a joystick is relative to the centre (i.e. neutral) position, whereas in a computer-mouse they are always relative to last location of the pointer. Thus, for a joystick the user can roughly estimate the position of the pointer without having to visually locate it on the UI. This is not the case for a computer mouse. Therefore, the cognitive load while using the joystick should be presumably lower than using a computer-mouse for annotations.
- The use of joystick brings an element of gamification into the annotation process, thus probably improving user enjoyment [3].

During the experiment, the UI is displayed in the upper right corner of the video (see Figure 1.1, right panel). This allows the participant to simultaneously view the videos and annotate his affect state. During the experiment the subject is instructed to annotate her/his perceived affective experience by positioning the red cursor in the appropriate region of the UI. When the elicited affect changes, the participant is supposed to change the position of the cursor to the region that best characterizes her/his perceived affect state (as characterised by the valence-arousal levels), thereby continuously annotating her/his affective response to the video.

Figure 1.2 shows a sample result from the recorded data of the joystick-based annotation tool. The x and y coordinate values of the joystick cursor annotate the valence and arousal levels during the course of the continuous stimuli. The resulting trajectory of the previous positions is made visible using a black path. The two-dimensional approach allows the user to annotate the video stimulus freely and without restrictions. For the purpose of comparison, the resulting data can be structured according to subjects (Figure 1.3) or according to the video stimuli

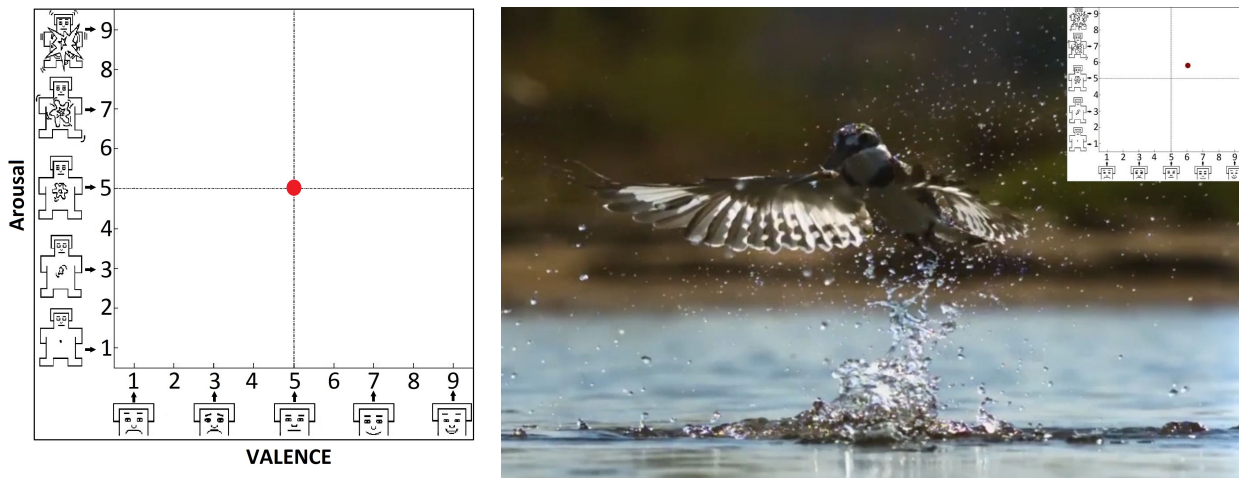


Figure 1.1: The annotation UI (left) and how it was embedded in the video stimuli (right).

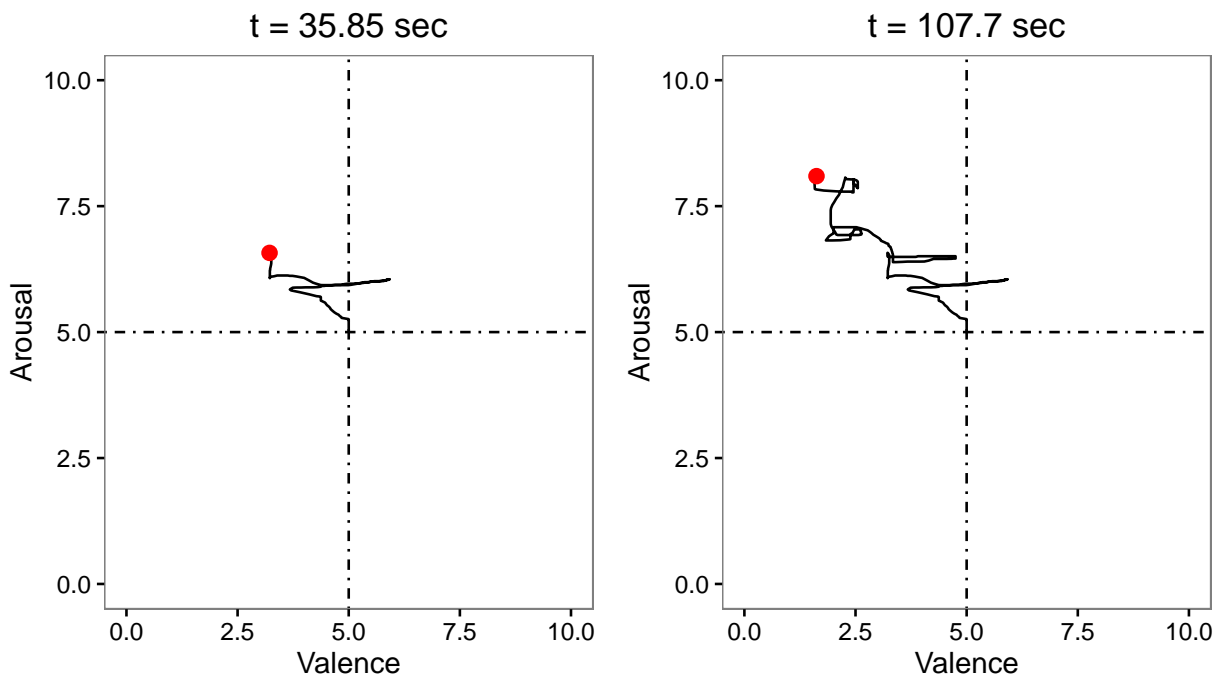


Figure 1.2: The annotations made by a single subject for scary-2 video stimuli. The red dot marks the actual position of the joystick cursor at the timepoint t . The line shows the annotation trajectory that the subject already made.

(Figure 1.4).

Figure 1.3 shows the affect state annotation values of a typical subject for the entire duration of the test. The data consists of the annotations for the video stimuli. The annotations are colour coded according to the emotion that the video stimuli were expected to elicit. For the experiment, the emotion label associated with each video was determined through an initial evaluation, undertaken by a different set of participants [26]. Furthermore, the order of all the video stimuli was randomly shuffled and each subject was exposed to a different sequence. Also, videos eliciting the same type of emotional response would never be shown one after another, and each video was followed by a blue screen to isolate the impact of each video on the affect state of the participant and to avoid carry-over effects. The total number of videos in a sequence was 18, i.e. 8 videos for emotion elicitation (2 for each emotion label), 9 blue screens and 1 video at the start of the sequence.

Figure 1.4 shows a comparison of 15 annotations for the same video stimuli. The left plot shows annotations for an amusing video stimuli, the right plot for a scary video stimuli. The different colors indicate different subjects. By structuring the data in this manner we see that there are signs for a congruent annotation behaviour across individuals. For the amusing video one would assume that this video evokes higher valence and arousal levels which is reflected by the annotation data. The same applies for the scary video which is expected to elicit high arousal but low valence levels.

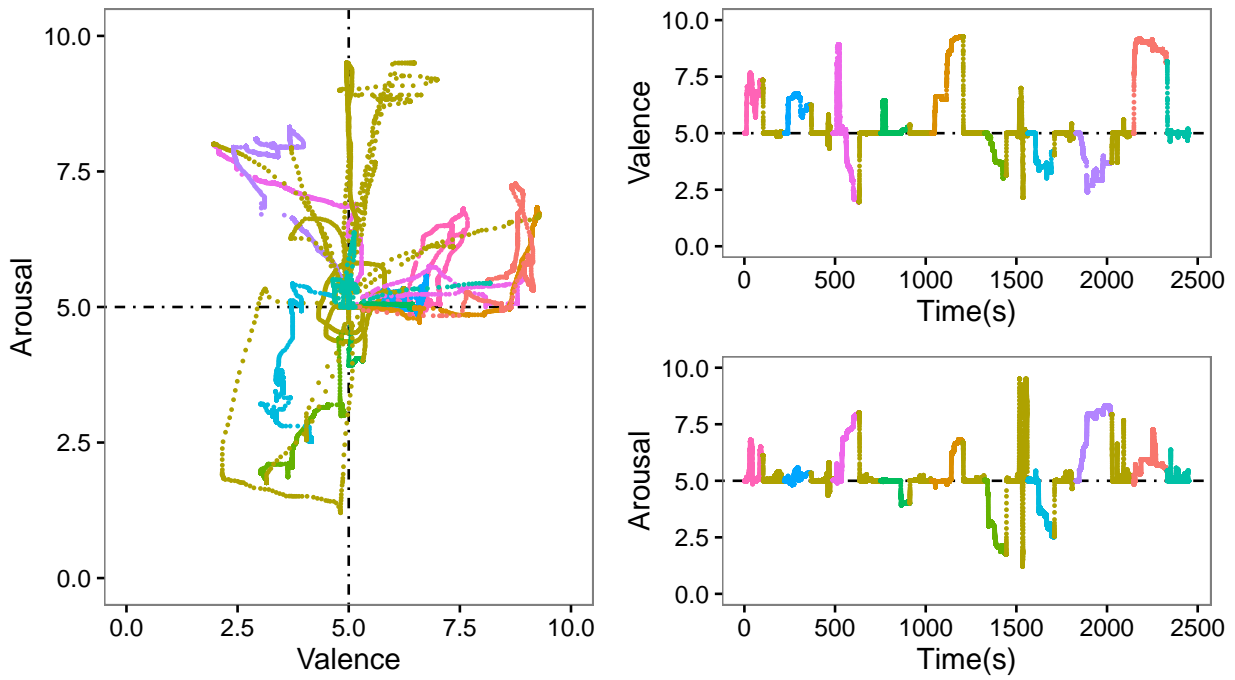


Figure 1.3: All annotations made by one subject. Different colors indicate different video stimuli.

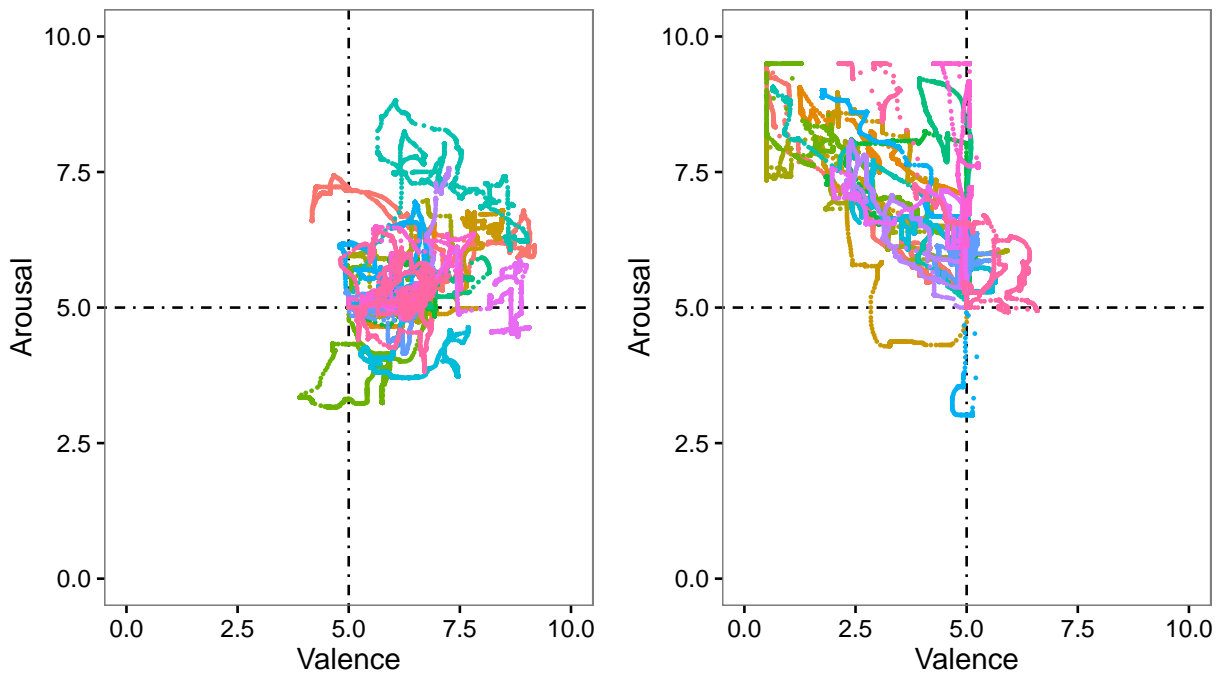


Figure 1.4: Annotations of 15 different subjects for two different video stimuli. Different colors indicate different subjects.

1.4 Methodological Challenges of Continuous Annotations

In order to use continuous annotation data as a means to associate emotions with physiological responses, several challenges have to be addressed. While some of these tasks pertain to the statistical learning of subjective annotations in general [27], major challenges are a result of the continuous nature and complexity of the data [3]. Figures 1.3 and 1.4 illustrate the need for a suitable statistical methodology that allows to organise, analyse and interpret continuous annotations. More precisely, the following challenges have to be addressed:

- *Representing Multiple Subjective Ratings:* There are close connections to longitudinal methods since the data can be regarded as massive repeated measures on each subject [28]. This methodology often models the expected values as explicit low-dimensional functions of time [29]. But it is not guaranteed that it allows enough flexibility to represent the experimental annotation data. Also there are close connections to the analysis of high-dimensional data. A naive approach would be to use the data points as input to a multivariate analysis [30]. However, this would not take the serial structure (of time or space) into account and thus may lose power by neglecting inherent structure or result in non-smooth estimates [29].
- *Comparing Multiple Subjective Ratings:* In order to compare which of the video stimuli yielded the highest levels of valence and arousal another traditional methodology would calculate an average for each of the stimuli over the time course of the experiment and compare those means. Using ANOVA one then could analyse the data and with either planned orthogonal contrasts or post-hoc tests, could ascertain which arithmetic means, if any, differed from the others [30]. Yet, it is obvious from Figure 1.3 that such an average would obscure interesting time-based variations in the annotations [31]. A second traditional approach for the comparison would be to treat each observation as a repeated measure (RM) and conduct an RM ANOVA or RM MANOVA. None of them is satisfactory because of the obvious autocorrelation between successive points: With the subjects using a joystick, observations are not independent because the cursor must pass

through intermediate points on its way from one desired position to another. The data also might violate stationarity. Although one could conduct correlational analyses among subjects, factors such as different reaction times and individual use of the valence-arousal plane could obscure relations in the data. Thus, traditional statistical methodology would allow us to answer only simple questions, and then, only with certain independence assumptions violated [31].

- *Combining Multiple Subjective Ratings:* Any given stimulus is continuously annotated by several participants, resulting in multiple subjective annotations for that stimulus. For further machine learning it is often desired to have one single ground truth annotation which is representative for a set of subjective annotations [32, 33, 3, 27].

To overcome these challenges, another approach is more suitable to the continuous nature of this data. It extends the classical statistical techniques by changing the perspective on the data considering each observation a continuous function [34]. Figure 1.5 illustrates this change of perspective. The upper left corner shows the domain of a data matrix in a standard situation: Normally one would organise the data matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ in a manner that each of the n subjects are paired with each of the p covariates and to each pair a response x_{ij} is assigned as the consequence of the experiment or data collection. Moving down to the lower left corner would correspond to the situation where n is effectively infinite and population characteristics (all possible human subjects) are analysed. This is not of interest here. The lower right corner shows that one may even could have an infinity of subjects or cases to consider. Also this is beyond the scope of this thesis. Moving to the upper right corner is now the interesting case: The number of subjects are fixed and the number of variables p are allowed to increase without limit and even beyond countability so that they define a continuum. This makes it natural that the experiment or data collection of each subject i yields its own function $x_i(t)$. This change of perspective resulted in an own branch of statistics named functional data analysis [35, 36, 37, 38].


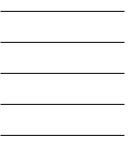
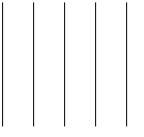
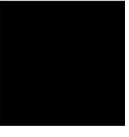
		Number of covariates	
		$p < \infty$	$p = \infty$
Number of observ. entities	$n < \infty$		
	$n = \infty$		
Data		x_{i1}, \dots, x_{ip}	$x_i(t), 0 \leq t \leq T$

Figure 1.5: Possible domains for statistical modeling of observation data referring to [34]. The domain depends on the perspective on the observed data: The number of observation entities (n) can be finite or infinite. The number of covariates (p) that are associated with each observation can be finite or lie on a continuum.

1.5 Aim of the Thesis

The aim of the thesis is twofold: 1. to analyse the data of the joystick-based annotation tool with respect to the aforementioned challenges and 2. to introduce state-of-the-art methodology from functional data analysis for a problem class where each observation is a function of arbitrary dimension. Thus, the work presented in this thesis aims to:

- remove noise from the data and represent several subjective discrete ratings by annotation functions using P-Splines,
- compare complex multivariate annotation functions from multiple subjects using multivariate functional principal component analysis,
- combine these multiple subject ratings in a characteristic annotation for each video stimuli in order to estimate ground truth by exploiting the eigenfunction space of the annotation functions of each video

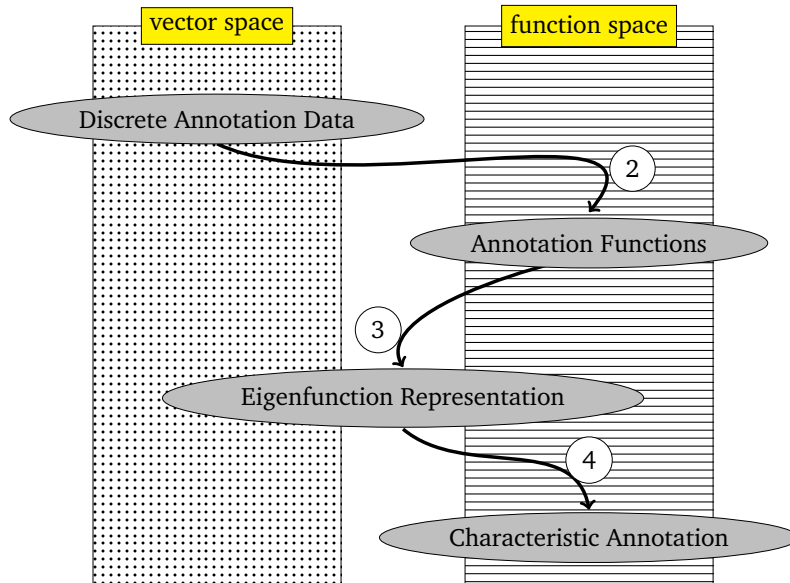


Figure 1.6: Structure of the thesis according to the perspective on the annotation data.

The second goal for the methodology takes up what we already saw from Figure 1.5, the change of perspective on the data. This will be a recurring topic throughout the thesis and the main idea behind the thesis structure as depicted in Figure 1.6. The experimental annotation

data consisting of thousands of discrete valence and arousal pairs is the starting point for the analysis. As common in statistics, it can be represented using a suitable vector space.

In Chapter 2, we introduce how the transition in perspective that was shown in Figure 1.5 can be made. By introducing basis function representation using P-Splines, a non-parametric method that allows not only a functional representation for discrete data is used. It also ensures that the resulting functions have beneficial properties such as reduced noise and high smoothness and also allows estimation of smooth derivatives. By transforming discrete annotations into annotation functions the data is represented using a function space. Chapter 3 explains a framework for simultaneously comparing these annotation functions. Due to their multivariate nature (valence and arousal dimensions) we introduce a very general method that allows the analysis of even much more complex functional observations. The resulting eigenfunction representation of the annotation functions is a hybrid in a sense that it combines discrete and functional representation and exists both in vector and function space. In Chapter 4, we show how to benefit from the theoretical properties of this eigenfunction representation in order to combine the annotation functions of all subjects for estimating a characteristic annotation for each video stimuli. Finally, we will critically discuss the results and the generality of the methodology presented here.

