Karen Fuchs

Functional Data Analysis Methods for the Evaluation of Sensor Signals

Dissertation an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

vorgelegt am 30. August 2017

Karen Fuchs

Functional Data Analysis Methods for the Evaluation of Sensor Signals

Dissertation an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

vorgelegt am 30. August 2017

Erstgutachter: Herr Prof. Dr. Gerhard Tutz
Zweitgutachter: Herr Prof. Dr. Friedrich Leisch

Tag der Disputation: 22. Dezember 2017

Danksagung

Viele Menschen haben zum Gelingen dieses Promotionsvorhabens beigetragen.
Ich bin ihnen allen von Herzen dankbar.

Insbesondere möchte ich meinen Doktorvater Herrn Prof. Dr. Gerhard Tutz hervorheben, der mich von Anfang an vorbehaltlos unterstützt hat. Seine Aufgeschlossenheit und sein Interesse gegenüber dem Thema machten meine Promotion mitunter erst möglich. Auch wenn wir nicht ganz regelmäßig die Möglichkeit zu persönlichen Gesprächen hatten unterstützte er mich wo und wann immer nötig, ohne dass sein Enthusiasmus und seine Geduld in Zeiten des verlangsamten Voranschreitens der Arbeit je nachgelassen haben.

Genauso sei Frau Prof. Dr. Sonja Greven erwähnt, die sich sehr spontan bereit erklärte meine Arbeit zu unterstützen. Sie nahm mich in ihre Arbeitsgruppe auf und begleitete mich bei meinem Einstieg in die Welt der funktionalen Datenanalyse. Durch unermüdliche Gespräche und Korrekturen führte sie mich an die Publikation wissenschaftlicher Ergebnisse heran.

Herrn Prof. Dr. Friedrich Leisch danke ich für die Bereitschaft das externe Gutachten zu übernehmen.

Natürlich gebührt meiner Abteilung bei der Siemens AG mein besonderer Dank, vor allem Herrn Prof. Dr. Maximilian Fleischer und Frau Dr. Kerstin Wiesner-Fleischer für die Initialisierung des Promotionsvorhabens und die stete Unterstützung in allen Bereichen.

Es sei auch an alle weiteren Koautoren gedacht, ohne die die dieser Arbeit zugehörigen Publikationen so nicht zustande gekommen wären. Mit ihnen allen war die gemeinsame Er- und Bearbeitung unserer Projekte nicht nur erfolgreich, sondern hat auch Spaß gemacht. Besonders sollen Dr. Fabian Scheipl und Prof. Dr. Jan Gertheiss Erwähnung finden, von denen ich in zahlreichen Diskussionen und emails auch Manches über Statistik im Allgemeinen gelernt habe.

Viele Grüße an die Biostatistik-Arbeitsgruppe (hoffentlich nicht das letzte Mal). Sie hat mich herzlich aufgenommen und Zweifel immer wieder im Keim erstickt. Merci!

Weiterhin möchte ich mich beim gesamten Institut für Statistik bedanken, wo einem immer freundlich begegnet wird. Vielen Dank an Micha Schneider, Frau Oberschmidt und Frau Robeck, die für jede Frage offen waren.

Ebenfalls ein grosser Dank an die Biologie- und Gassensorgruppe bei der Siemens AG für ihre Unterstützung bei der Einarbeitung in die jeweiligen Themen und Labore sowie die immer wieder in Anspruch genommenen Hilfestellungen. Herrn Dr. Tobias Paust für viele Tipps und Tricks rund um und die Übersetzung einer meiner Methoden

in Matlab. Herrn Dr. Clemens Otte und Herrn Dr. Hans-Georg Zimmermann für fruchtbare Diskussionen. Und nicht zuletzt Allen für die angenehme und oft lustige Arbeitsatmosphäre!

An dieser Stelle sei auch nochmal allen Korrekturlesern gedankt, vor allem einigen lieben Menschen aus der Biostatistik-AG und bei Siemens - und Anne für den letzten Schliff.

Meine Freunde haben meine (manchmal schwer schwankenden) Launen ertragen und reden immer noch mit mir. Danke dafür! :)

–

Am meisten aber hat meine Familie zu meiner Unterstützung beigetragen, indem sie mich bei allen Höhen und Tiefen begleitet und nie an der erfolgreichen Beendigung der Arbeit gezweifelt hat. Ich hab' euch lieb!
Das Gleiche gilt auch für meine Schwiegerfamilie, und nicht zuletzt für Remik. Ohne dich hätte ich (vielleicht einfach zu) früh aufgegeben. Kocham cie, wróbelek mój!