Chapter 2

From Discrete Measurements to Functions

In this chapter we introduce the main ideas of functional representation. It is shown how discrete measurements can be turned into smooth functions. The methodology is illustrated based on the annotation data but its generality allow many other applications.

The foundation for this approach is the use of linear combinations of simple basis functions. This computational technique is well suited for representing information about functions. It also provides the flexibility that is needed for complex data as the annotation trajectories and links it with the computational power to fit even hundred of thousands of data points. Due to its linear structure this approach allows to express the required calculations within the familiar context of simple matrix algebra [35].

The chapter consists of two parts:

- The first part aims to elaborate the P-Spline method that will be used for the annotation data. After some preliminaries the general concept of basis representation is introduced. The focus here lies on a functional representation using basis splines by having a closer look of its components in order to understand how these very simple objects can be used to form a functional representation of rather complex functions. Finally the basis spline approach is extended by a roughness penalty to assure smooth functions.

- In the second part the P-Spline approach is applied to the annotation data to form annotation functions. The parameters for the representation will be discussed by showing the behaviour of different quality criterions.

Throughout this chapter the main computations were implemented using the `fda` package for R [39, 40].

2.1 Estimation of Functions

The basic idea of functional data analysis is that the observed data are generated from underlying continuous functions [38]. The recorded data for each observed entity consists of discrete data points x_1, \dots, x_n taken at time or location points t_1, \dots, t_n . This temporal ordering is indicated by simply writing $x(t_j)$.

In contrast to classical parametric statistics where the data is assumed to be independent samples from a certain distribution, FDA assumes that the data points arise from a smooth function \tilde{x} . This means that $x(t_j)$ is considered to be an observation of $\tilde{x}(t_j)$.

Replications of this underlying function may be available. By replications we mean multiple samples of a single function. The i th replication of the underlying function can be written as $\tilde{x}_i(t)$.

For the annotations data and for observational data in general it can be assumed that the sequence of observed values are commonly affected by noise [38, 31],

$$x(t) = \tilde{x}(t) + \varepsilon(t) \tag{2.1}$$

$$\mathbf{x} = \tilde{\mathbf{x}} + \boldsymbol{\varepsilon}, \tag{2.2}$$

which means that the underlying smooth function \tilde{x} might be obscured by some error ε .

Note that (2.1) refers to one single arbitrary data point and (2.2) uses vector notation to denote all n observations on the function x , organised in a row vectors, for instance $\mathbf{x} =$

$$(x(t_1), \dots, x(t_n))^T \in \mathbb{R}^{n \times 1}.$$

The main goal is now to estimate $\tilde{x}(t)$ from the data $x(t)$ and two popular approaches exist: Through basis approximation and by convolution with kernel functions [37]. The latter approach was taken in previous work [30] by using a Savitzky-Golay filter [41] but only in order to reduce the observational noise. The approach taken here aims not only to reduce the noise but to estimate the complete function $\tilde{x}(t)$ so that it can be evaluated for any arbitrary argument t in the observed domain. This is why the basis approximation methods are preferred here.

2.1.1 Preliminaries for a functional representation

In order to prepare the raw annotation data to be represented by functions, two steps will be necessary: A proper notation similar to equations (2.1) and (2.2) will be introduced to avoid confusion and to keep the link between methodological theory and annotation data as close as possible. As a first consequence the data is harmonised to simplify the subsequent calculations.

Formalisation of the Data

As seen in the descriptive analysis the annotation data can be structured according to the subjects-video combinations. The raw annotation data for each video $i = 1, \dots, 8$ consists of $j = 1, \dots, 30$ continuous annotations $\mathbf{x}_{i,1}, \dots, \mathbf{x}_{i,30}$. Each annotation for video i and subject j was sampled at discrete time points $t_1, t_2, \dots, t_{n_{ij}}$. The data for one single data point in an annotation can be written similar to (2.1),

$$\mathbf{x}_{ij}(t_k) = \begin{bmatrix} x_1(t_k) & x_2(t_k) \end{bmatrix} \in \mathbb{R}^{1 \times 2} \quad (2.3)$$

$$(i = 1, \dots, 8, \quad j = 1, \dots, 30, \quad t_k \in \{t_1, \dots, t_{n_{ij}}\})$$

More compact a subject's annotation of one video can be written in matrix notation similarly to (2.2),

$$\mathbf{x}_{ij} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 \end{bmatrix} \in \mathbb{R}^{n_{ij} \times 2}, \quad (\mathbf{t}_{ij} \neq \mathbf{t}_{ij'}) \quad (2.4)$$

Thus each $\mathbf{x}_{ij}(t_k)$ is entirely determined:

- i and j are given by video and subject.
- t_k represents the single time stamp that was recorded with the annotation data. \mathbf{t}_{ij} contains all time stamps.
- $x_1(t_k)$ and $x_2(t_k)$ correspond to valence and arousal at time t_k . \mathbf{x}_1 denotes the whole annotation on valence axis as shown in Figure 1.3.

The relation of the annotation function that we seek to estimate and the raw annotation data can thus be expressed as,

$$x_{ij}(t) = \tilde{x}_{ij}(t) + \varepsilon(t). \quad (2.5)$$

Resampling

As a first step the timings of the annotations have to be made equal for each video. In the formalization above we referred to this as $\mathbf{t}_{ij} \neq \mathbf{t}_{ij'}$ because the timings can differ between subjects due to technical reasons. We will go into this more deeply since it will exacerbate the consequent analysis heavily.

The problem becomes clear with an appropriate illustration. Figure 2.1 shows the differences in the sampling times between subjects for the same video. Note that the data is taken from the last 20 elements of the timing vector for each video. These 20 data points span a temporal interval of one second. Each dot indicates at which time a data point was recorded. Although the sensor is supposed to have a constant sampling rate of 20hz (meaning 20 data points per second or every 50 ms) [26] the sampling points between the subjects vary strongly. A second issue lies in the unequal lengths of the different timing vectors. This also becomes visible here by means of the different temporal positions of the end points. A possible reason for this lies

in the method that was used to label the random annotation sequences [30]. The non-equal samplings deteriorate this issue by adding small temporal distortions.

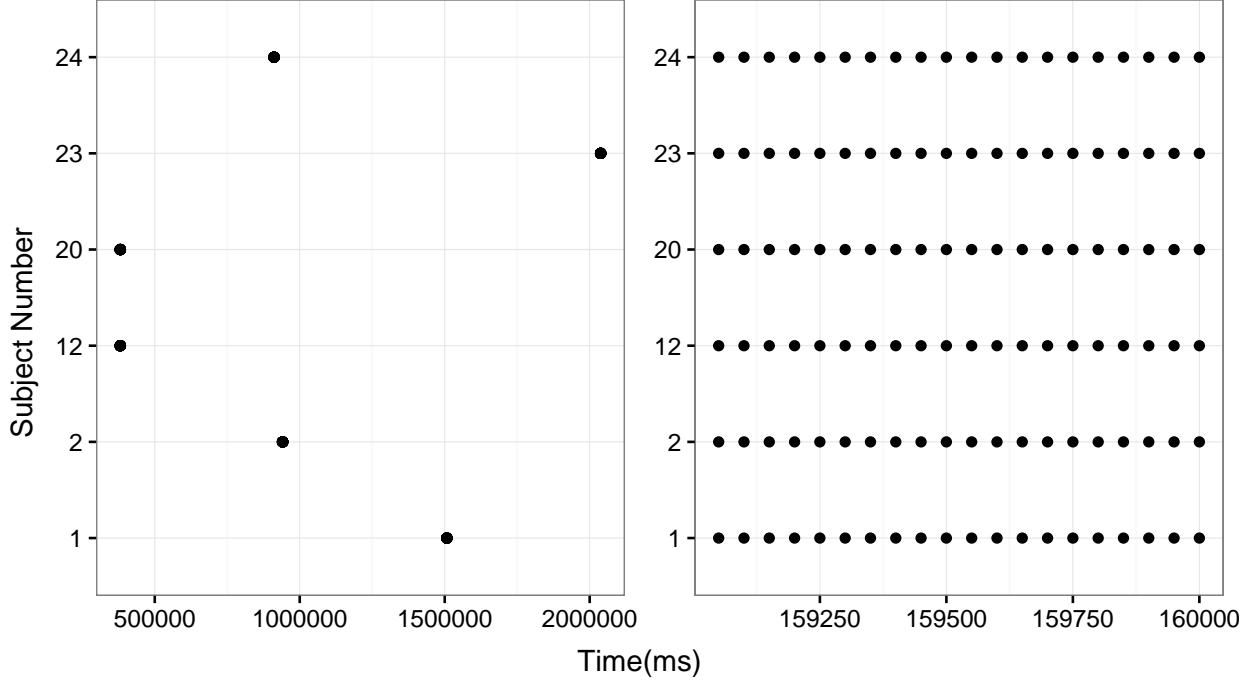


Figure 2.1: Raw sampling times (left) and the effect of resampling (right)

However, when identified, this problem can be overcome easily by interpolation and resampling the data using a shared time grid for each video. Again an illustration makes the matter clear instantly.

The final result gives us the individual annotations on a joint time grid for each video. Formally we now ensured that $\mathbf{t}_{ij} = \mathbf{t}_{ij'}$. Details on the interpolation and the resampling strategy are given in the appendix.

2.1.2 Basis Representation

The smooth function that we aim to find is now estimated through basis representation. The intuition of a basis function system is very similar to the basis system known from linear algebra. Accordingly it can be described as a set of known functions $\{\phi_1(t), \dots, \phi_K(t)\}$ that are mathematically independent of each other and have the property that any function can be

approximated arbitrarily well by taking a weighted sum or equivalently a linear combination of a sufficiently large number of these functions. The elegance of this approach is that once the basis functions have been determined, the basis approximation is linear in these variables and the fitting becomes simple [42]. This means that basis function procedures represent a function $x(t)$ by linear expansion

$$x(t) \approx \tilde{x}(t) = \sum_{k=1}^K c_k \phi_k(t). \quad (2.6)$$

By defining $\mathbf{c} \in \mathbb{R}^{K \times 1}$ as the vector of the coefficients c_k and $\boldsymbol{\phi}$ as the functional vector whose elements are the basis functions ϕ_1, \dots, ϕ_K , (2.6) can be expressed in matrix notation as,

$$\tilde{x} \approx \mathbf{c}^T \boldsymbol{\phi} = \boldsymbol{\phi}^T \mathbf{c} \quad (2.7)$$

An exact representation or interpolation is achieved when the number of basis functions exceeds the number of observations of the function, $n \leq K$. Therefore the degree to which the data are smoothed as opposed to interpolated depends highly on the number K of basis functions. This means that a basis representation is not only defined by the underlying basis system but also by K .

Another advantage of basis representation is that derivatives of arbitrary order can be expressed in simple expressions. If D_m denotes the operation of taking the m -th derivative, then the derivatives are only based on the derivatives of the basis elements but not on the weights,

$$D_m \tilde{x}(t) = \sum_{k=1}^K c_k D_m \phi_k(t) = \mathbf{c}^T D_m \boldsymbol{\phi}. \quad (2.8)$$

This equation also underlines the importance of the choice of basis, because bases that work well for function representation may give poor derivative estimates. This is because an accurate representation of the observations may force \tilde{x} to have small but high-frequency oscillations that have bad consequences for its derivatives [37, 42, 35].

2.1.3 B-Splines

In theory, any basis function system that satisfies equations (2.6) and (2.7) in terms of linearity and independence would be suitable. But ideally, the bases should have features that match those known to belong to the functions being estimated. This makes it easier to achieve a satisfactory approximation.

This is why in practice most basis representations involve either a Fourier basis for periodic data or a B-Spline basis for non-periodic data. As seen in Figure 1.3 the annotation data appears to be highly complex and non-periodic which is why a B-Spline basis representation is preferred here.

Since the goal is to derive a functional representation using P-Splines the following section is structured similar to [43, 42, 35]: First spline functions are introduced and it is shown how they can form a spline basis. After describing important characteristics of B-Splines the estimation of a B-Spline representation is derived using simple least-squares. Finally the problem of overfitting that comes with increasing basis size is discussed.

Spline Functions

Before deriving a rigorous mathematical definition of spline functions, their construction becomes intuitively clearer by giving an informal motivation. The aim is to approximate the annotation data in (2.3) and (2.4) as flexible as possible to be able to approximate any kind of annotation behaviour. In order to account for local changes in the function (e.g. jumps, wiggles or changes in shape) the approximation is defined locally or piecewise. The resulting approximation then consists of several basis functions, each of them representing a partition in the domain of the data.

The main challenge is to fuse the piecewise functions in a way that the resulting function is continuous at the points where two functions border each other.

More technically, a B-Spline basis function consists of $(l + 1)$ polynomial pieces of degree l , which are joined in an $(l - 1)$ -times continuously differentiable way. The positions where two functions border each other are called knots. All B-Spline basis functions are set up based

on a given and arbitrary knot configuration, but in practice it is very common to place knots equidistantly, which simplifies the computations [44, 45].

Figure 2.2 is referring to [43] and illustrates this matter. For example the top right panel shows a linear basis function (degree $l = 1$) consisting of $1 + 1 = 2$ linear elements.

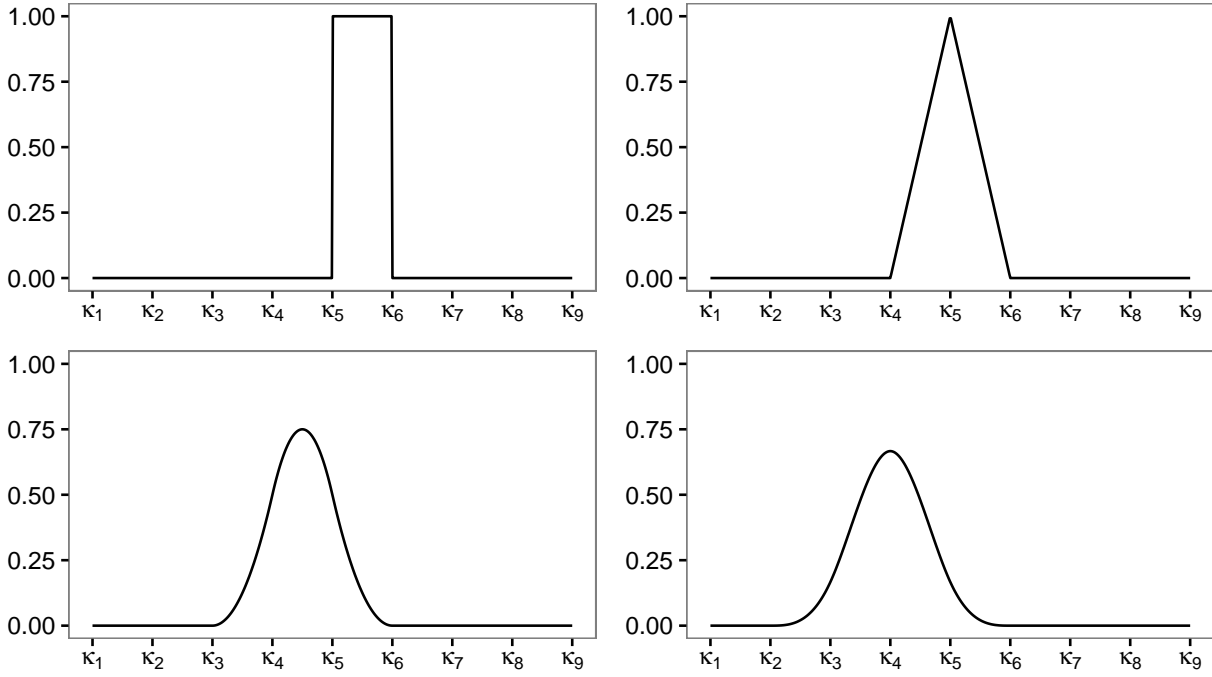


Figure 2.2: Single B-Spline basis functions for degrees $l = 0, 1, 2, 3$ with equidistant knots

The top left panel of Figure 2.2 also makes the following definition of B-Splines clear: For polynomial degree of $l = 0$ the B-Spline basis is defined by constant functions between two knots κ_j and κ_{j+1}

$$B_j^0(t) = I(\kappa_j \leq t < \kappa_{j+1}) = \begin{cases} 1, & \kappa_j \leq t < \kappa_{j+1} \\ 0, & \text{otherwise} \end{cases}, j = 1, \dots, d-1 \quad (2.9)$$

where $I(\cdot)$ denotes the indicator function. Higher order B-Splines are based on combinations of piecewise polynomials of degree l as shown in the other panels,

$$B_j^1(t) = \frac{t - \kappa_{j-1}}{\kappa_j - \kappa_{j-1}} I(\kappa_{j-1} \leq t < \kappa_j) + \frac{\kappa_{j+1} - t}{\kappa_{j+1} - \kappa_j} I(\kappa_j \leq t < \kappa_{j+1}). \quad (2.10)$$

In the top right panel, each basis function is defined by two linear segments on $[\kappa_{j-1}; \kappa_j)$ and $[\kappa_j; \kappa_{j+1})$ which are continuously combined at knot κ_j .

For an arbitrary order l a recursive definition of B-Splines is recursively defined by

$$B_j^l(t) = \frac{t - \kappa_{j-l}}{\kappa_j - \kappa_{j-l}} B_{j-1}^{l-1}(t) + \frac{\kappa_{j+1} - t}{\kappa_{j+1} - \kappa_{j+1-l}} B_j^{l-1}(t). \quad (2.11)$$

To use this recursive definition of B-Splines for the calculation of basis functions we need $2l$ outer knots outside of the domain $[a; b]$ in addition to the m interior knots $\kappa_1, \dots, \kappa_m$. This leads to an expanded knots sequence $\kappa_{1-l}, \kappa_{1-l+1}, \dots, \kappa_{m+l-1}, \kappa_{m+l}$.

It can be shown that the functions defined through (2.11) form a basis for function approximation [46]. That is, every continuous function in the function space can be represented as a linear combination of basis functions, just as every vector in a vector space can be represented as a linear combination of basis vectors. This can be denoted in terms of the general basis representation we introduced at the beginning of this chapter in equation (2.6),

$$\tilde{x}(t) = \sum_{k=1}^K c_k \phi_k(t) = \sum_{j=1}^d \gamma_j B_j(t). \quad (2.12)$$

Thus, $\tilde{x}(t)$ can be represented through a linear combination of $d = m + l - 1$ basis functions. Such a basis is shown in Figure 2.3 where function values on the domain $[\kappa_1, \kappa_9]$ would be approximated by 10 basis functions. The single basis functions are separated using different line types.

Characteristics of B-Spline basis functions

Now the B-Spline basis functions have mathematical properties that will be useful for the estimation of P-Splines later [43],

1. B-splines form a local basis: Each basis function is positive only in an interval formed by $l + 2$ adjacent knots. When using equidistant knots, all basis functions have the same form and are only shifted along the t -axis. At any point, $l + 1$ basis functions are positive. This can be seen in Figure 2.2

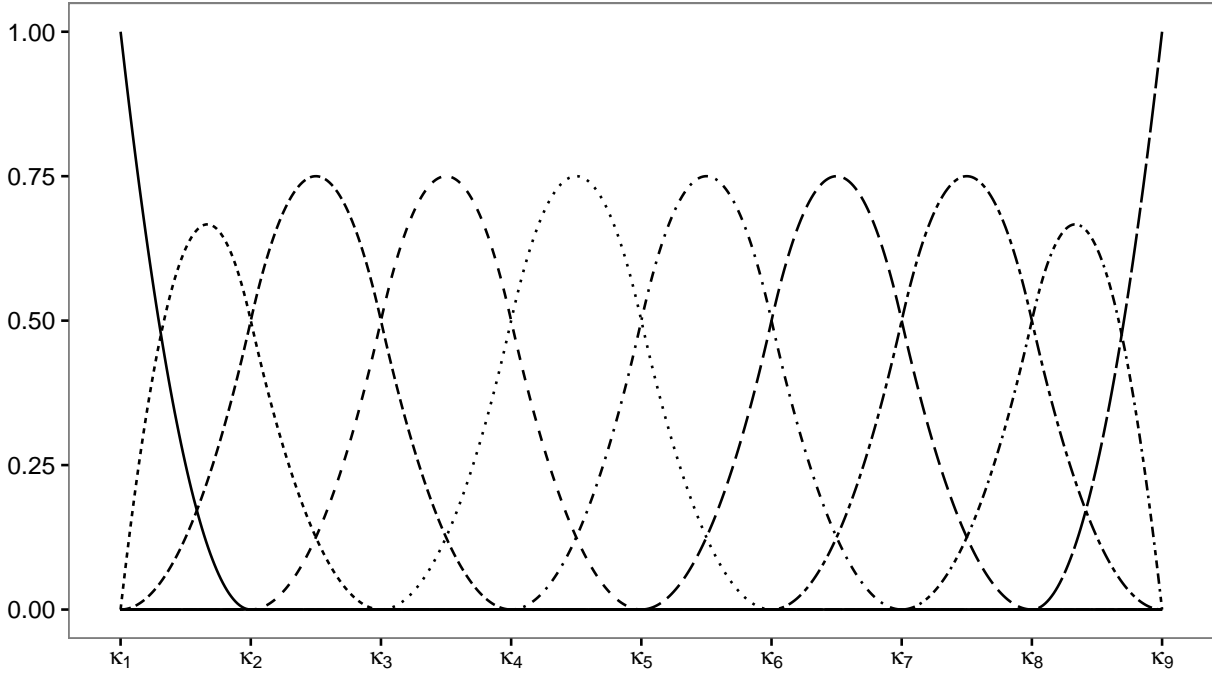


Figure 2.3: Complete B-Spline Basis for degree $l = 2$ formed by $d = 10$ basis functions.

2. Unity decomposition: For every point in the domain, $t \in \mathcal{T}$, we have

$$\sum_{j=1}^d B_j(t) = 1. \quad (2.13)$$

3. Overlapping with $2l$ adjacent basis functions: Every basis function (within \mathcal{T}) overlaps with exactly $2l$ adjacent basis functions. Visualised in Figure 2.3.
4. Bounded basis functions: The domain of the individual basis functions is bounded upwards.
5. Derivatives: As we have seen in equation (2.8) for the relation of basis representation and taking derivatives it is easy to verify that this relation holds. Because for derivatives of single B-Spline basis functions, since

$$\frac{d}{dt} B_j^l(t) = l \cdot \left(\frac{1}{\kappa_j - \kappa_{j-l}} B_{j-1}^{l-1}(t) + \frac{1}{\kappa_{j+1} - \kappa_{j+1-l}} B_j^{l-1}(t) \right). \quad (2.14)$$

Now taking the derivative for the entire spline simplifies to,

$$\frac{d}{dt} \sum_j \gamma_j B_j^l(t) = l \cdot \sum_j \frac{\gamma_j - \gamma_{j-1}}{\kappa_j - \kappa_{j-l}} B_j^{l-1}(t). \quad (2.15)$$

which is exactly (2.8). As a consequence, we are able to express the derivative of the entire spline in terms of differences of adjacent basis coefficients and basis functions of one lower degree. Thus, by estimating the coefficients γ_j , we do not only obtain an estimate for the function itself but also for its derivative. This property will be used in the context of P-Splines in the next section.

Least Squares Estimation

To approximate the annotation data using B-Spline functions their linear structure can be exploited: One can set-up the complete basis by putting the d different single basis function in the column of a matrix and evaluate the basis functions for each of the given timings rowwise. This yields the $n \times d$ design matrix \mathbf{Z} ,

$$\mathbf{Z} = \begin{bmatrix} B_1^l(t_1) & \dots & B_d^l(t_1) \\ \vdots & & \vdots \\ B_1^l(t_n) & \dots & B_d^l(t_n) \end{bmatrix}. \quad (2.16)$$

The need for a set of expanded knots becomes more clear from this matrix: Applying the recursive formula for the basis functions in \mathbf{Z} demands l additional evaluations of the lower order basis functions. This is why an initial set of m knots is not sufficient for $l > 0$.

The elegance of this approach is that \mathbf{Z} can be regarded as a design matrix \mathbf{Z} of linear model. This means that the estimation of a B-Spline representation can be traced back to a least-squares estimation of a linear model with where the covariates are given by the basis functions,

$$\begin{aligned} \mathbf{x} &= \tilde{\mathbf{x}} + \boldsymbol{\varepsilon} \\ &= \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\varepsilon}. \end{aligned} \quad (2.17)$$

To fit given data using the B-Spline basis system, least squares estimation is straightforward since estimation of linear models is well-known [43, 35],

$$\begin{aligned}
 \hat{\gamma} &= \arg \min_{\gamma} \sum_{j=1}^n \left(x(t_j) - \sum_{k=1}^d \gamma_k B_k(t_j) \right)^2 \\
 &= \arg \min_{\gamma} (\mathbf{x} - \mathbf{Z}\gamma)^T (\mathbf{x} - \mathbf{Z}\gamma) \\
 &= \arg \min_{\gamma} \|\mathbf{x} - \mathbf{Z}\gamma\|^2.
 \end{aligned} \tag{2.18}$$

Taking the derivative of the least squares criterion above yields the equation

$$2\mathbf{Z}^T \mathbf{Z} \gamma - 2\mathbf{Z}^T \mathbf{x} = \mathbf{0}. \tag{2.19}$$

Solving for γ provides the estimate $\hat{\gamma}$ that minimizes the least squares solution

$$\hat{\gamma} = (\mathbf{Z}^T \mathbf{Z})^{-1} \mathbf{Z}^T \mathbf{x}. \tag{2.20}$$

Following [43, 45] Figure 2.4. provides an intuition for the connection of the B-Spline representation as a linear model (2.17) with the results of the least-squares fit (2.20). The top left panel shows the observed discrete data points \mathbf{x} (red) and the complete B-Spline basis \mathbf{Z} (black). Each B-Spline function is a column of the design matrix $\mathbf{Z}_{\cdot i}$. The top right panel shows how $\hat{\gamma} \mathbf{Z}_{\cdot i}$ looks like, that is how each of the B-Spline functions is scaled according to the least-squares criterion. The bottom panel shows the resulting fit $\mathbf{Z} \hat{\gamma}$.

Although the fit captures the most important features of the annotation data the quality of the approximation here is rather poor. The annotation seems to have a much more complex structure than we can capture with 20 basis functions. This leads to the question of how to optimally choose the number of basis functions.

Number of Basis functions and the problem of overfitting

The main problem with the basis size is: The larger, the better the fit to the data, but of course we then run into overfitting the data and risk also fitting noise or variation that we wish to

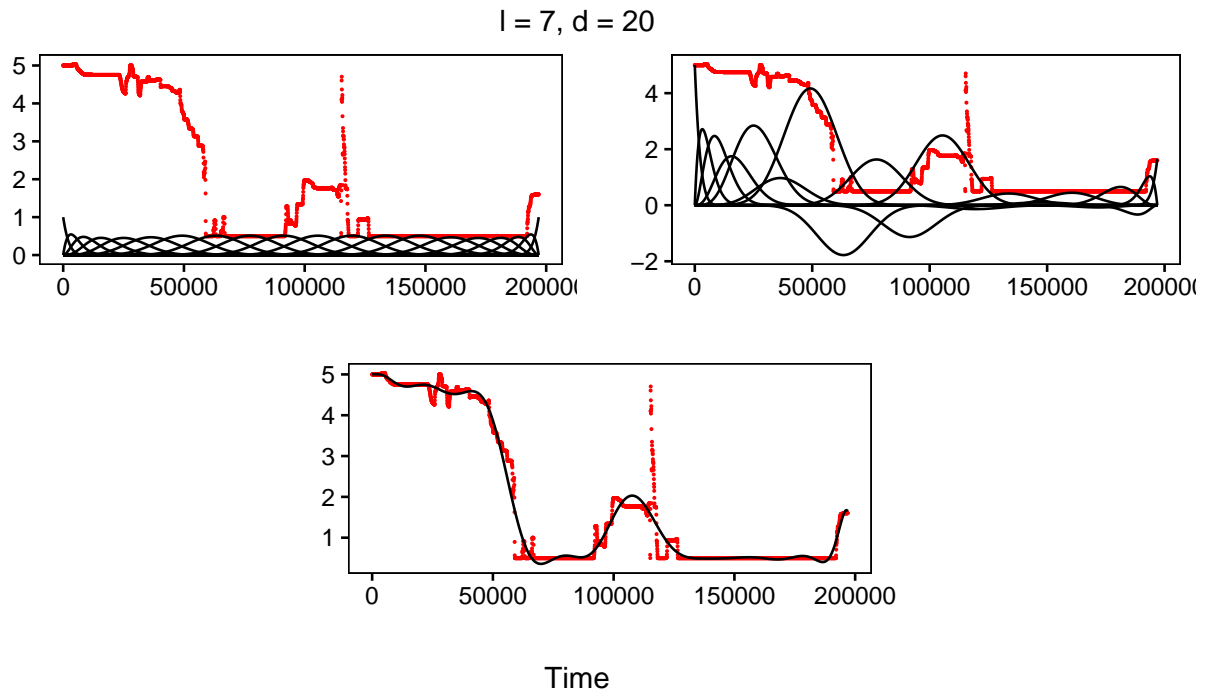


Figure 2.4: Visualisation of the components of a B-Spline representation.

ignore. On the other hand if we make d too small, we may miss some important aspects of the smooth function that we are trying to estimate [35].

Figure 2.5 shows this relation by illustrating how the basis size d affects the resulting functional representation. From the top left to the bottom right panel the number of basis functions increases by factors 3, 4 and 5. It can be seen that the functional representation using the largest basis almost perfectly represents the discrete data. The smallest basis on the other hand fails to capture the peak in the middle of the observed discrete data sequence.

We already discussed how the quality of the nonparametric function estimate \tilde{x} depends on the number of knots in Section 2.1.2. But instead of applying heuristics for the basis size it would be preferred if the dependency between goodness of fit and the number of basis functions could be lowered or optimally be determined directly from the data.

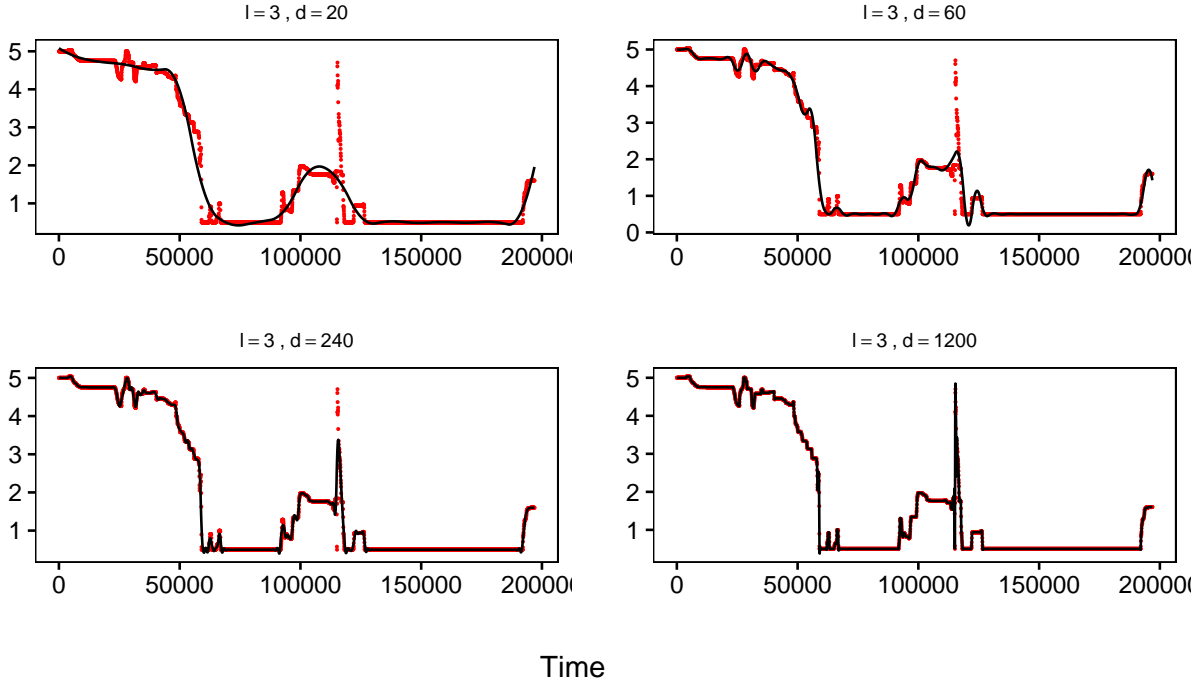


Figure 2.5: B-Spline fits for varying basis sizes.

2.1.4 Penalized Splines

To overcome this issue, a regularisation of the least-square minimisation problem (2.18) is performed by introducing roughness penalties. This is another consequence of regarding the spline representation as a linear model (2.17): Regularisation is a well-known strategy in the context of linear models, where approaches such as ridge regression aim to restrict the regression coefficients [42]. This is the approach that has been proposed two decades ago [47] and that penalized splines (P-Splines) are taking. The main idea is still quite popular today [44] and can be summarized as follows:

- Estimate the functional representation $\tilde{x}(t)$ with a polynomial spline that uses a generous number of knots d . This ensures that the underlying function can be approximated with enough flexibility to represent even highly complex functions.
- Introduce an additional penalty term that prevents overfitting and minimize a penalised least squares (PLS) criterion instead of the usual least squares criterion.

Smoothness Penalty

In order to create a penalty for B-Splines we want to characterise the smoothness of our functional representation $\tilde{x}(t)$. One measure is given by the squared derivatives since it represents the variability of the function. For instance a measure that is based on the second derivative of a function characterises the curvature of this function. The idea is now to incorporate this smoothness measure in the least square estimation by adding,

$$\lambda \int (\tilde{x}''(t))^2 dt \quad (2.21)$$

to the least square estimation,

$$\sum_{j=1}^n (x(t_j) - \tilde{x}(t_j))^2 + \lambda \int (\tilde{x}''(t))^2 dt, \quad (2.22)$$

The closed form of the derivatives for B-Splines (2.15) now allows to express the derivatives of $\tilde{x}''(t)$ as a difference of the coefficient vector

$$\text{PLS}(\lambda) = \sum_{j=1}^n \left(x(t_j) - \sum_{k=1}^d \gamma_k B_k(t_j) \right)^2 + \lambda \sum_{k=r+1}^d (\Delta^r \gamma_k)^2, \quad (2.23)$$

where Δ^r denotes r th-order differences, recursively defined as

$$\begin{aligned} \Delta^1 \gamma_k &= \gamma_k - \gamma_{k-1} \\ \Delta^2 \gamma_k &= \Delta^1 \Delta^1 \gamma_k = \Delta^1 \gamma_k - \Delta^1 \gamma_{k-1} = \gamma_k - 2\gamma_{k-1} + \gamma_{k-2} \\ &\vdots \\ \Delta^r \gamma_k &= \Delta^{r-1} \gamma_k - \Delta^{r-1} \gamma_{k-1} \end{aligned} \quad (2.24)$$

The idea behind this term is that the parameter λ weights how much the smoothness of $\tilde{x}(t)$ should influence the estimation. For $\lambda \rightarrow 0$ the smoothness of $\tilde{x}(t)$ is not taken into account and for $\lambda \rightarrow \infty$ the functional representation is maximally smooth.

Penalised Least Squares Estimation

The derivation of the PLS estimate is straightforward. Bu the notation will be more convenient if the vector of the first differences is represented by the difference matrix \mathbf{D}

$$\mathbf{D}_1 = \begin{bmatrix} -1 & 1 & & \\ & -1 & 1 & \\ & & \ddots & \ddots \\ & & & -1 & 1 \end{bmatrix} \quad (2.25)$$

$$\mathbf{D}_1 \boldsymbol{\gamma} = \begin{bmatrix} \gamma_2 - \gamma_1 \\ \vdots \\ \gamma_d - \gamma_{d-1} \end{bmatrix} \quad (2.26)$$

Higher differences can be expressed recursively,

$$\mathbf{D}_r = \mathbf{D}_1 \mathbf{D}_{r-1}. \quad (2.27)$$

For instance with $r = 2$, we obtain a $(d - 2) \times d$ difference matrix,

$$\mathbf{D}_2 = \mathbf{D}_1 \mathbf{D}_1 = \begin{bmatrix} -1 & 2 & 1 & & \\ & -1 & 2 & 1 & \\ & & \ddots & \ddots & \ddots \\ & & & -1 & 2 & 1 \end{bmatrix} \quad (2.28)$$

This yields the penalty

$$\lambda \sum_{k=r+1}^d (\Delta^r \gamma_k)^2 = \lambda \boldsymbol{\gamma}^T \mathbf{D}_r^T \mathbf{D}_r \boldsymbol{\gamma} = \lambda \boldsymbol{\gamma}^T \mathbf{K}_r \boldsymbol{\gamma}. \quad (2.29)$$

For the functional representation with B-Splines we obtain,

$$\begin{aligned}
 \int (\tilde{x}''(t))^2 dt &= \int \left(\sum_{k=1}^d \gamma_k B_k''(t) \right)^2 dt \\
 &= \int \sum_{r=1}^d \sum_{j=1}^d \gamma_r \gamma_j B_r''(t) B_j''(t) dt \\
 &= \sum_{r=1}^d \sum_{j=1}^d \gamma_r \gamma_j \int B_r''(t) B_j''(t) dt \\
 &= \boldsymbol{\gamma}^T \mathbf{K} \boldsymbol{\gamma}
 \end{aligned} \tag{2.30}$$

with $\mathbf{K}_{r,j} = \int B_r''(t) B_j''(t) dt$. The entries of the penalty matrix \mathbf{K} result from the integrated products of second derivatives of the B-Spline basis functions [46]. In fact, it allows to approximate an derivative of arbitrary order r .

This allows to reformulate the penalised least squares criterion (2.23) with the penalised differences from (2.26). This will be the minimisation problem to solve in order to get a smooth functional representation with respect to the r th-order derivative,

$$\begin{aligned}
 \text{PLS}(\lambda) &= (\mathbf{x} - \mathbf{Z}\boldsymbol{\gamma})^T (\mathbf{x} - \mathbf{Z}\boldsymbol{\gamma}) + \lambda \boldsymbol{\gamma}^T \mathbf{K} \boldsymbol{\gamma} \\
 &= \mathbf{x}^T \mathbf{x} - \mathbf{x}^T \mathbf{Z} \boldsymbol{\gamma} - \boldsymbol{\gamma}^T \mathbf{Z}^T \mathbf{x} + \boldsymbol{\gamma}^T \mathbf{Z}^T \mathbf{Z} \boldsymbol{\gamma} + \lambda \boldsymbol{\gamma}^T \mathbf{K} \boldsymbol{\gamma} \\
 &= \mathbf{x}^T \mathbf{x} - 2\boldsymbol{\gamma}^T \mathbf{Z}^T \mathbf{x} + \boldsymbol{\gamma}^T (\mathbf{Z}^T \mathbf{Z} + \lambda \mathbf{K}) \boldsymbol{\gamma}
 \end{aligned} \tag{2.31}$$

Minimization of the following

$$-2\mathbf{Z}^T \mathbf{x} + 2(\mathbf{Z}^T \mathbf{Z} + \lambda \mathbf{K}) \boldsymbol{\gamma} = \mathbf{0} \tag{2.32}$$

gives the PLS estimate

$$\hat{\boldsymbol{\gamma}} = (\mathbf{Z}^T \mathbf{Z} + \lambda \mathbf{K})^{-1} \mathbf{Z}^T \mathbf{x}. \tag{2.33}$$

In order to illustrate the influence of the smoothness penalty on the resulting functional representation we visually compare $\tilde{x}(t)$ for different values λ in Figure 2.6: The top left panel

seems to just reproduce the functional representation of Figure 2.5 for $d = 240$ since the penalisation is not influencing the functional representation at all. The other panels show the effect of further increasing lambda: The fit approaches a polynomial of degree $r - 1$ with r -th order differences which is a straight line in the case shown here.