Zusammenfassung

Dank des technologischen Fortschritts sowohl auf Hardware- als auch auf Software-Ebene werden die Möglichkeiten der Datengenerierung immer vielschichtiger. Dabei nimmt nicht nur die Menge an Daten zu, sondern häufig werden auch Signale von beispielsweise unterschiedlichen Sensoren simultan aufgenommen. Dadurch erhöht sich die Komplexität der Datenstruktur, und die Zusammenhänge zwischen den Messungen können zusätzliche Information liefern. Die vorliegende Arbeit beschäftigt sich mit verschiedenen Möglichkeiten, die Signale solcher Sensoren statistisch auszuwerten. Dabei werden die Sensorsignale als funktionale Kovariablen betrachtet und über statistische Modelle mit skalaren Zielgrößen verknüpft.

Dem ersten Teil der Arbeit liegt die Idee zugrunde, dass ein generalisiertes funktionales lineares Regressionsmodell, in das multiple funktionale Kovariablen rein additiv eingehen, den Zusammenhang zwischen Einfluss- und Zielgrößen möglicherweise unzureichend beschreibt. Deswegen wird es um einen Kovariablen-Interaktionsterm erweitert, der die Überprüfung der Additivitätsannahme ermöglicht. In einer Simulationsstudie wird die Schätzung des Interaktionsterms bezüglich verschiedener Datensituationen analysiert. Am Beispiel von zellbasierten Biochip- und Spektroskopiedaten wird gezeigt, dass die Vorhersagegüte der Zielgröße gegenüber Vergleichsmethoden gesteigert werden kann, wenn die Daten die Aufnahme der Interaktion in das Modell rechtfertigen.

Die Unterscheidung chemisch ähnlicher Substanzen mittels Sensorsignalen, die auf chemisch-physikalischen Messprinzipien basieren, ist Gegenstand des zweiten Teils der Arbeit. Ein hier entwickelter, nicht-parametrischer Ensembleansatz bezieht neben der funktionalen Natur der Kovariablen auch deren Charakteristika mit ein. Durch eine Penalisierung der Ensemblekoeffizienten wird eine interpretierbare Merkmals- und, bei multiplen funktionalen Kovariablen, Variablenselektion ermöglicht. In einer Simulationsstudie wird die Interpretierbarkeit und die – verglichen mit anderen Klassifizierungsmethoden – gute Vorhersagegüte nachgewiesen. Anhand realer Daten kann gezeigt werden, dass die geschätzten Ensemblekoeffizienten sinnvoll interpretiert werden können und dass sie zusätzliche Erkenntnisse über die unterscheidenden Charakteristika der Daten liefern können.

Des Weiteren wird ein zweiter Schätzansatz für das obige Ensemble entwickelt. Dieser ermöglicht es dem Anwender aus verschiedenen Penalisierungen zu wählen, von denen

einige die Aufnahme klassenspezifischer Koeffizienten in das Ensemble zulassen. Die Unterschiede der beiden Schätzansätze werden diskutiert, und die Ergebnisse verglichen.

Abschließend wird ausführlich auf mögliche Weiterentwicklungen der vorgestellten Modelle eingegangen.

Summary

Due to technological advances in both hardware and software, data that has to be analysed is becoming more and more complex. Not only the pure quantity of available data is growing rapidly, but also the complexity of the data structure, for example, if signals of various sensor types are measured concurrently. Also, the interaction of these signals can contain information. This thesis developes and examines statistical approaches for the evaluation of such sensor signals, which will be regarded as functional covariates. They will be related to scalar responses via the model approaches described in the following.

The first part of the thesis is motivated by the notion that a generalized functional regression model, which assumes the effects of multiple functional covariates to be purely additive, does not exhaustively determine the relationship between a scalar response and the functional covariates. Thus, a covariate interaction term is included that allows to verify the assumption of additivity. By means of a simulation study, the estimation of this interaction term is examined for differing data situations. Cell based biochip data as well as spectroscopic data are used to demonstrate that, if the interaction contains information, the prediction performance can be improved by including the interaction term in the regression model. The prediction results are better or comparable relative to competitive methods.

The second part of the thesis focuses on the discrimination of substances that are, from a chemical point of view, similar. A non-parametric ensemble approach is developed, which comprises both the functional nature of the sensor signals as well as their characteristic features. Due to a penalty put on the ensemble coefficients, the ensemble extracts interpretable features and yields variable selection if multiple functional covariates are included. A simulation study is conducted to demonstrate the ability to select relevant features, giving also a good classification performance. Real world data is used to show that the interpretation of the estimated ensemble coefficients is consistent with the background knowledge of the respective data. The estimated coefficients can also offer new insights regarding the discriminative characteristics of the data.