Note also that there is a strong resemblance between the third panel with the B-Spline solution for $d = 20$ in Figure 2.5. This indicates the major advantage of using P-Splines: A B-Spline representation with a rich basis will tend to overfit the data and a slender B-Spline basis results in an undercomplex functional representation (as seen in Figure 2.5). But with P-Splines even for a much bigger basis the overfitting or underfitting of the data can now be controlled in terms of the single parameter lambda. Thus the influence of the basis size on the resulting functional representation does not cause concern anymore since d needs to be selected just large enough to represent the most important features of the data. If there are no computational constraints d can even exceed the number of data points [44, 45]. But instead we will now have to find an optimal smoothing parameter λ . We will see that there is no need to manually select it since it can be directly estimated from the data.

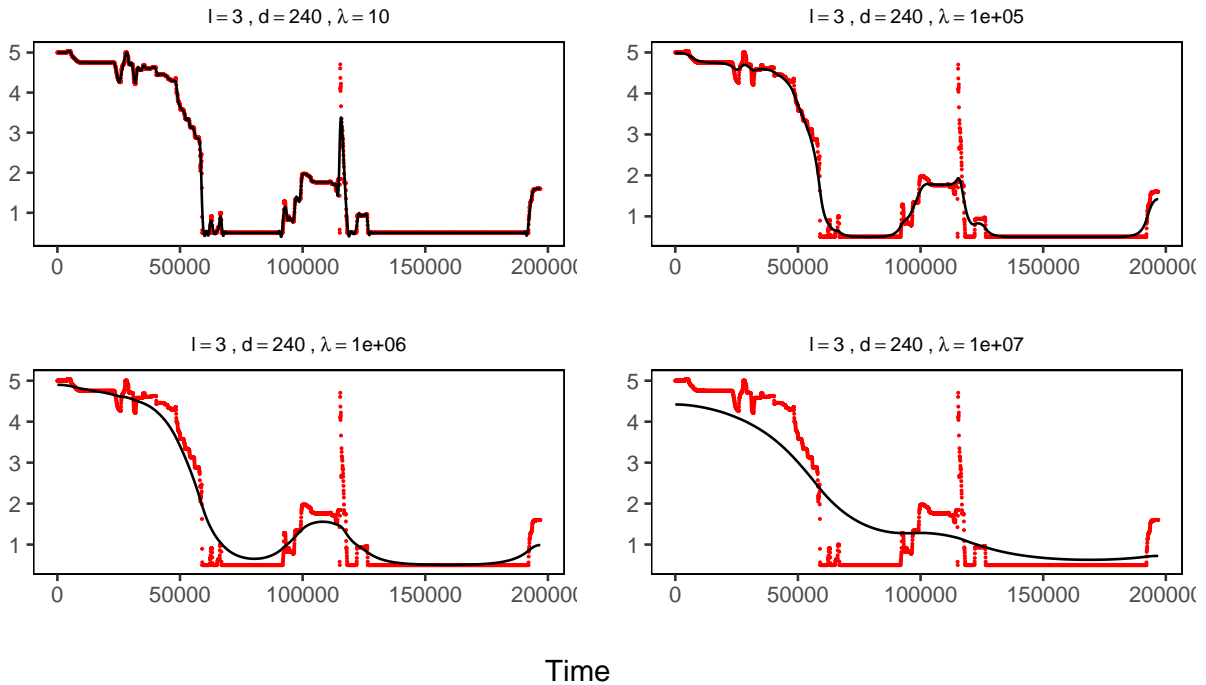


Figure 2.6: P-Spline fits for varying smoothing parameters.

Generalized Cross-Validation

To select an optimal smoothing parameter it is vital to have an estimate for the goodness of fit of the functional representation which is done by cross-validation.

The basic idea behind cross-validation is to set part of the data to one side, calling it a validation sample and fit the model to the remaining data, called training data. In that way it can be evaluated how well the model fits data that were not used to estimate model, thus avoiding to use the same data for both training the model and assess its fit [35].

For P-Splines this mechanism is taken to the extreme by leave-one-out-cross-validation. This means that the validation sample only consists of one observation and the model is fitted. This procedure is repeated for each observation in turn and the resulting error sum of squares is summed for all values.

Let $\tilde{x}_{-k}(t)$ denote the functional representation that was estimated without observation $x(t_k)$. Then the cross validation (CV) criterion can be estimated by,

$$CV(\lambda) = \frac{1}{n} \sum_{i=1}^n (x(t_i) - \tilde{x}_{-i}(t_i))^2 \quad (2.34)$$

The optimal λ can be found by comparing a number of possible λ values in terms of their CV values and choose the one that yields the minimal CV criterion.

It is obvious that the approach in (2.34) is computational highly intensive because for the calculation one needs to cycle through all n observations $x(t_k)$ and estimate n functional representations on the training data in order to compute the CV criterion. Luckily the linear model notion of the basis function approach allows a very elegant mathematical shortcut for the calculation of the CV criterion without too much computational overhead. In fact only one single fit is needed [44].

In a similar manner as with the hat matrix of a simple linear model we can easily define the smoother matrix \mathbf{S} from (2.33). Equivalent to the hat matrix, \mathbf{S} is defined as the linear

mapping from the discrete data to the functional representation,

$$\tilde{\mathbf{x}} = \mathbf{Z}(\mathbf{Z}^T \mathbf{W} \mathbf{Z} + \lambda \mathbf{K})^{-1} \mathbf{Z}^T \mathbf{W} \mathbf{x} = \mathbf{S} \mathbf{x}. \quad (2.35)$$

One can prove that $x(t_i) - \tilde{x}_{-i}(t_i) = (x(t_i) - \tilde{x}(t_i))(1 - \mathbf{S}_{ii})$ [48]. This simplifies the calculation of the CV heavily, since it allows to obtain the CV score by only performing one fit using all the data. Using the diagonal elements \mathbf{S}_{ii} of the smoother matrix CV can be calculated by,

$$\text{CV}(\lambda) = \frac{1}{n} \sum_{i=1}^n \left(\frac{x(t_i) - \tilde{x}(t_i)}{1 - \mathbf{S}_{ii}} \right)^2. \quad (2.36)$$

Nevertheless, the calculation of the smoothing matrix and its diagonal elements can still be numerically complex (especially for large data sets). For this reason one often replaces the diagonal elements with their average, yielding the generalized cross validation criterion (GCV) [43],

$$\text{GCV}(\lambda) = \frac{1}{n} \sum_{i=1}^n \left(\frac{x(t_i) - \tilde{x}(t_i)}{1 - \text{tr}(\mathbf{S})/n} \right)^2. \quad (2.37)$$

Figure 2.7 visualises the GCV criterion for an varying amount of smoothing for the annotation data as used in Figure 2.6. Until a smoothing level of $\lambda = 40$ the GCV seems to be almost constant. Afterwards the smoothing leads to a biased functional representation. For λ values between 10 and 20 the GCV seems to be minimal which is also the range of the optimal degree of smoothing λ^* . But also note that the overall improvement in terms of $\text{GCV}(\lambda^*)$ is rather small compared to the GCV level in a range from $\lambda \in [0; 40]$.

2.2 Estimating Multivariate Multi-Subject Annotation Functions using P-Splines

In this section we will discuss how to choose the parameters for the functional representation of the annotation data with P-Splines. Since the data itself has a rather complex multivariate functional structure, some emphasis will also lie on the right adaption to the data.

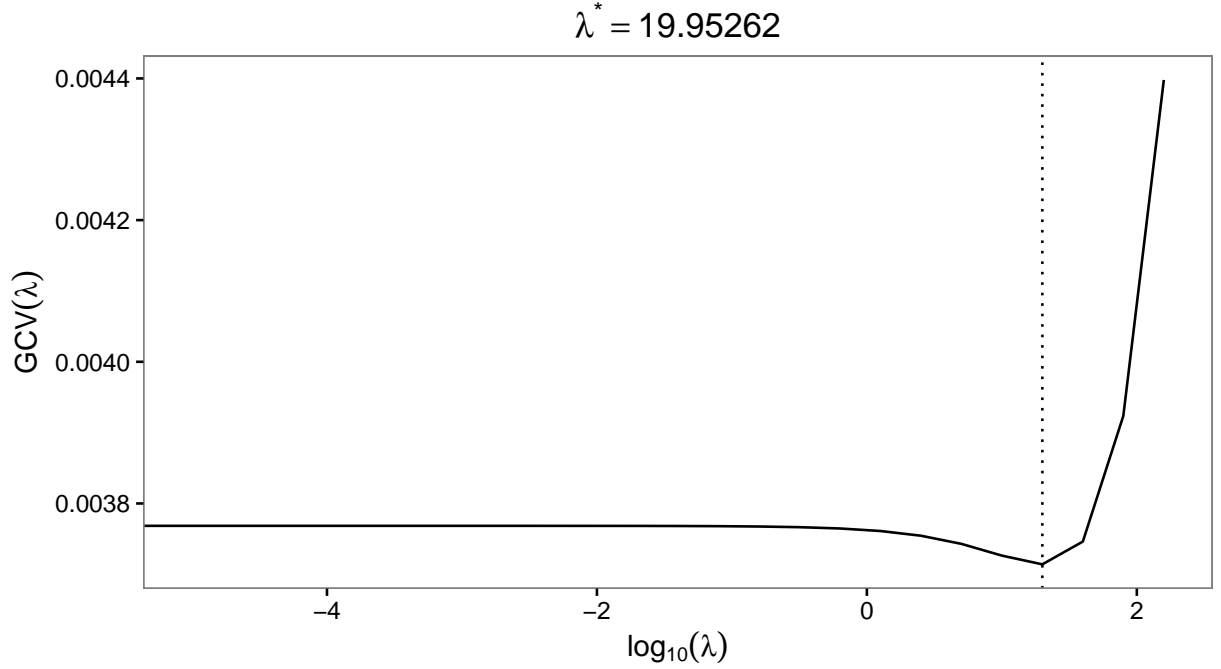


Figure 2.7: Relation between the GCV criterion and the smoothing level.

The functional representation that we aim for is bivariate since the annotations consist of valence and arousal dimensions as seen in equation (2.3). Accordingly, we can use the smoother matrix from (2.35) to express this representation in terms of the discrete input vectors of valence and arousal. Then the function representation of valence and arousal $\mathbf{x}_1, \mathbf{x}_2$ for video i and subject j can be regarded as a simple linear mapping,

$$\tilde{\mathbf{x}}_{ij} = (\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2)_{ij} = (\mathbf{S}_{\lambda, Z}^{(i,j)} \mathbf{x}_1, \mathbf{S}_{\lambda, Z}^{(i,j)} \mathbf{x}_2). \quad (2.38)$$

The notation becomes somewhat cumbersome since it is possible to fit a functional representation for each of the annotations separately. The indices denote that the linear mapping may depend on subject j and video i . Our aim will be to find parameters that are suitable for all i, j . That is, we have to set the degree of the spline functions, the number of functions and we have to estimate the smoothing parameter λ for all subjects and videos. We will discuss these choices here.

Degree of the basis function

For later research purposes the joint velocity $v(t)$ of the two axes,

$$v(t) = \sqrt{(D_1 x_1(t))^2 + (D_1 x_2(t))^2} \quad (2.39)$$

might contain additional information about the movement of the cursor. It also might be helpful in identifying salient periods in the annotation or in other words periods where the cursor was moved.

A common approach to calculate $v(t)$ is to use numerical derivatives (e.g. first central differences). But in practise the numerical derivation D_1 is prone to gross error [49]. This is why one requirement of the functional representation for the annotations should not only yield smooth annotation functions $\tilde{x}(t)$ but also smooth derivatives.

This has an influence on the choice of the basis functions. For smooth first derivatives as needed in (2.39) cubic polynomial splines are considered to be a good choice. This becomes obvious with respect to the functional measure of roughness that was defined in (2.21) as the integral of the squared second derivative. The smoothness of the first derivative in the functional representation therefore is measured in terms of the third derivative [35]. This implies that the underlying basis functions need to have continuous derivatives up to the third order and is the reason for choosing cubic splines. Note that this smooth first derivative also implies smooth $\tilde{x}(t)$.

Figure 2.8 illustrates this relationship: The top plot shows the raw data of both valence and arousal x_1 and x_2 in two different lines. On both lines there are segments of no movement of the joystick cursor, where the signal is parallel to the x axis and segments where the joystick is moved. The second plot shows the joint first central differences. Here the constant segments relate to segments in the first plot where there was no movement. These segments are interrupted by activity segments where the joystick was moved. Comparing the two plots shows that the second plot can be used to identify the segments of activity of the first plot.

Applied on the annotation data we see the effect of the roughness penalty of the first derivative quite well: The last two plots show how the penalty smooths this first derivative and how the

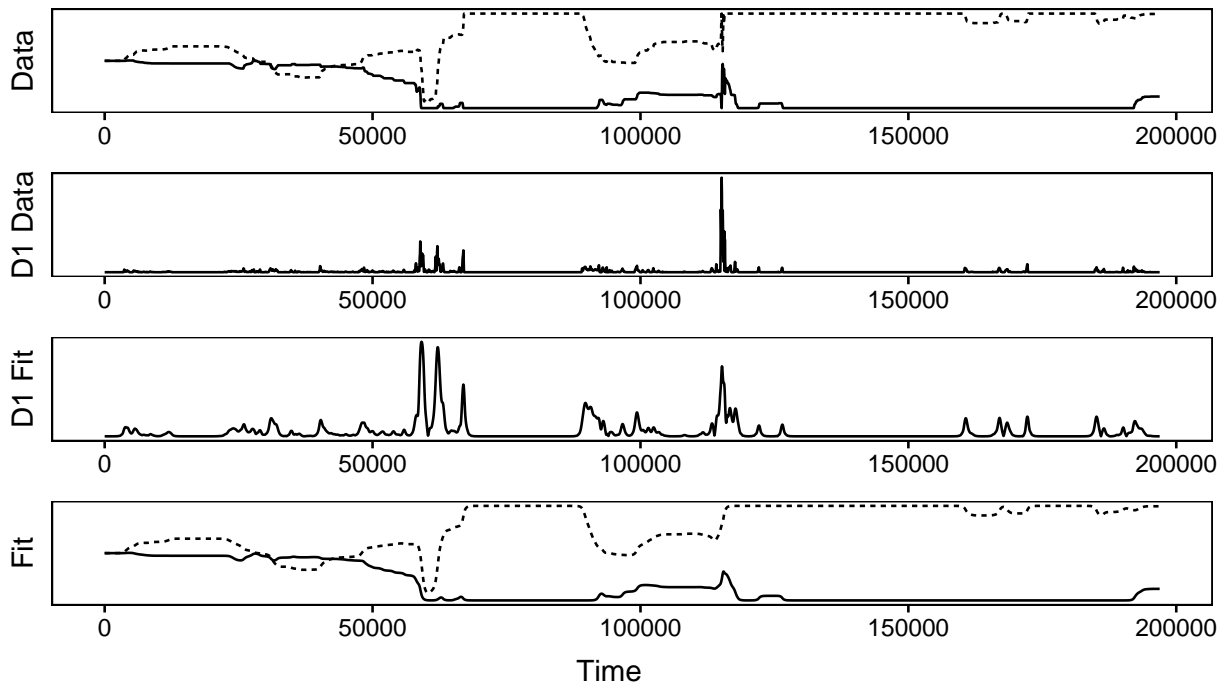


Figure 2.8: Relationship between discrete data, first central differences (two top plots) and the functional representation based on a penalization of the roughness of the first derivative (two bottom plots). Solid line for valence axis, dashed line for arousal axis.

resulting annotation function looks like. Compared to the first plot the last plot shows much less variation and is also much smoother. This shows that the choice of the cubic basis functions works as discussed in the aforementioned manner. Note that the smoothing parameter used here is not our final choice.

Number of Basis Functions

One of the obstacles is the amount of discrete data points that were recorded during the annotation process. We could use a basis function for each data points but this would result in a massive computational effort for calculating a functional representation for all annotations. This is why we reduce the number of knots but aim to keep the basis rich enough too be able to capture almost the whole variation. This trade-off is based on the annotation with the largest number of data points. This allows us to define a lower boundary for the trade-off since the shorter annotations will be represented better due to a higher ratio of basis functions per observation. As we saw in section 2.1.4 this is no problem for a P-Spline representation since

overfitting is taken care of by regularisation.

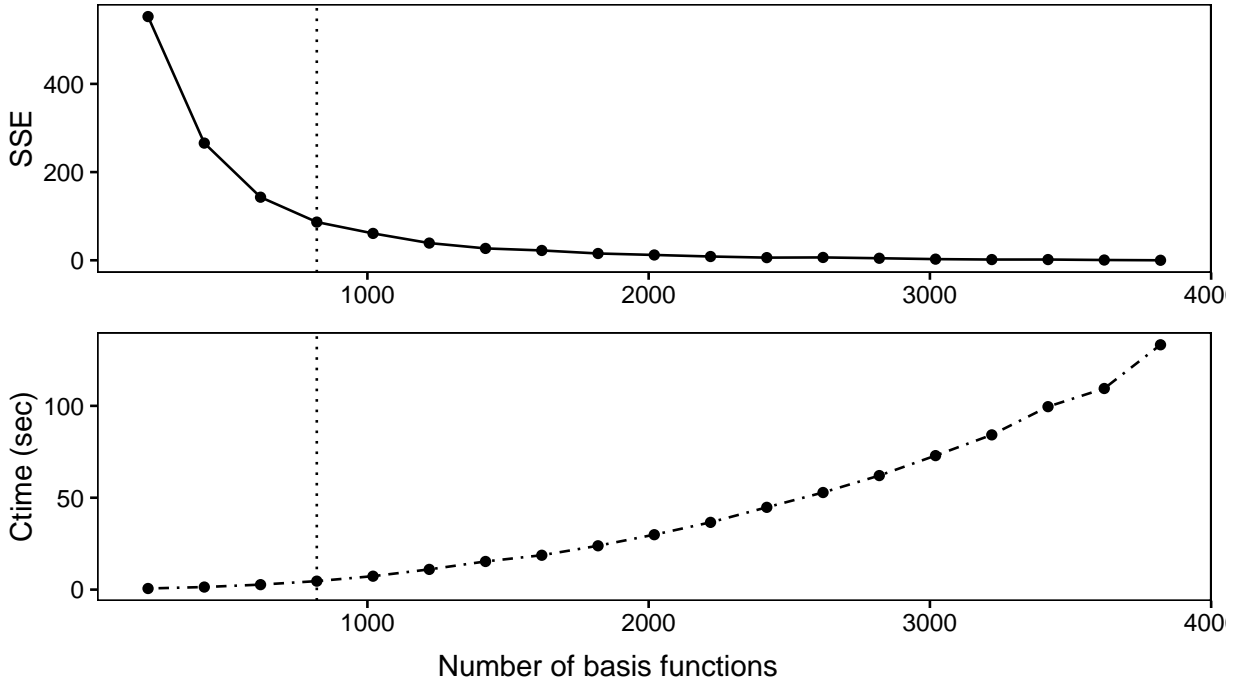


Figure 2.9: SSE and computation time for different basis sizes. The dotted vertical line marks the selected basis size for the annotation data.

Figure 2.9 shows how the goodness of fit and the computational effort depends on the number of basis functions. The quality of the fit is assessed using the mean squared error (MSE), the computational effort by using the computation time. The SSE falls off rapidly and after 1500 basis functions the improvement in error becomes small. The increase in computation time seems to be small for the small basis sizes shown in the plot. At around 1000 basis functions the computational effort seems to accelerate. Following this plot and Figure 2.5 the basis size is set to 820 basis functions. The two bottom plots of Figure 2.5 shows that the representations using 240 and 1200 basis functions are big enough to capture most of the important variation of the annotation data. From Figure 2.9 we see that with 820 basis functions the decay in SSE is captured before it stagnates. Also the computation time is reasonably high. Note that the annotation data used in this plot defines a lower boundary for the annotation data of the all the other annotation data since it contained the annotations with the most data points.

Lambda

Finally the smoothing parameter is determined from the data. This is done by means of generalized cross validation (2.37) as described before. Note that we want to find one optimal single smoothing parameter for all videos, that is λ^* that minimizes the GCV of all the different videos simultaneously. Since it is easy to calculate the GCV separately, this λ^* can be found easily by plotting the GCV values on a grid of λ values as shown in Figure 2.10.

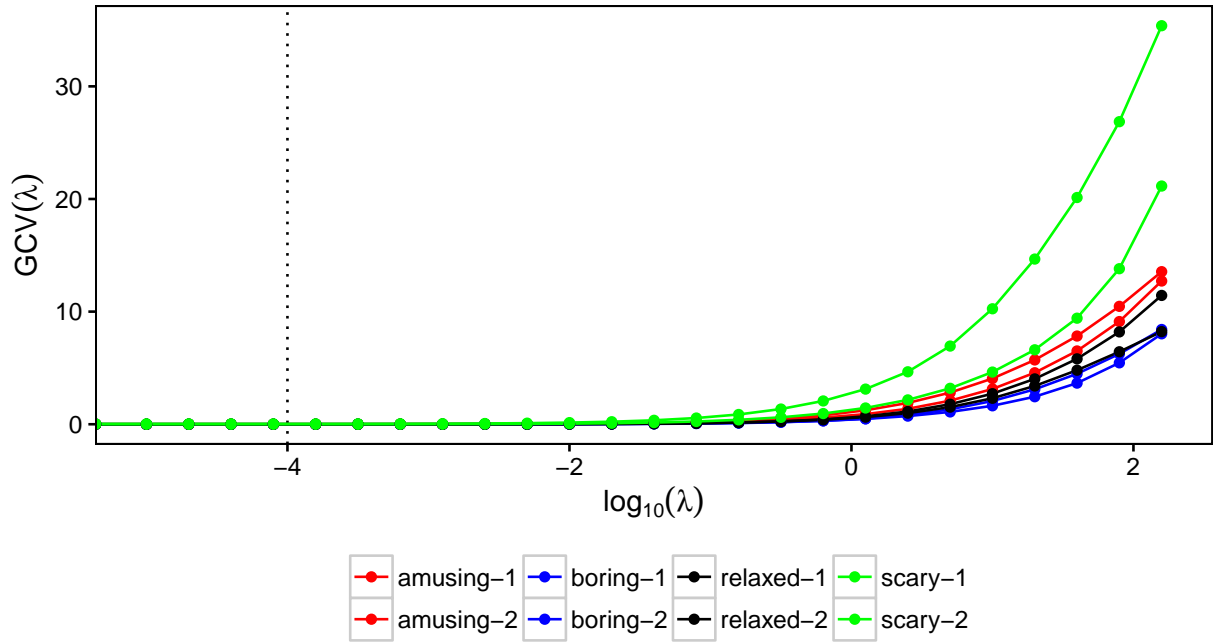


Figure 2.10: GCV criterion for different values of the smoothing parameter. Each line represents one video type. The vertical dotted line marks the optimal smoothing parameter for all videos.

Figure 2.10 shows the behaviour of the GCV criterion from (2.37) for the P-Spline representation for each video stimuli separately for all 30 subjects. With increasing λ also the GCV increases. This is in line with what was seen from Figure 2.6. The figure also indicates that the amount of smoothness present in the raw data is highest for the annotation made for the boring and relaxed video stimuli. This can be seen from the behaviour of the GCV criterion for high λ values. Compared to the functional representation of the other video stimuli the error grows slowest which indicates that smoothing does not add too much bias to the data and thus the smoothness level must be higher. In contrast, the annotation functions for the

scary video stimuli are more sensitive to smoothing since it introduces a stronger bias.

But it is striking how the GCV criterion decreases and then stagnates for decreasing λ . This behaviour applies for all the different video stimuli.

An explanation lies in the nature of the data: One would expect that the GCV criterion would increase also for small values of λ since the P-Spline representation would start to overfit the data. As the plot shows, overfitting is not a problem for the annotation data. The explanation lies in the nature of the data. The sensor data have a high signal-to-noise ration which means that the amount of smoothing needed is very low and the risk of fitting noise is also low because there is almost no noise present in the data. This indicates that the range of optimal smoothing is quite broad as long as oversmoothing is avoided. Therefore the optimal smoothing was chosen to be $\lambda^* = 10^{-4}$ which indicates a small amount of smoothing.

In cases where the different curves are more different so that the optimal smoothing varies strongly for different curves another possible approach is to use multi-objective optimisation methods to find the pareto optimal λ^* using genetic algorithms such as NSGA-II [50, 51].

Chapter 3

Statistical Analysis of Annotation Functions

Although a functional representation of the annotation data is beneficial, the problem of the high complexity of these functions remains. Therefore, the challenges formulated in section 1.3 of comparing and combining different annotation functions are still present.

In this chapter we will introduce a methodology to handle the multivariate annotation functions that were estimated in Chapter 2 in a much easier way. By using the same gist as principal component analysis (PCA), a widely used method for reducing dimensionality of multivariate observation, we are able to represent each of the multivariate annotation functions as a multivariate data point in an eigenfunction space. This makes it then possible to use the whole toolbox of multivariate statistics for comparing and combining the annotation functions.

3.1 Principal Component Analysis for Functional Data

We will first explain how the idea behind PCA can be extended to functions. Then we show how functional principal component analysis (FPCA) can be generalised to multivariate functions as in our case. Finally multivariate functional principal component analysis (MFPCA) is applied to the annotation data. At the end of this section PCA, FPCA and MFPCA are contrasted to

emphasize their main similarities and differences.

3.1.1 PCA

In order to give a brief outline of the main ideas of PCA, we follow [52, 53]:

The standard context for PCA corresponds to the upper left corner of Figure 1.5 and involves p numerical covariates observed on n entities or individuals. These data values define p n -dimensional vectors $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^p$ or equivalently a data matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$, whose j -th column is the vector \mathbf{x}_j of observations on the j -th variable.

PCA seeks for a linear combination of the columns of matrix \mathbf{X} with maximum variance. Such linear combinations are given by $\sum_{j=1}^p a_j \mathbf{x}_j = \mathbf{X}\mathbf{a}$, where \mathbf{a} is a vector of constants a_1, a_2, \dots, a_p . The variance of any such linear combination is given by $\text{Var}(\mathbf{X}\mathbf{a}) = \mathbf{a}^T \mathbf{S} \mathbf{a}$, where \mathbf{S} is the sample covariance matrix associated with the dataset. Identifying the linear combination with maximum variance is equivalent to obtaining a p -dimensional vector \mathbf{a} which maximises the quadratic form $\mathbf{a}^T \mathbf{S} \mathbf{a}$.

For this problem to have a well-defined solution, an additional condition must be imposed and the most common restriction involves working with unit-norm vectors, i.e. requiring $\mathbf{a}^T \mathbf{a} = 1$. By using a Lagrange multiplier λ this is equivalent to maximisation of $\mathbf{a}^T \mathbf{S} \mathbf{a} - \lambda(\mathbf{a}^T \mathbf{a} - 1)$. Differentiating with respect to \mathbf{a}^T and equating to the null vector leads to

$$\mathbf{S}\mathbf{a} - \lambda\mathbf{a} = \mathbf{0} \Leftrightarrow \mathbf{S}\mathbf{a} = \lambda\mathbf{a}. \quad (3.1)$$

Thus \mathbf{a} must be an eigenvector and λ the corresponding eigenvalue of the covariance matrix \mathbf{S} . In particular we are interested in the largest eigenvalue λ_1 since the eigenvalues are the variances of the linear combinations defined by the corresponding eigenvector \mathbf{a} : $\text{Var}(\mathbf{X}\mathbf{a}) = \mathbf{a}^T \mathbf{S} \mathbf{a} = \lambda \mathbf{a}^T \mathbf{a} = \lambda$.

The linear combinations $\mathbf{X}\mathbf{a}_k$ are called the principal components of the dataset. In standard

PCA terminology, the elements of the eigenvectors \mathbf{a}_k are called principal component scores as they are the values that each individual would score on a given principal component.

3.1.2 FPCA

In the context of standard PCA above we consider linear combinations of p vectors which produce new vectors. Each element of the new vectors is the result of an inner product of row i of the data matrix (x_{i1}, \dots, x_{ip}) with a p -dimensional vector of weights $\mathbf{a} = (a_1, \dots, a_p)$: $\sum_{j=1}^p a_j x_{ij}$. If rows of the data matrix become functions as depicted in the upper right corner of Figure 1.5, a functional inner product must be used instead, between a score function $a(t)$ and the i -th observation $x_i(t)$. The standard functional inner product is an integral of the form $\int a(t)x_i(t)dt$ on some appropriate compact interval. Likewise, the analogue of the $(p \times p)$ covariance matrix \mathbf{S} is a bivariate function $S(s, t)$ which, for any given two time instants s and t , returns the respective covariance defined as,

$$S(s, t) = \frac{1}{n-1} \sum_{i=1}^n (x(s_i) - \bar{x}(s_i))(x(t_i) - \bar{x}(t_i)) = \frac{1}{n-1} \sum_{i=1}^n x_i^*(s_i)x_i^*(t_i) \quad (3.2)$$

where $\bar{x}(t) = N^{-1} \sum_{i=1}^N x_i(t)$ denotes the mean function and $x_i^*(t) = x_i(t) - \bar{x}(t)$ is the i -th centred function. The functional analogue of the eigenequation (3.1) involves an integral transform which reflects the functional nature of $S(s, t)$ and of inner products,

$$\int S(s, t)a(t)dt = \lambda a(s). \quad (3.3)$$

The eigenfunctions $a(t)$ which are the analytic solution of this equation cannot in general be determined. But using the basis representation (2.6) and (2.12) a simplification of this equation is possible. The eigenfunction can be written in terms of the basis representation of the data, $a(t) = \phi(t)^T \mathbf{b}$ for some K -dimensional vector of coefficients $\mathbf{b} = (b_1, \dots, b_K)$. Assuming centred $x(t)$ and $\phi(t)$ the covariance function at time (s, t) simplifies to

$$S(s, t) = \frac{1}{n-1} \tilde{\mathbf{x}}(s)^T \tilde{\mathbf{x}}(t)^T = \phi(s)^T \mathbf{C}^T \mathbf{C} \phi(t) \quad (3.4)$$

and the eigen-equation becomes, after some simplifications

$$\frac{1}{n-1} \phi(s)^T \mathbf{C}^T \mathbf{C} \mathbf{W} \mathbf{b} = \lambda \phi(s)^T \mathbf{b} \quad (3.5)$$

where $\mathbf{W} \in \mathbb{R}^{K \times K}$ is the matrix of inner products $\int \phi_j(s) \phi_j(t) dt$ between the basis functions. Since this equation must hold for all values of s it reduces to

$$\frac{1}{n-1} \mathbf{C}^T \mathbf{C} \mathbf{W} \mathbf{b} = \lambda \mathbf{b}. \quad (3.6)$$

There is another interpretation of PCA in continuous domain which emphasises the approximation of the function $x_i(t)$ by an infinite sum of functional principal components. It is also referred to as the Karhunen-Loève (KL) expansion. [54, 55],

$$x_i(t) = \mu(t) + \sum_{k=1}^{\infty} A_{ik} \phi_k(t), \quad (3.7)$$

where $A_{ik} = \int ((x_i(t) - \mu(t)) \phi_k(t) dt)$ are the FPCs of x_i and are referred to as scores. From (3.7) it also becomes clear that the score can be used to characterise the functional observation x_i compared to other observations since it depends on i .

For the annotation data the use of FPCA is impractical since it has to be applied on each valence and arousal separately. It is not clear, how the scores those two dimensions are related then and how they can be used to jointly characterise the annotation functions. Another challenge for the annotation data is that the annotation for the different videos do not have the same time domain. The annotations for scary might be much longer than for boring. This also has to be taken into account when comparing annotation functions for different videos.

3.1.3 FPCA for Multivariate Functions

We aim for an even more general PCA approach that allows to take into account multivariate functional observations on arbitrary domains. A very recent approach is multivariate functional principal component analysis (MFPCA) [56].

We summarize MFPCA by introducing the method step by step: First the data structure and

notation is introduced. This is mainly because MFPCA is quite general and allows to be used with even more complex functional structures than we face for the annotation functions, which is why it has to be formalised more rigorously. Then the necessary assumptions and theoretical foundations are introduced. Afterwards the Karhunen-Loève theorem is stated in a multivariate functional setting and a link between a univariate and multivariate Karhunen-Loève decompositions is established.

Data structure and notation

In extension to Figure 1.5 now each observation consists of $p \geq 2$ functions $X^{(1)}, \dots, X^{(p)}$. Also each function can be defined on different domains $\mathcal{T}_1, \dots, \mathcal{T}_p$ with possibly different dimensions. For example the first dimension \mathcal{T}_1 could be time, the second dimension \mathcal{T}_2 three-dimensional space and the third \mathcal{T}_3 frequency.

Technically, \mathcal{T}_j must be compact sets in \mathbb{R}^{d_j} , $d_j \in \mathbb{N}$ with finite (Lebesgue-) measure and each element $X^{(j)} : \mathcal{T}_j \rightarrow \mathbb{R}$ is assumed to be square-integrable $X^{(j)} \in \mathcal{L}^2(\mathcal{T}_j)$. This assures that the functional observations are well-defined and well-behaved e.g. so that scalar products can be defined.

The different functions are combined in the function vector X ,

$$X(\mathbf{t}) = (X^{(1)}(t_1), \dots, X^{(p)}(t_p)) \in \mathbb{R}^p. \quad (3.8)$$

Note that $\mathbf{t} = (t_1, \dots, t_p) \in \mathcal{T} = \mathcal{T}_1 \times \dots \times \mathcal{T}_p$ is a p -tuple of d_1, \dots, d_p dimensional vectors. In the case of time, space and frequency one would obtain a 3-tuple of (1, 3, 1)-dimensional vectors.

Assumptions and theoretical foundations

It will be further assumed that the functional observations are centered,

$$\mu(\mathbf{t}) = \mathbb{E}(X^{(1)}(t_1), \dots, X^{(p)}(t_p)) = \mathbf{0}, \quad \forall \mathbf{t} \in \mathcal{T} \quad (3.9)$$

For different positions in the domain $\mathbf{s}, \mathbf{t} \in \mathcal{T}$ the matrix of covariances C has a more complex structure due to the more complex data but the same notion as before. This added complexity can be taken into account by using the tensor product \otimes to generalise the outer product, $C(\mathbf{s}, \mathbf{t}) = \mathbb{E}(X(\mathbf{s}) \otimes X(\mathbf{t}))$. The entries of C are defined by the covariances of elements of the function vector,

$$C_{ij}(s_i, t_j) = \mathbb{E}(X^{(i)}(s_i)X^{(j)}(t_j)) = \text{Cov}(X^{(i)}(s_i), X^{(j)}(t_j)), \quad s_i \in \mathcal{T}_i, t_j \in \mathcal{T}_j. \quad (3.10)$$

This refers to the calculation of the covariance of the i -th and j -th element of the function vector, e.g. the covariance between time and frequency.

As noted in the previous sections, a suitable inner product is the basis of all approaches of PCA. This is because these inner products allow intuitive geometrical notions such as lengths and angles. For multivariate functions such as $f = (f^{(1)}, \dots, f^{(p)})$ with elements $f^{(j)} \in \mathcal{L}^2(\mathcal{T}_j)$ define the space $\mathcal{H} = \mathcal{L}^2(\mathcal{T}_1) \times \dots \times \mathcal{L}^2(\mathcal{T}_p)$ and the multivariate, multi-domain functional scalar product can be defined through,

$$\langle\langle f, g \rangle\rangle = \sum_{j=1}^p \langle f^{(j)}, g^{(j)} \rangle_2 = \sum_{j=1}^p \int_{\mathcal{T}_j} f^{(j)}(t_j) g^{(j)}(t_j) dt_j, \quad f, g \in \mathcal{H} \quad (3.11)$$

[56] showed that \mathcal{H} is a Hilbert space with respect to this scalar product $\langle\langle \cdot, \cdot \rangle\rangle$.

This allows to introduce a multivariate, multi-domain covariance operator $\Gamma : \mathcal{H} \longrightarrow \mathcal{H}$ with the j -th element of Γf , $f \in \mathcal{H}$ defined as,

$$(\Gamma f)^{(j)}(t_j) = \sum_{i=1}^p \int_{\mathcal{T}_i} C_{ij}(s_i, t_j) f^{(i)}(s_i) ds_i = \langle\langle C_{\cdot j}(\cdot, t_j), f \rangle\rangle, \quad t_j \in \mathcal{T}_j \quad (3.12)$$

In the initial example the covariance structure of functional observations on time, space and frequency could now be expressed more conveniently using Γ instead of using the univariate covariances of C from (3.10). Showing the equivalency of those two covariances is then crucial for MFPCA.

The Karhunen-Loève Theorem for multivariate functional data

[56] showed that under mild conditions, Γ has the same properties as the covariance operator in the univariate case and therefore a Karhunen-Loève representation for multivariate functional data exists. This is done by showing that Γ is a linear, self-adjoint and positive operator and then by concluding that there exists a complete orthonormal basis of eigenfunctions $\psi_m \in \mathcal{H}$, $m \in \mathbb{N}$ of Γ such that $\Gamma\psi_m = \nu_m\psi_m$ and $\nu_m \rightarrow 0$ for $m \rightarrow \infty$. Using the spectral theorem, [56] state that it holds that

$$\Gamma f = \sum_{m=1}^{\infty} \nu_m \langle f, \psi_m \rangle \psi_m. \quad (3.13)$$

This allows to establish the link between the two covariances C and Γ by Mercer's Theorem which gives absolute and uniform convergence of the sum,

$$\text{Cov}(X^{(j)}(s_j), X^{(j)}(t_j)) = C_{jj}(s_j, t_j) = \sum_{m=1}^{\infty} \nu_m \psi_m^{(j)}(s_j) \psi_m^{(j)}(t_j), \quad (3.14)$$

for $j = 1, \dots, p$ and $s_j, t_j \in \mathcal{T}_j$.

[56] show that this relationship can be used for a formulation of a multivariate Karhunen-Loève representation:

Theorem 1. *Under the assumptions of [56], Proposition 2,*

$$X(\mathbf{t}) = \sum_{m=1}^{\infty} \rho_m \psi_m(\mathbf{t}), \quad \mathbf{t} \in \mathcal{T} \quad (3.15)$$

with zero mean random variables $\rho_m = \langle X, \psi_m \rangle$ and $\text{Cov}(\rho_m, \rho_n) = \nu_m \delta_{mn}$.

Also,

$$\mathbb{E}(\|X(\mathbf{t}) - \sum_{m=1}^M \rho_m \psi_m(\mathbf{t})\|^2) \longrightarrow 0 \text{ for } M \rightarrow \infty \text{ uniformly for } \mathbf{t} \in \mathcal{T}. \quad (3.16)$$

The theorem yields an infinite approximation of the functional observation vector similar to (3.7). The eigenvalues ν_m represent the amount of variability in X explained by the single multivariate functional principal components ψ_m , while the multivariate functional principal components scores ρ_m serve as weights of ψ_m in the Karhunen-Loève representation of X . As the eigenvalues ν_m decrease towards 0, leading eigenfunctions reflect the most important features of X . In practice optimal M -dimensional approximations to X are used

$$X_{[M]}(\mathbf{t}) = \sum_{m=1}^M \rho_m \psi_m(\mathbf{t}), \quad \mathbf{t} \in \mathcal{T}. \quad (3.17)$$

This yields also the approximation that is used for the annotation functions later in section 3.2.

Relationship Between Univariate and Multivariate FPCA for Finite Karhunen-Loève Decompositions

Since the covariances are equivalent, the Karhunen-Loève representation of multivariate functional data can also be directly connected to the univariate Karhunen-Loève representations (3.7) of the single elements $X^{(j)}$. The aim is to represent the multivariate representation in terms of the univariate representation and vice versa. Such a relation allows the computation of MFPCA by using existing FPCA approaches.