The following part of the thesis presents an alternative estimation approach to calculate the coefficients of the ensemble introduced above. The new approach enables the user to choose from various penalties. Depending on the chosen penalty, it becomes possible to include class-specific coefficients in the ensemble. The differences between the two estimation approaches are discussed. Furthermore, their respective results are compared.

The thesis concludes with a thorough discussion of further possible developments concerning the presented models.

Contents

Chapter 1

Introduction

1.1 The Basics of Functional Data Analysis

Functional data analysis is an active field of research. The amount and diversity of functional data is growing due to technical developments in signal recording and inspires researchers in the field of statistics as well as practitioners.
The term "functional data" comprises all data that is supposed to be realized points of an underlying (quasi-) continuous process, which in turn should be describable by a function (Ramsay and Silverman, 2005; Ferraty and Vieu, 2006; Horvath and Kokoszca, 2012). As an example, consider the monthly temperature data of Canadian weather stations presented in Ramsay and Silverman (2005). Figure 1.1 shows the temperature courses at 35 Canadian weather stations, averaged from 1960 to 1994. The black dots indicate the 12 points in time at which measurements actually took place. At least theoretically, the temperature could be measured at any point in time, i.e. arbitrarily close on the time domain. Thus, the 12 measurement points can be taken as realized points of an underlying continuous process, namely temperature change. The dashed lines in Figure 1.1 show the interpolation between the measurement points. They give an idea of the approximate shapes of the temperature change function per weather station.

In the theory of functional data analysis, the (quasi)-continuous process, i.e. a variable X, is assumed to be a random element of a function space. By choosing the function space, the definition of e.g. the mean function or the covariance operator arise from the respective mathematical framework. The realizations of X, namely the data $x(t)$, are observed on a domain $\mathcal{T} \ni t$. Often, a model or theoretical consideration makes use of a defined norm or inner product. This is why common choices for the function space are the separable Banach space $\mathcal{C}(\mathcal{T}) \ni X$ of real continuous functions, or, if an inner product is needed, the seperable Hilbert space $\mathcal{L}^2(\mathcal{T}) \ni X$ of square integrable functions (the inner product induces a norm, and Hilbert spaces are by construction also Banach spaces).

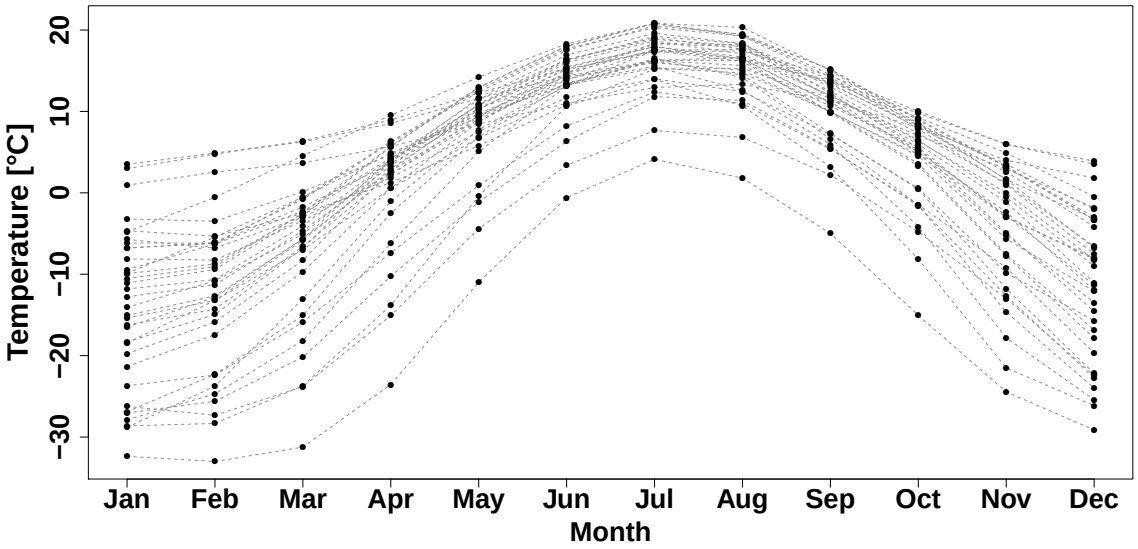


Figure 1.1: Temperature course at 35 Canadian weather stations, averaged from 1960 to 1994. The black dots give the 12 points per weather station at which measurements actually took place. The dashed lines show the interpolation between the measurement points per weather station.