Theorem 2. *The multivariate functional vector $X = (X^{(1)}, \dots, X^{(p)})$ has a finite Karhunen-Loève representation if and only if all univariate elements $X^{(1)}, \dots, X^{(p)}$ have a finite Karhunen-Loève representation. In this case it holds:*

1. *Given the multivariate Karhunen-Loève representation of Theorem 1, the positive eigenvalues $\lambda_1^{(j)} \geq \dots \geq \lambda_{M_j}^{(j)} > 0, M_j \leq M$ of the univariate covariance operator $\Gamma^{(j)}$ associated with $X^{(j)}$ correspond to the positive eigenvalues of the matrix $\mathbf{A}^{(j)} \in \mathbb{R}^{M \times M}$ with entries*

$$A_{mn}^{(j)} = (\nu_m \nu_n)^{1/2} \langle \psi_m^{(j)}, \psi_n^{(j)} \rangle_2, \quad m, n = 1, \dots, M \quad (3.18)$$

The eigenfunctions of $\Gamma^{(j)}$ are given by

$$\phi_m^{(j)} = (\lambda_m^{(j)})^{-1/2} \sum_{n=1}^M \nu_n^{1/2} [\mathbf{u}_m^{(j)}]_n \psi_n^{(j)}(t_j), \quad t_j \in \mathcal{T}_j, m = 1, \dots, M_j, \quad (3.19)$$

where $\mathbf{u}_m^{(j)}$ denotes an (orthonormal) eigenvector of $\mathbf{A}^{(j)}$ associated with eigenvalue $\lambda_m^{(j)}$ and $[\mathbf{u}_m^{(j)}]_n$ denotes the n -th entry of this vector. For the univariate scores

$$\xi_m^{(j)} = \langle X^{(j)}, \phi_m^{(j)} \rangle_2 = (\lambda_m^{(j)})^{-1/2} \sum_{n=1}^M \nu_n^{1/2} [\mathbf{u}_m^{(j)}]_n \sum_{k=1}^M \rho_k \langle \psi_n^{(j)}, \psi_k^{(j)} \rangle_2 \quad (3.20)$$

2. Assuming the univariate Karhunen-Loève representation $X^{(j)} = \sum_{m=1}^{M_j} \xi_m^{(j)} \phi_m^{(j)}$ with $\Gamma^{(j)} \phi_m^{(j)} = \lambda_m^{(j)} \phi_m^{(j)}$ for each element of $X^{(j)}$ of X , the positive eigenvalues $\nu_1 \geq \dots \geq \nu_M > 0$ of Γ with $M \leq \sum_{j=1}^p M_j = M_+$ correspond to the positive eigenvalues of the matrix $\mathbf{Z} \in \mathbb{R}^{M_+ \times M_+}$ consisting of blocks $\mathbf{Z}^{(jk)} \in \mathbb{R}^{M_j \times M_k}$ with entries

$$Z_{mn}^{(jk)} = \text{Cov}(\xi_m^{(j)} \xi_n^{(k)}), \quad m = 1, \dots, M_j, n = 1, \dots, M_k, j, k = 1, \dots, p. \quad (3.21)$$

The eigenfunctions of Γ are given by their elements

$$\psi_m^{(j)}(t_j) = \sum_{n=1}^{M_j} [\mathbf{c}_m]_n^{(j)} \phi_n^{(j)}(t_j), \quad t_j \in \mathcal{T}_j, m = 1, \dots, M, \quad (3.22)$$

where $[\mathbf{c}_m]^{(j)} \in \mathbb{R}^{M_j}$ denotes the j -th block of an (orthonormal) eigenvector \mathbf{c}_m of \mathbf{Z} associated with eigenvalue ν_m . The scores are given by

$$\rho_m = \sum_{j=1}^p \sum_{n=1}^{M_j} [\mathbf{c}_m]_n^{(j)} \xi_n^{(j)}. \quad (3.23)$$

Estimation of Multivariate FPCA

Since the previous theorem establishes a link between univariate and multivariate Karhunen-Loève decompositions, the estimation of the MFPCA becomes straightforward:

The first step calculates univariate FPCAs on each of the elements of the functional vector. The resulting principal component scores are then used to form a matrix that contains the scores

of all the different univariate FPCAs. An eigenanalysis of this matrix yields eigenvalues and eigenvectors that can be used to calculate multivariate eigenfunctions and multivariate scores. The detailed steps are as follows [56],

1. For each element $X^{(j)}$ estimate a univariate FPCA based on the observations $x_1^{(j)}, \dots, x_N^{(j)}$. This results in estimated eigenfunctions $\hat{\phi}_m^{(j)}$ and scores $\hat{\xi}_{i,m}^{(j)}$, $i = 1, \dots, N, m = 1, \dots, M_j$ for suitably chosen truncation lags M_j .
2. Define the matrix $\Xi \in \mathbb{R}^{N \times M_+}$, where each row $(\hat{\xi}_{i,1}^{(1)}, \dots, \hat{\xi}_{i,M_1}^{(1)}, \dots, \hat{\xi}_{i,1}^{(p)}, \dots, \hat{\xi}_{i,M_p}^{(p)})$ contains all estimated scores for a single observation. An estimate $\hat{\mathbf{Z}} \in \mathbb{R}^{M_+ \times M_+}$ of the block matrix \mathbf{Z} in Theorem 2 is given by $\hat{\mathbf{Z}} = (N - 1)^{-1} \Xi^T \Xi$.
3. Perform a matrix eigenanalysis for $\hat{\mathbf{Z}}$ resulting in eigenvalues $\hat{\nu}_m$ and orthonormal eigenvectors $\hat{\mathbf{c}}_m$.
4. Estimates for the multivariate eigenfunctions are given by their elements

$$\psi^{(j)}_m(t_j) = \sum_{n=1}^{M_j} [\hat{\mathbf{c}}_m]_n^{(j)} \hat{\phi}_m^{(j)}(t_j), \quad t_j \in \mathcal{T}_j, m = 1, \dots, M_+ \quad (3.24)$$

and multivariate scores can be calculated via

$$\hat{\rho}_{i,m} = \sum_{j=1}^p \sum_{n=1}^{M_j} [\hat{\mathbf{c}}_m]_n^{(j)} \hat{\xi}_{i,n}^{(j)} = \Xi_{i,\cdot} \hat{\mathbf{c}}_m. \quad (3.25)$$

MFPCA for Functional Representations by Basis Expansion and Implementation

In practice or as it is the case with the annotation data, the univariate elements $X^{(j)}$ are expanded in finitely many, not necessarily orthonormal basis functions $b_m^{(j)}$ with coefficients $\theta_m^{(j)}$ which denotes exactly the functional representation using P-Splines,

$$X^{(j)}(t_j) = \sum_{m=1}^d \gamma_m B_m(t_j) = \sum_{m=1}^K \theta_m^{(j)} b_m^{(j)}(t_j), \quad t_j \in \mathcal{T}_j. \quad (3.26)$$

the right term is more general since it denotes an arbitrary basis expansion.

In this general case [56], the resulting eigenanalysis problem for MFPCA is $\mathbf{BQc} = \nu \mathbf{c}$. \mathbf{B} denotes a block diagonal matrix of scalar products $\langle b_m^{(j)}, b_n^{(j)} \rangle_2$ of univariate basis functions associated

with each element $X^{(j)}$. The symmetric block matrix \mathbf{Q} with entries $Q_{mn}^{(jk)} = \text{Cov}(\theta_m^{(j)}, \theta_n^{(k)})$ corresponds to \mathbf{Z} .

The estimation of MFPCA in the case of functional representation by basis expansion then can be generalised. Given weights $w_1, \dots, w_p > 0$ and demeaned observations x_1, \dots, x_N of X with estimated basis function coefficients $\hat{\theta}_{i,m}^{(j)}$ for each element, the eigenanalysis problem to solve is,

$$(N-1)^{-1} \mathbf{B} \mathbf{D} \mathbf{\Theta}^T \mathbf{\Theta} \mathbf{D} \mathbf{c} = \nu \mathbf{c}. \quad (3.27)$$

The matrix \mathbf{B} is defined as above as the block diagonal matrix of basis scalar product. $\mathbf{D} = \text{diag}(\mathbf{w}_1^{1/2}, \dots, \mathbf{w}_p^{1/2})$ accounts for the weights. $\mathbf{\Theta}$ with rows $(\hat{\theta}_{i,1}^{(1)}, \dots, \hat{\theta}_{i,K_1}^{(1)}, \dots, \hat{\theta}_{i,1}^{(p)}, \dots, \hat{\theta}_{i,K_p}^{(p)})$ corresponds to $\mathbf{\Xi}$ and $(N-1)^{-1} \mathbf{\Theta}^T \mathbf{\Theta}$ is an estimate for \mathbf{Q} . For the detailed calculations see [56].

The authors also provide an implementation of the method described here. Their R package MFPCA [57] will be used here for the following analysis.

Comparison of the three methods

Since the motivation for using this new approach came from Figure 1.5, is beneficial to summarise PCA, FPCA and MFPCA using the the same structure.

The methodological differences for the three methods arise from the nature of one observation.

The first two columns of the table are identical to the upper two corners of Figure 1.5.

The differences arise from the nature of one observation in each case. This results in different covariance objects for each case. Since all three methods perform an eigenanalysis of these covariance objects, the resulting eigenequations have the same ingredients.

	PCA	FPCA	MFPCA
Situation:	p discrete observations on n entities	one continuous observation on n entities	p continuous observations on n entities
Data Structure:	$\mathbf{X} = \mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{x}_i \in \mathbb{R}^p$	$x(t) = x_1(t), \dots, x_n(t), x_i(t) \in \mathcal{L}^2$	$\mathbf{x}(\mathbf{t}) = x_1(\mathbf{t}), \dots, x_n(\mathbf{t}), x_i(\mathbf{t}) \in$ $x_i(\mathbf{t}) = (x^{(1)}(t_1), \dots, x^{(p)}(t_p)) \in \mathbb{R}^p$
Covariance:	Matrix Σ with elements $\Sigma_{ij} = \text{Cov}(\mathbf{x}_i, \mathbf{x}_j)$	Function V defined as $V(s, t) = \text{Cov}(x(s), x(t))$	Operator Γ defined as $(\Gamma x)^{(j)}(t_j) = \sum_{i=1}^p \int_{\mathcal{I}_i} C_{ij}(s_i, t_j) f^{(i)}(s_i) ds_i$ $C_{ij}(s_i, t_j) = \text{Cov}(x^{(i)}(s_i), x^{(j)}(t_j))$
Eigenproblem:	$\Sigma \mathbf{a} = \lambda \mathbf{a}$ or $V\xi = \rho \xi$	$V\xi = \rho \xi$	$\Gamma \psi_m = \nu_m \psi_m$
Low-Rank			
Approximation:	$\mathbf{x}_i \approx \sum_{m=1}^M x_{im} a_m$	$x_i(t) \approx \sum_{m=1}^M f_{im} \xi_m(t)$	$x_i(\mathbf{t}) \approx (\sum_{m=1}^M \rho_m^{(1)} \psi_m^{(1)}(t_1), \dots, \sum_{m=1}^M \rho_m^{(p)} \psi_m^{(p)}(t_p))$
References:	[52, 53]	[35, 58]	[56, 59, 60]

Table 3.1: Overview of PCA, FPCA and MFPCA.

3.2 Applying multivariate FPCA on the annotation data

Using [56, 57] MFPCA is applied on the annotation data. To this end, two main issues will have to be addressed: 1. As a consequence of Theorem 1 the truncated Karhunen-Loève representation (3.17) requires the parameter M to determine the number of eigenfunctions to approximate the annotation data. Due to the similarity to existing PCA methodology this can be addressed using existing strategies to determine the number of principal components. 2. Due to the complexity of the data an intuition about the functional principal components for the annotation functions is needed. This will be addressed with in the second part of this section.

3.2.1 Number of functional principal components

The scree plot [61] is one approach to determine the number of principal components for given set of data. In the situation of MFPCA its main idea can be used to determine an optimal number of principal components. For this purpose the ratio of explained variance per principal component is compared to each other. A cutoff point is identified after which the increase in explained variance is only marginal.

The calculation of the criterium is straightforward: As described in Theorem 2 the MFPCA yields M eigenvalues ν_1, \dots, ν_M that correspond to the covariance operator Γ . In a similar fashion as [62, 63] we are interested in a measure of how much of the total variance in the annotation data is explained by each multivariate functional principal component:

$$\pi_k = \frac{\sum_{j=1}^k \nu_j}{\sum_{j=1}^{\infty} \nu_j} \quad (3.28)$$

The approach taken here is now to estimate the number \hat{k}_α of principal components that account for at least $(\alpha \cdot 100)\%$ of total variation in the data. This can be done by estimation of π_k using an M -dimensional approximation,

$$\hat{\pi}_k = \frac{\sum_{j=1}^k \nu_j}{\sum_{j=1}^M \nu_j}. \quad (3.29)$$

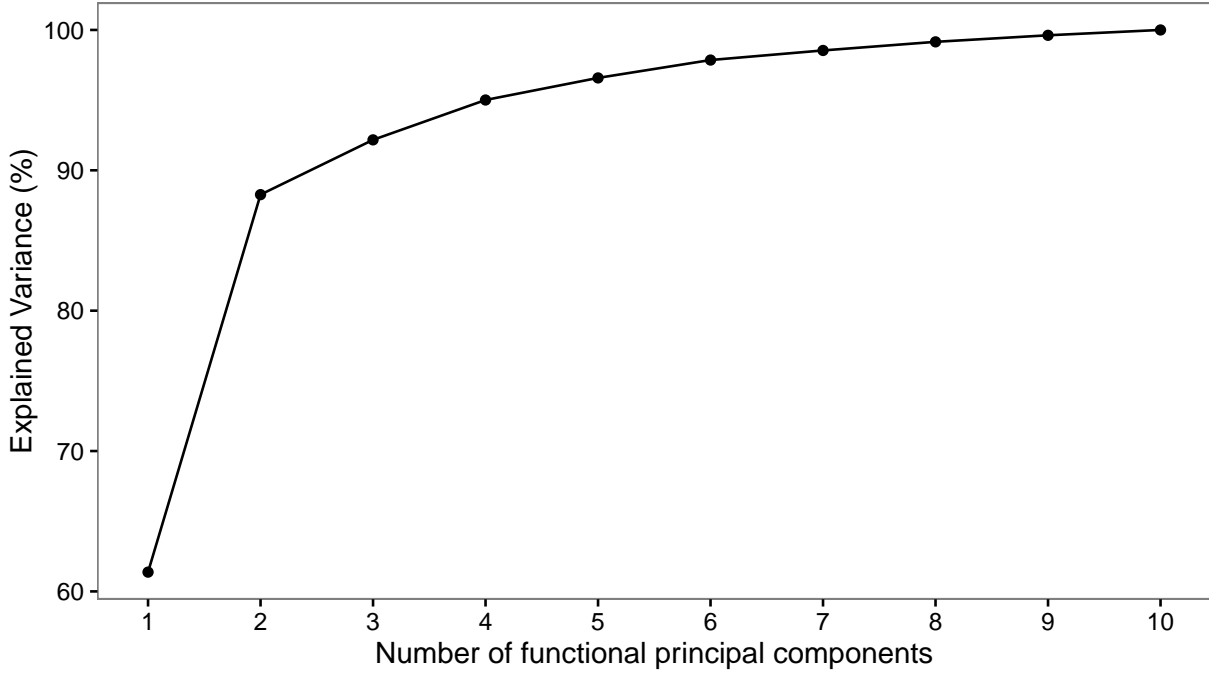


Figure 3.1: Scree plot of the MFPCA results

As a result we obtain the scree plot in Figure 3.1. The plot shows how $\hat{\pi}_k$ behaves for increasing k values. It is striking how much variance lies within the first two principal components, almost 90%. From a contentual point of view this shows how well the MFPCA was able to reproduce the two-dimensional valence-arousal space that underlies the annotation data. The plot also shows that the first component accounts for twice as much variation in the data than the second.

From the plot also follows to choose $\hat{k}_{0.95} = 4$, at least 95% of the total variance in the annotation data can be explained using only four multivariate functional principal components. This choice allows to account for a reasonable amount of variation but still to keep the dimensionality of the eigenfunction representation small enough to avoid problems regarding to the curse of dimensionality.

3.2.2 The resulting eigenfunction space

Since the MPFCA method projects each annotation function in a M dimensional eigenfunction space, each annotation can be represented by its M -dimensional score vector $\hat{\rho}$ (3.23 and

3.25) that corresponds to the M eigenfunctions. This allows a substantial simplification of this complex multivariate functional data since they can now be represented using both a vector and a function space as indicated in Figure 1.6. One major advantage that this representation allows analysing the functional data using well-known statistical methods for simple multivariate data.

In order to link this representation of rather complex functions to an intuition, Figure 3.2 gives a visual approach to the MFPCA solution similar to [42]. It shows how the first two dimensions of the eigenfunction space can be mapped to annotation functions. By showing this mapping it can be seen how the first two multivariate functional principal components work.

The left plot shows the position of all 240 annotations in the eigenfunction space. The (x,y)-coordinates are given by the respective score values. The rectangular grid superimposed on the plot is defined by the 5%, 25%, 50%, 75% and 95% quantiles of the two axis. The orange points indicate the closest annotations to the vertices of the grid, where the distance measure focuses on these projected coordinates. By choosing these annotations the whole eigenfunction space is exploited with respect to regions where the annotations are more dense.

The right plot shows how the original annotations corresponding to these orange points look like. The bottom leftmost orange point marks the annotation that is closest to the leftmost bottom vertex of the grid and thus is plotted at the bottom leftmost position of the plot matrix of the right. By doing so we obtain a visualisation of the nature of the first two principal components.

It is striking that the principal component reproduces the second quadrant in the valence-arousal plane. This means that the annotations for the scary video stimuli seem to have low first principal component scores. Higher values of first principal component scores (right side of the plot matrix) are not as easy to interpret. This is where the effect of the second principal component becomes visible: Higher scores on the second principal component lead to amusing annotations, low scores to boring.

Since the plot only was able to take the first two dimensions of the eigenfunction space into account we want to investigate if the higher dimensions can help to better distinguish the annotations for different stimuli. That is, how consistent the annotations for the different videos and different subjects have been.

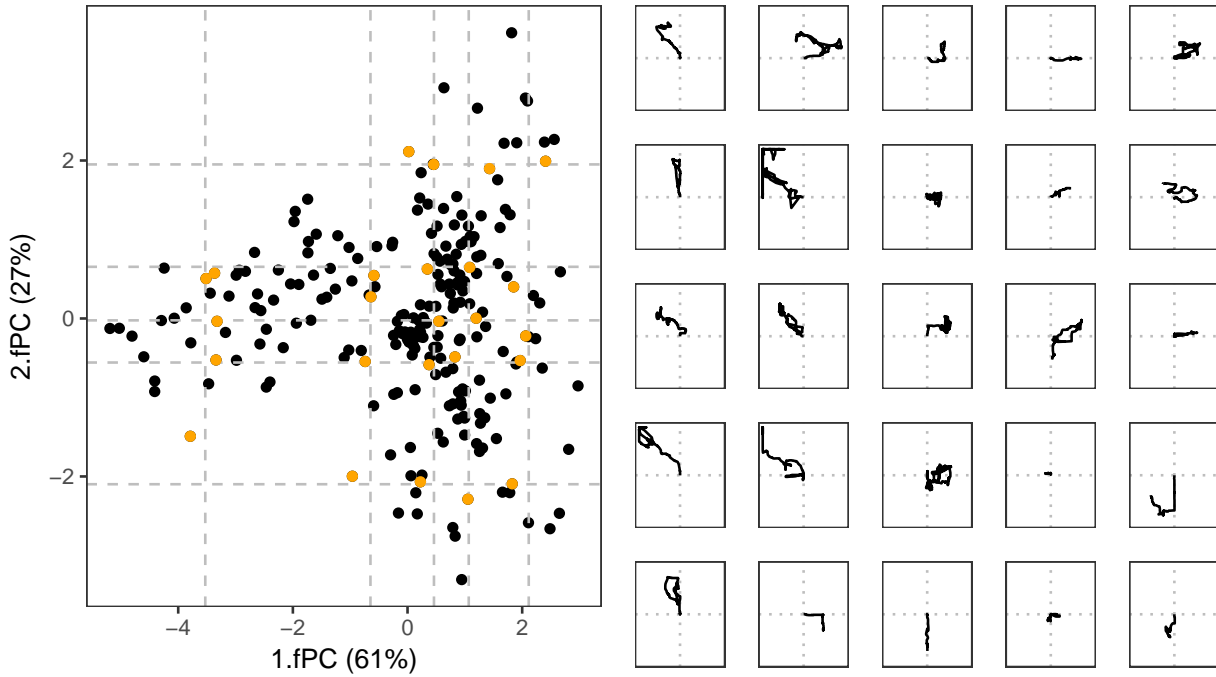


Figure 3.2: Scatterplot of the first two fPC scores (left) and illustration of the behaviour of the multivariate fPCs based on the scores of each annotation (right)

An advantage of the eigenfunction space representation of the annotation functions is that we can easily calculate the similarities of two annotations. By this we can assess how the stimuli changed the annotation behaviour of the subjects and if this behaviour was consistent over all subjects.

In the eigenfunction space, the similarity of annotations can now be easily analysed by calculating a distance matrix from the score values of each annotation function. If two annotation functions have a low distance in the eigenfunction space this means that they will look similar. This can be easily checked by the plot above where the actual annotation functions were plotted. A high distance value then means that the two functions are not alike at all.

Figure 1.3 visualises the distance matrix of the score values of each annotation function. The

pairwise distances were calculated for each possible pair of the 240 annotation functions resulting in $240^2/2$ distance values. Low distance values refer to high similarity of the annotations and are colored black. High distance values indicate a low similarity and are colored white. The grid separates each of the 8 different videos.

From the plot we learn two things: How similar were the annotations for the same stimuli? And how similar were the annotations for different stimuli? The first question refers to the diagonal of the heatmap shown in the figure. A high similarity of annotations for the same video type means that the blocks around the main diagonal should be ideally all black so that. We see that this applies for all of the videos. For the scary video stimuli it is striking that the overall similarity within these annotations is lower and shows more variation than the other video stimuli.

The second question is related to all the off-diagonal blocks. Ideally the annotation functions for the different video stimuli are all different from each other and show a low similarity which would be indicated by brighter colors. In the plot this only applies for the two scary videos. The other videos form a block with rather low distances. Amusing seems to be slightly more distinguishable than the other videos. The annotations for the relaxing videos, especially relaxing-2 are the worst. Here the distances to the other videos seem to be rather low although the stimuli were different.

Since the distances for the heatmap were calculated with an euclidean distance that takes into account all dimensions of the eigenfunction space we want to have a look at the distances separately. From the scree plot we can see that the first four dimensions of the eigenfunctions space explain approximately 95 percent of the variance in the annotation functions. The hope is that instead of using a euclidean distance in a 10-dimensional space, a 4-dimensional space is more suitable. Also because the curse of dimensionality could inflate the distances badly.

Figure 3.4 visualises how the score values can be used to distinguish the different annotations for their different stimuli. The plot has two parts: The off-diagonal plots show the scatterplots of the functional principal component scores combining the different fPCs on x- and y-axis. For

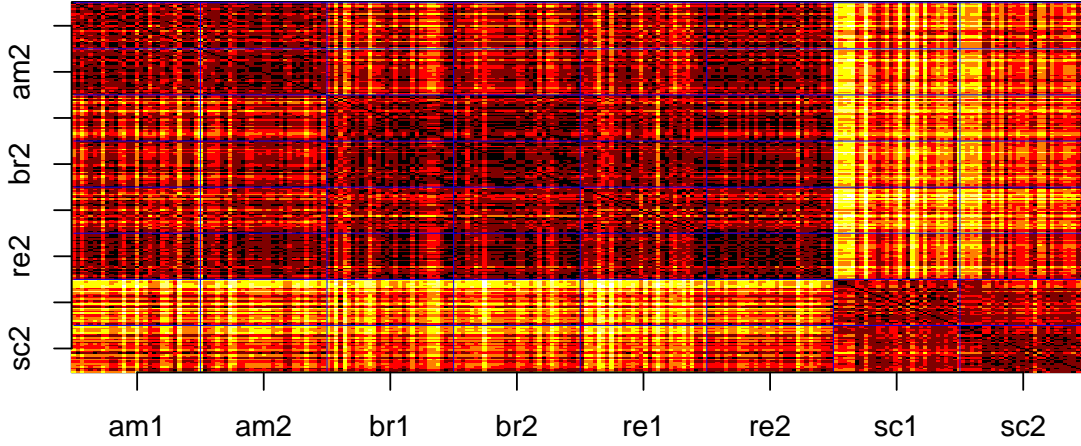


Figure 3.3: Heatmap of the similarity of the annotation functions. Black indicates high similarity, brighter colors indicate lower similarity.

instance the off-diagonal plots in the first column show scatterplots of the fPCs combining first (x) and second (y), first (x) and third (y) and first (x) and fourth (y) principal components. This is why the first plot in the first column is the same as shown in Figure 3.2. Note that the off-diagonal plots in the first row are identical to the first column but with flipped axis. The points in the scatterplots are the 240 annotations colored by their video type (amusing - black, boring - red, relaxing - blue, scary - gray). This allows to directly the similarity of the annotation functions for each dimension at one glance.

The main diagonal plots summarize how much variance is explained by the fPCs and quantify the strength of the cluster separability in terms of Fisher's Discriminant Ratio (FDR) [64] using the given fPC for a pairwise comparison. Note that the FDR serves as a one-dimensional similarity measure for the annotation functions here. Low values indicate a high similarity, a low similarity is reflected in high FDR values.

The scatterplots for the first principal component show how well it can be used to separate

the gray (scary) annotations from the rest. This is reflected by the high separability measure of this group yielding FDR values around 5. This also is in line with the observation from the distances in 10-dimensions as shown in Figure 3.3.

The second fPC has the highest separability between amusing (black) and boring (red) annotations. The relaxing annotations (blue) interfere strongly since they lie just between those two groups which results in small FDR values for amusing-relaxing and boring-relaxing. Also the scatterplots illustrate this issue (see second column/row). This is what also was seen in 10-dimensions.

The third and fourth fPC have the highest FDR for amusing-relaxing but unfortunately this yields only a poor separability power compared to the first two functional principal components.

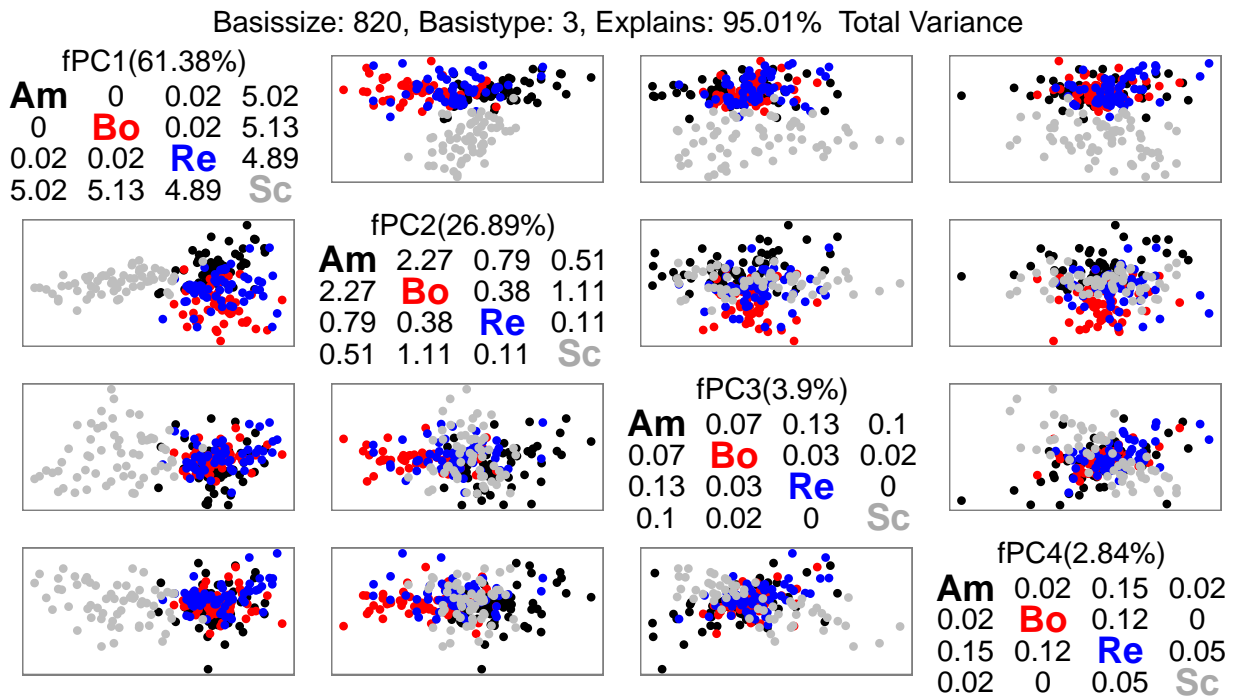


Figure 3.4: Discriminative power of the first four dimensions of the eigenfunction space. Off-diagonals: Scatterplots of the (column, row)-th principal component scores. Main diagonal: Explained variance and Fisher's Discriminant Ratio for each principal component. Coloring according to the video stimulus.

Chapter 4

Ground Truth through Characteristic Annotations

In this chapter a method to calculate a ground truth from the annotation functions is developed. Estimating the ground truth can easily be based on the MFPCA approach. In contrast to the previous chapter where the an eigenfunction space for all the annotations was found simultaneously, the focus now lies on an eigenfunction space for each video stimuli. This results in 8 different eigenfunction representations. It then will be seen how these different eigenfunction spaces provides a useful multivariate structure to tackle the task of finding ground truth that is robust against outliers.

4.1 Number of components

Calculation of MFPCA on each video separately yields the questions how many multivariate functional principal components are needed for each.

Figure 4.1 below indicates a two-component solution: The first two components explain at least almost 70% of the total variance for amusing-2 and almost 90% of total variance for boring-1. Since the videowise MFPCA aims to find the characteristic annotations for valence and arousal, a two-component solution is chosen for the characteristic annotation.

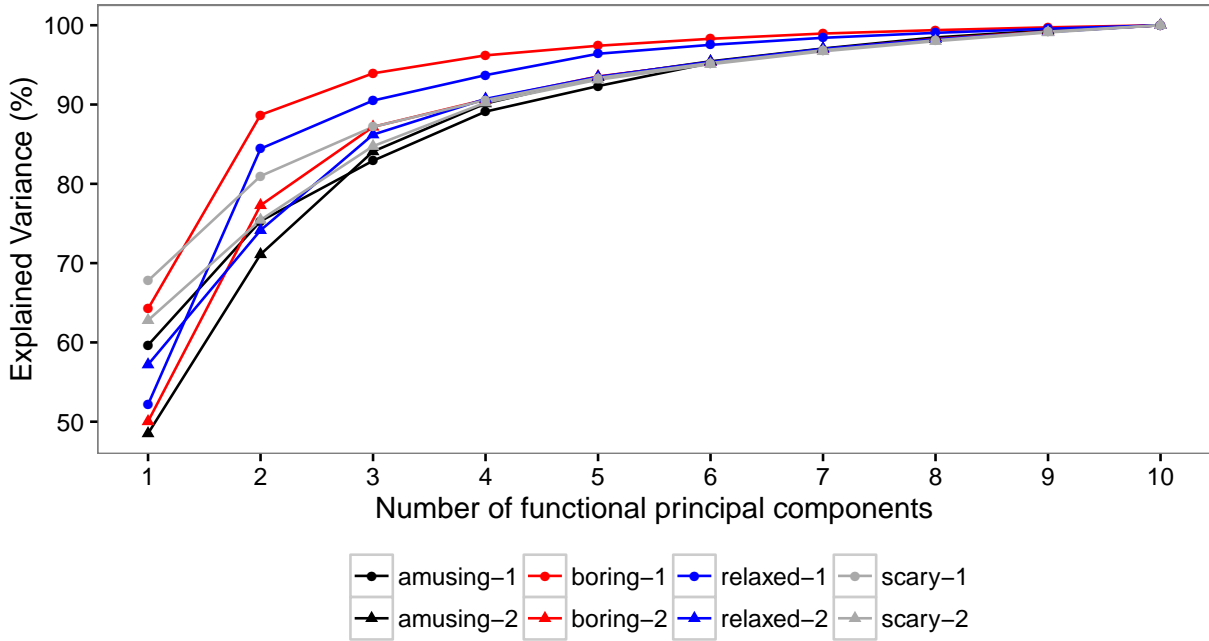


Figure 4.1: Scree plot for annotation functions seperately

4.2 Outlier removal

One of the requirements for the characteristic annotations is that it needs to represent a set of annotation functions from different subjects. The emphasis lies on the precision of the characterisation as opposed to characterising as many annotation functions as possible. A characteristic annotation in the situation of the right panel of Figure 1.4 is required to represent the annotations in the middle of the upper left quadrant. There three outlying annotations should not be taken into account. This describes the approach taken here: To remove all annotations that differ from the rest and then calculating the characteristic annotation.

As seen in Figure 3.2 and 3.4, the principal component scores are able to quantify the dissimilarity of the annotation functions: If the euclidean distance of the score vectors for two different annotation functions is large then the annotation functions themselves are different. This relation is exploited to remove outlying observations that bias the characteristic annotation.

To take into account the shape of the data cloud, the Mahalanobis distance (MD) is used to calculate the distances of the principal component score values. The MD for the annotation data is a covariance-weighted measure of each annotation functions' distance to the origin,

$$MD = (\hat{\boldsymbol{\rho}}^T \mathbf{C}^{-1} \hat{\boldsymbol{\rho}})^{-1/2} \quad (4.1)$$

since the mean function has score $\mathbf{0}$ by definition. The covariance matrix of the score values is denoted by \mathbf{C} . In fact, a robust version of MD [65] is used in order to detect outlying score values. The main idea is to use robust estimators for mean and covariance in (4.1) since they are sensitive to outliers. Details are given in the publication.

Figure 4.2 illustrates the approach: From left to right, the two top plots show the outlier detection in the vector space of the principal component scores and the function space of the annotation functions. The left plot shows all 30 annotations of one of the amusing video stimuli. The red points are outliers based on the robust Mahalanobis distance [66]. Gray points indicate non-outlying functional annotations. The top right plot shows how the annotations that have been marked as outliers look like (also highlighted in red). It can be verified that the outlier selection seems to correctly mark the annotation that runs through the lower-valence higher-arousal quadrant. This annotation behaviour is not expected for the amusing video stimuli and deviates strongly from the other subjects. The selection of the right annotations is somewhat vague: Their behaviour seems to be in line with this video type although the amplitude of valence seems to be higher.

4.3 Characteristic Annotations

The approach for calculating the characteristic annotation is straightforward: It is simply the mean function of the reduced set of annotation functions after the outliers have been removed. In the top left plot Figure 4.2 the mean function is indicated by a black dot. By definition it has score values $\mathbf{0}$. The mean function of the adjusted data is denoted by the blue asterisk. The outlier removal results only in a slightly different position compared to the original mean

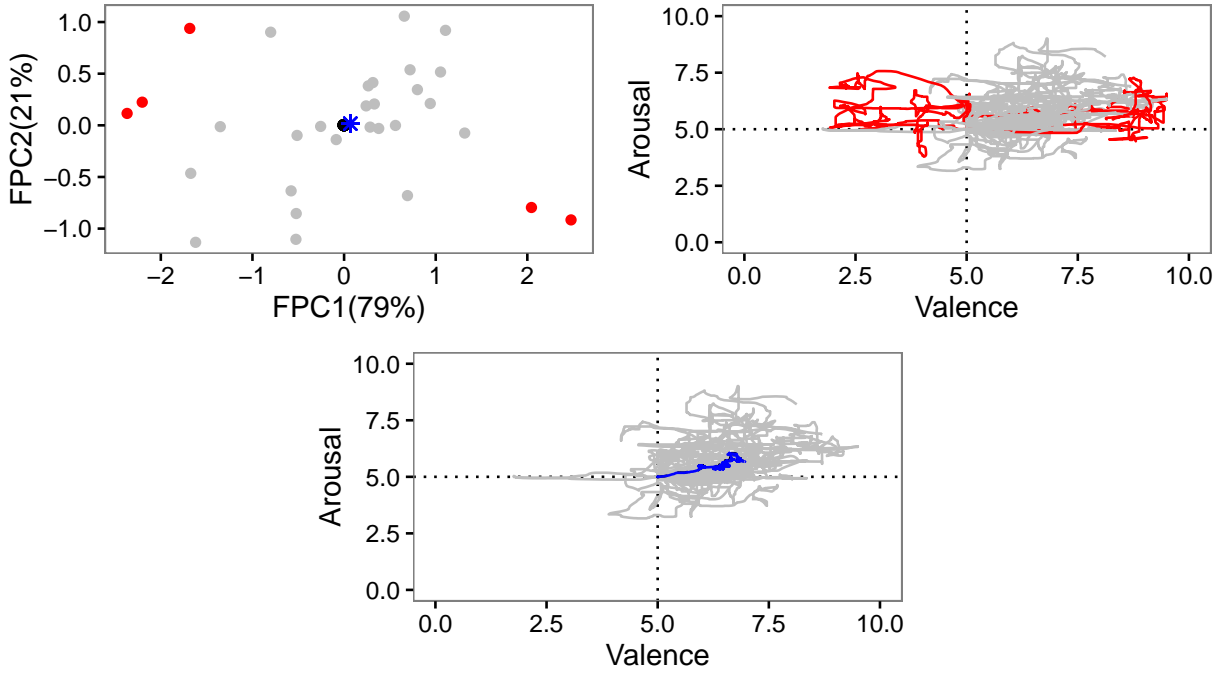


Figure 4.2: Multivariate outlier detection using the eigenfunction space (top left) and the corresponding outlying annotation functions (top right). Characteristic annotation calculation is based on a reduced set of annotation functions (bottom).

function. The bottom plot shows the reduced annotations and the two mean functions their corresponding colors. Also here the differences between characteristic annotation and mean function seem to be not visible.

A clearer assessment of the resulting characteristic annotation can be performed by means of Figure 4.3. The figure shows the resulting characteristic annotation for valence and arousal separately. The colored lines mark the different approaches for calculation of characteristic annotations: The solid black line and the dashed black line are the pointwise and functional mean which are expected to be quite similar since the smoothing shows effect in segments of single annotations where there are abrupt changes. Since these single annotations are all averaged here which is why these differences between pointwise and functional mean will be quite small. The green line marks the pointwise median which seems to be slightly more unstable than the characteristic annotation. The blue line marks the characteristic annotation. The black lines are completely covered by the characteristic annotation which becomes obvious from the eigenfunctions space representation in the previous plot: Black and blue dots are close

to each other which indicates that the resulting functions will show the same behaviour.

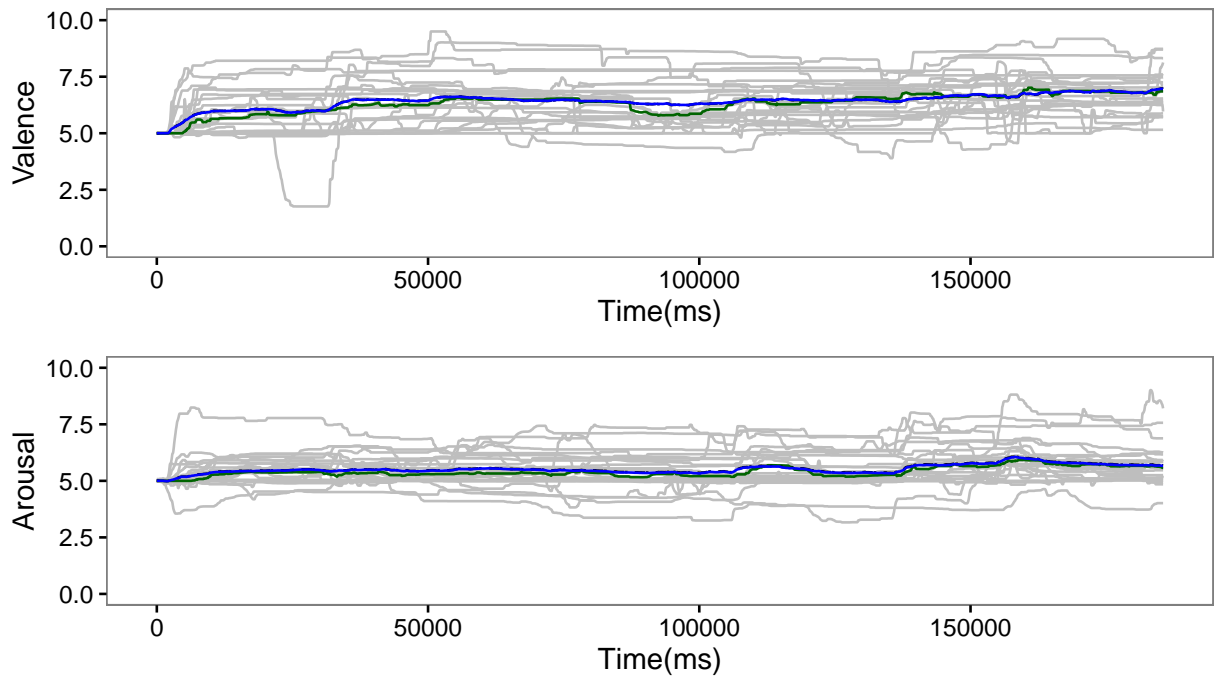


Figure 4.3: Characteristic Annotations for Valence and Arousal

Chapter 5

Discussion and Outlook

The aim of this work was to address the challenges of continuous annotations [3] using a new perspective on the data by applying methodology from functional data analysis [38, 35]. Previous work [30, 26] did not take into account the continuous nature and serial structure of the data and used methods based on restrictive assumptions [29, 31]. Consequently, the inherent structure of the data could not reflect in the methodology. By adding a new perspective to the data, that is, regarding each of the continuous annotations as a multivariate functional observation [34], the methodological shortcomings were addressed properly. The strength of the FDA based approach presented here lies in its capability to provide a functional representation that provides noise reduction. Furthermore, it was used to reduce the complexity of annotation functions in the context of multivariate functional principal component analysis [56].

With respect to the novel joystick annotation tool, this approach was applied to annotation data of 30 subjects on 8 different video stimuli pertaining to 4 affect states. It was shown that the eigenfunction representation estimated through P-Splines and MFPCA not only allows to properly explain the variance in the data, but also provides an expedient link between function space and vector space. In this way, complex annotation functions can be analysed using simple methods from multivariate statistics.

Results of Functional Representation using P-Splines

The P-Spline fit performs quite well in terms of reducing the amount of data points to only 820 spline coefficients. Since the signal-to-noise ratio is very high in the annotation data, the GCV yields a low penalisation as shown in Figure 2.10. This is expected to be a common behaviour of sensor data. Furthermore, the estimation of smooth derivatives was integrated easily into the P-Spline representation and allows to further investigate the joystick activity of the participants during the annotation process. Another important benefit of the functional representation is somewhat less obvious and pertains to the simplification of the eigenequation (3.3) for continuous data, which is a major step in the calculation of the MFPCA.

Opposed to the promoted ease of using P-Splines to avoid the challenge of choosing suitable parameters [44, 45], this choice does not become trivial. Especially the amount of data points per annotation results in a computational restriction that constrains an arbitrary choice of the basis size. Furthermore, the `fda` package [39] does not seem to be designed for such dense data since the computation time for estimating the P-Spline fits is rather high. Switching to the Matlab implementation might improve the speed of the calculations. Also the package documentation [40] is sometimes too lean to be helpful and an update to functional data analysis of sensor data would preserve the popularity of the package.

Results of MFPCA

The application of the MFPCA approach [56] yields very convincing results. It allows to quickly estimate a joint eigenfunction space for all 240 annotations of the 8 videos that are defined on different temporal domains. Accordingly, each annotation function can be represented by only a small number of principal component scores. This allows the analysis of this complex functions in a simple low-dimensional vector space as shown in Figures 3.2 to 3.4. Also the R implementation of this method, MFPCA [57], performs very well for the annotation data.

However, the conversion from the data format required for the `fda` package to the `funData` format [67] for the MFPCA calculation is somewhat cumbersome. This is because the func-

tional representation data was created using the `fda` package that has limited compatibility with the `funData` format.

Results of Characteristic Annotation

The MFPCA approach can also yield stimuli-based eigenfunction spaces for each of the 8 videos in order to calculate characteristic functions. This approach provides interesting results since it only focusses on comparing each of the annotations pertaining to one stimuli and thus to identify outliers using simple multivariate methods as shown in Figure 4.2. Using the Mahalanobis distance seems to be a reasonable choice for outlier removal. Equation (4.1) indicates that there might be room for improvement since the theoretical properties of the eigenfunction space allow easier calculation of the variances as they are known from Theorem 2.

Unfortunately, the difference between characteristic annotation and the mean function seems to be rather small. This could be due to an unidentified source of variation.

Overall Results

It was shown that the proposed FDA-based methodology is able to fully adress the three main challenges pertaining to continuous annotations:

- *Representing Multiple Subjective Ratings*: Using P-Splines, the data is represented through a basis of substantially lower dimension than the original data. Yet, the functional representation is powerful enough to keep the most important functional features of the original annotations. The resulting annotation functions as well as their derivatives are smooth and can be evaluated at arbitrary time points within their domains.
- *Comparing Multiple Subjective Ratings*: Due to the advantageous properties of the eigenfunction representation through MFPCA, all 240 annotations can be expressed in a simple vector space. The similarity of the different annotation functions is assessable in terms of euclidean distances. Annotation functions of similar stimuli form clusters that can be identified.

- *Combining Multiple Subjective Ratings*: Based on the MFPCA method, outlying annotation functions that display a different structure can be detected and removed in order to estimate a ground truth annotation.

Despite these positive outcomes, the results for the characteristic annotation are behind expectations. Possible reasons could be a high annotational heterogeneity between subjects and the presence of phase variation in the data. The latter would obscure sharp timing features and veil them when combining multiple annotations. The phase variation in the data could be due to the individual reaction times of the subjects, which is a known matter for continuous annotations [3]. Additionally, the sample size of 30 subjects and 2 videos per video stimuli yields only a limited generalisability of the results. This might affect the outlier removal for the characteristic annotations since the analysis of only 30 principal component score vectors has only limited power.

Outlook

The work presented here yields further pointers for improving the methodology for the analysis of joystick annotation data and continuous annotations in general. Future research needs to address:

- The identification of phase variation: New approaches for continuous registration and time-warping [68] may identify new sources of variation in the data and yield sharper functional features, thus enhance the power of characteristic annotations.
- Derivation of functional features: Closely related to the first point is the derivation of informative functional features [69] to improve the separability of different stimuli groups or to be able to cluster groups of subjects showing similar annotation behaviour.
- Extending repeated measurement methodology to functional observations: To further isolate the effects of the stimuli on the functional observations, the experimental design could be incorporated in the FDA methodology. Hypotheses can then be tested using functional extensions of RM MANOVA.
- Estimation of a closed form model: Similar to [70, 71] the annotation data can be modeled to be able to identify salient segments.

The present thesis shall be concluded by adding another perspective to [34] as shown in Figure 5.1. The additional column describes the data situation for the annotation data. This is a very common situation in many sensor-driven applications. One example are Bionic applications, where muscular activity is measured at different positions of the human arm in order to control a prosthetic device. In these scenarios, the resulting data consist of several functions that have different domains. The functional perspective taken in this thesis allows to simplify the analysis and yields a promising new approach to the data.

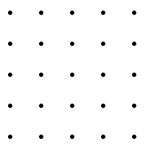

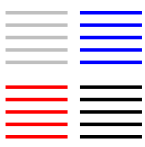
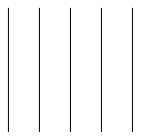

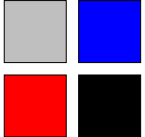
		Number of covariates		
		$p < \infty$	$p = \infty$	$p = \infty \times \dots \times \infty$
Number of observ. entities	$n < \infty$			
	$n = \infty$			
Data		x_{i1}, \dots, x_{ip}	$x_i(t), 0 \leq t \leq T$	$x_i(\mathbf{t}), \mathbf{t} = (t_1, \dots, t_d)$

Figure 5.1: More possible domains for statistical modelling of observation data referring to [34] and Figure 1.5. Using [56], the covariates per observation unit can be continuous and multivariate on different domains.

Bibliography

- [1] P. Rani, N. Sarkar, C. A. Smith, and L. D. Kirby, "Anxiety detecting robotic system—towards implicit human-robot collaboration," *Robotica*, vol. 22, no. 01, pp. 85–95, 2004.
- [2] R. W. Picard, *Affective Computing*. Cambridge, MA, USA: MIT Press, 1997.
- [3] A. Metallinou and S. Narayanan, "Annotation and processing of continuous emotional attributes: Challenges and opportunities," in *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, pp. 1–8, IEEE, 2013.
- [4] E. L. van den Broek and J. H. Westerink, "Considerations for emotion-aware consumer products," *Applied ergonomics*, vol. 40, no. 6, pp. 1055–1064, 2009.
- [5] J. H. Janssen, E. L. van den Broek, and J. H. Westerink, "Personalized affective music player," in *Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on*, pp. 1–6, IEEE, 2009.
- [6] E. L. Van den Broek, "Ubiquitous emotion-aware computing," *Personal and Ubiquitous Computing*, vol. 17, no. 1, pp. 53–67, 2013.
- [7] E. L. Broek, V. Lisy, J. H. Westerink, M. H. Schut, and K. Tuinenbreijer, "Biosignals as an advanced man-machine interface," 2009.
- [8] M. El Ayadi, M. S. Kamel, and F. Karray, "Survey on speech emotion recognition: Features, classification schemes, and databases," *Pattern Recognition*, vol. 44, no. 3, pp. 572–587, 2011.

- [9] H. Gunes and M. Piccardi, "Automatic temporal segment detection and affect recognition from face and body display," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 1, pp. 64–84, 2009.
- [10] M. M. Bradley and P. J. Lang, "Measuring emotion: The self-assessment manikin and the semantic differential," *Journal of behavior therapy and experimental psychiatry*, vol. 25, no. 1, pp. 49–59, 1994.
- [11] J. H. Janssen, P. Tacken, J. de Vries, E. L. van den Broek, J. H. Westerink, P. Haselager, and W. A. IJsselsteijn, "Machines outperform laypersons in recognizing emotions elicited by autobiographical recollection," *Human–Computer Interaction*, vol. 28, no. 6, pp. 479–517, 2013.
- [12] R. D. Ray, "Emotion elicitation using films," *Handbook of emotion elicitation and assessment*, pp. 9–28, 2007.
- [13] J. J. Gross and R. W. Levenson, "Emotion elicitation using films," *Cognition & emotion*, vol. 9, no. 1, pp. 87–108, 1995.
- [14] R. Cowie, E. Douglas-Cowie, S. Savvidou*, E. McMahon, M. Sawey, and M. Schröder, "‘feeltrace’: An instrument for recording perceived emotion in real time," in *ISCA tutorial and research workshop (ITRW) on speech and emotion*, 2000.
- [15] R. Cowie, M. Sawey, C. Doherty, J. Jaimovich, C. Fyans, and P. Stapleton, "Gtrace: General trace program compatible with emotionml," in *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*, pp. 709–710, IEEE, 2013.
- [16] G. Laurans, P. M. Desmet, and P. Hekkert, "The emotion slider: A self-report device for the continuous measurement of emotion," in *Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on*, pp. 1–6, IEEE, 2009.
- [17] A. M. Ruef and R. W. Levenson, "Continuous measurement of emotion," *Handbook of emotion elicitation and assessment*, pp. 286–297, 2007.

- [18] F. Nagel, R. Kopiez, O. Grewe, and E. Altenmüller, "Emujoy: Software for continuous measurement of perceived emotions in music," *Behavior Research Methods*, vol. 39, no. 2, pp. 283–290, 2007.
- [19] T. Eerola and J. K. Vuoskoski, "A comparison of the discrete and dimensional models of emotion in music," *Psychology of Music*, vol. 39, no. 1, pp. 18–49, 2011.
- [20] L. F. Barrett, "Discrete emotions or dimensions? the role of valence focus and arousal focus," *Cognition & Emotion*, vol. 12, no. 4, pp. 579–599, 1998.
- [21] J. A. Russell, "Core affect and the psychological construction of emotion," *Psychological review*, vol. 110, no. 1, p. 145, 2003.
- [22] P. J. Lang, "The emotion probe: Studies of motivation and attention.," *American psychologist*, vol. 50, no. 5, p. 372, 1995.
- [23] G. N. Yannakakis and H. P. Martinez, "Grounding truth via ordinal annotation," in *Affective Computing and Intelligent Interaction (ACII), 2015 International Conference on*, pp. 574–580, IEEE, 2015.
- [24] J. Antony, K. Sharma, C. Castellini, E. van den Broek, and C. Borst, "Continuous affect state annotation using a joystick-based user interface," in *Proceedings of Measuring Behavior*, 2014.
- [25] J. Russell, "A circumplex model of affect," *J. Personality and Social Psychology*, vol. 39, pp. 1161–78, 1980.
- [26] J. Antony, "Identification of affect patterns from bio-signals," Master's thesis, Fachhochschule Aachen, 2014.
- [27] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy, "Learning from crowds," *Journal of Machine Learning Research*, vol. 11, no. Apr, pp. 1297–1322, 2010.

- [28] T. A. Walls and J. L. Schafer, *Models for intensive longitudinal data*. Oxford University Press, 2006.
- [29] H. Sørensen, J. Goldsmith, and L. M. Sangalli, “An introduction with medical applications to functional data analysis,” *Statistics in medicine*, vol. 32, no. 30, pp. 5222–5240, 2013.
- [30] K. Sharma, C. Castellini, and E. L. van den Broek, “Continuous affect state annotation using a joystick-based user interface: Exploratory data analysis,” 2016.
- [31] D. J. Levitin, R. L. Nuzzo, B. W. Vines, and J. Ramsay, “Introduction to functional data analysis.,” *Canadian Psychology/Psychologie canadienne*, vol. 48, no. 3, p. 135, 2007.
- [32] R. Gupta, K. Audhkhasi, Z. Jacokes, A. Rozga, and S. Narayanan, “Modeling multiple time series annotations based on ground truth inference and distortion,” *IEEE Transactions on Affective Computing*, vol. PP, no. 99, pp. 1–1, 2016.
- [33] K. Audhkhasi and S. Narayanan, “A globally-variant locally-constant model for fusion of labels from multiple diverse experts without using reference labels,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 4, pp. 769–783, 2013.
- [34] J. Ramsay, “When the data are functions,” *Psychometrika*, vol. 47, no. 4, pp. 379–396, 1982.
- [35] J. Ramsay and B. Silverman, *Functional data analysis*. Springer Science & Business Media, 2005.
- [36] S. Ullah and C. F. Finch, “Applications of functional data analysis: A systematic review,” *BMC medical research methodology*, vol. 13, no. 1, p. 43, 2013.
- [37] A. Cuevas, “A partial overview of the theory of statistics with functional data,” *Journal of Statistical Planning and Inference*, vol. 147, pp. 1–23, 2014.
- [38] J.-L. Wang, J.-M. Chiou, and H.-G. Müller, “Functional data analysis,” *Annual Review of Statistics and Its Application*, vol. 3, pp. 257–295, 2016.

- [39] J. O. Ramsay, H. Wickham, S. Graves, and G. Hooker, *fda: Functional data analysis*, 2014. R package version 2.4.4.
- [40] J. O. Ramsay, G. Hooker, and S. Graves, *Functional data analysis with R and MATLAB*. Springer Science & Business Media, 2009.
- [41] R. Schafer, “What is a savitzky-golay filter?,” *IEEE Signal Processing Magazine*, vol. 28, pp. 111–117, 2011.
- [42] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: Data mining, inference and prediction*. Springer, 2 ed., 2008.
- [43] L. Fahrmeir, T. Kneib, S. Lang, and B. Marx, *Regression: Models, methods and applications*. Springer Science & Business Media, 2013.
- [44] P. H. Eilers, B. D. Marx, and M. Durbán, “Twenty years of p-splines,” *SORT-Statistics and Operations Research Transactions*, vol. 39, no. 2, pp. 149–186, 2015.
- [45] P. H. Eilers and B. D. Marx, “Splines, knots, and penalties,” *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 6, pp. 637–653, 2010.
- [46] C. De Boor, *A practical guide to splines; rev. ed.* Applied mathematical sciences, Berlin: Springer, 2001.
- [47] P. H. Eilers and B. D. Marx, “Flexible smoothing with b-splines and penalties,” *Statistical science*, pp. 89–102, 1996.
- [48] R. Myers, “Classic and modern regression with applications,” *PWS-KENT, Boston, Mass*, 1990.
- [49] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes: The art of scientific computing*. New York, NY, USA: Cambridge University Press, 3 ed., 2007.
- [50] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: Nsga-ii,” *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

-
- [51] O. Mersmann, *mco: Multiple Criteria Optimization Algorithms and Related Functions*, 2014. R package version 1.0-15.1.
- [52] I. T. Jolliffe and J. Cadima, “Principal component analysis: A review and recent developments,” *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, 2016.
- [53] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2002.
- [54] K. Karhunen, “Ueber lineare methoden in der wahrscheinlichkeitsrechnung,” *Annales Academiae Scientiarum Fennicae*, vol. 37, pp. 3–79, 1947.
- [55] M. Loève, “Fonctions aleatoires du second ordre,” 1948.
- [56] C. Happ and S. Greven, “Multivariate functional principal component analysis for data observed on different (dimensional) domains,” *Journal of the American Statistical Association*, no. just-accepted, 2016.
- [57] C. Happ, *MFPCA: Multivariate functional principal component analysis for data observed on different dimensional domains*, 2016. R package version 1.0-1.
- [58] J. Kleffe, “Principal components of random variables with values in a separable hilbert space,” *Statistics: A Journal of Theoretical and Applied Statistics*, vol. 4, no. 5, pp. 391–406, 1973.
- [59] J. Jacques and C. Preda, “Model-based clustering for multivariate functional data,” *Computational Statistics & Data Analysis*, vol. 71, pp. 92–106, 2014.
- [60] J.-M. Chiou, Y.-T. Chen, and Y.-F. Yang, “Multivariate functional principal component analysis: A normalization approach,” *Statistica Sinica*, pp. 1571–1596, 2014.
- [61] R. B. Cattell, “The scree test for the number of factors,” *Multivariate behavioral research*, vol. 1, no. 2, pp. 245–276, 1966.

- [62] J. R. Berrendero, A. Justel, and M. Svarc, “Principal components for multivariate functional data,” *Computational Statistics & Data Analysis*, vol. 55, no. 9, pp. 2619–2634, 2011.
- [63] D. Poskitt and A. Sengarapillai, “Description length and dimensionality reduction in functional data analysis,” *Computational Statistics & Data Analysis*, vol. 58, pp. 98–113, 2013.
- [64] L. Fahrmeir, A. Hamerle, and G. Tutz, *Multivariate statistische Verfahren*. Walter de Gruyter GmbH & Co KG, 1996.
- [65] P. Filzmoser, R. G. Garrett, and C. Reimann, “Multivariate outlier detection in exploration geochemistry,” *Computers & geosciences*, vol. 31, no. 5, pp. 579–587, 2005.
- [66] P. Filzmoser and M. Gschwandtner, *mvoutlier: Multivariate outlier detection based on robust methods*, 2015. R package version 2.0.6.
- [67] C. Happ, *funData: An S4 class for functional data*, 2016. R package version 1.0.
- [68] J. S. Marron, J. O. Ramsay, L. M. Sangalli, A. Srivastava, *et al.*, “Statistics of time warpings and phase variations,” *Electronic Journal of Statistics*, vol. 8, no. 2, pp. 1697–1702, 2014.
- [69] K. Fuchs, J. Gertheiss, and G. Tutz, “Nearest neighbor ensembles for functional data with interpretable feature selection,” *Chemometrics and Intelligent Laboratory Systems*, vol. 146, pp. 186–197, 2015.
- [70] T. Hastie, E. Kishon, M. Clark, V. Clayton, and A. J. Fan, “A model for signature verification.” 1992.
- [71] T. Hastie, E. Kishon, M. Clark, and J. Fan, “A model for signature verification,” in *Systems, Man, and Cybernetics, 1991. Decision Aiding for Complex Systems, Conference Proceedings., 1991 IEEE International Conference on*, pp. 191–196, IEEE, 1991.

Declaration of Authorship

I hereby declare that the thesis submitted is my own unaided work. All direct or indirect sources used are acknowledged as references.

Marius Wagner

Munich, 15th May 2017