In both cases, X is defined on some finite domain $\mathcal{T}$, with typically, but not necessarily $\mathcal{T} \subseteq \mathbb{R}^p$. In the prevalent case of $p = 1$, the observation units are curves, as in the above temperature data example. For higher values of p, observations correspond to surfaces, images, etc.
Depending on the data at hand, the covariate X, the response Y, and potential model parameters Θ might be of functional character.

Functional data analysis allows to analyse, model and predict many of the (quasi-) continuous data measured in various fields of research. The empirical data actually available for examination is taken to be vectors comprising pointwise realizations of continuous processes. Thus, functional data is infinite dimensional in theory, and, especially when observed on dense grids, high dimensional in practice. The vectors containing the realizations represent structured objects instead of a cluster of data points, as can be understood from the weather data example. For the respective temperature data, the structure, or order, of the 12 measurement points is essential. It reflects the seasons, with low values at the beginning and end of the year, and an interjacent maximum. In contrast to non-functional data, the order of the measurement points per weather station can not be altered, since the seasonal trend would be lost. Important issues arising in the context of functional observations include intensity or phase shifts in the functions which are met by registration and alignment techniques (see Ramsay and Li, 1998, among others). Also, the sampling scheme

of the data is of importance. Usually one differs between densely or sparsely (see Zhang and Wang, 2016, for a short discussion on this topic), and possibly regularly or irregularly observed data. When handling multiple functional covariates, they could be observed on different grids or types of domain.
Accordingly, specialized plotting methods are necessary for functional data. In most cases, curves and surfaces can be presented quite easily, as described and exemplified, for instance, in Ramsay and Silverman (2005) as well as Happ (2017). For functional data of higher dimensions $p \geq 3$, convenient ways of visualization are not yet available. The notion of $x(t)$ being a realization of X often is accompanied by the assumption that the $x(t)$ are observed with measurement error. So far, most approaches imply an additive error that is independently distributed. A common way to handle data observed with measurement error is to estimate the true $x(t)$ by (penalized) smoothing techniques, such as spline or (multivariate) functional principal component (e.g. Yao et al., 2005; Happ and Greven, 2016) representations. Then, a model approach effectively operates on an approximation of $x(t)$ rather than on the originally observed data.

Regarding the vast amount of literature that has evolved in functional data analysis in the last decades, methodological developments range from (semi-) parametric to nonparametric regression to classification (e.g. Ferraty and Vieu, 2003, 2006) and clustering (Jacques and Preda, 2014; Yassouridis and Leisch, 2016, and references therein) approaches. Morris (2015) and Greven and Scheipl (2017) present a review and a discussion of functional regression methods. Overviews of established and recent methods in functional data analysis can be found in Gonzalez Manteiga and Vieu (2007), Ferraty and Romain (2011), Cuevas (2014), Bongiorno et al. (2014), or Wang et al. (2016). Among the research concerned with theoretical aspects are He et al. (2000), Bosq (2000), Cardot and Sarda (2005), Ferraty and Vieu (2006), Bosq and Blanke (2007), Delaigle and Hall (2010), Horvath and Kokoszca (2012), or Scheipl and Greven (2016).
The probably most popular functional modeling approach is the functional generalized linear model (Müller and Stadtmüller, 2005) with scalar and (conditionally) mutually independent response y, link function $g(\cdot)$, a square integrable functional covariate $x(t)$ defined on the respective Hilbert space, an intercept term β_0, and a coefficient function $\beta(t)$,

$$g\left(E(y)\right) =: g(\mu) = \beta_0 + \int_{\mathcal{T}} x(t)\beta(t)dt. \tag{1.1}$$

Here, the intercept term and the coefficient function are unknown and have to be estimated from the data.
An explicit example for a functional regression model is the functional linear model,

$$y = \beta_0 + \int_{\mathcal{T}} x(t)\beta(t)dt + \varepsilon, \tag{1.2}$$

with additive error ε (following for example a normal distribution with zero mean and variance σ^2, $\varepsilon \sim N(0, \sigma^2)$).

The functional logit model

$$y = \pi + \varepsilon := \frac{\exp\left(\beta_0 + \int_{\mathcal{T}} x(t)\beta(t)dt\right)}{1 + \exp\left(\beta_0 + \int_{\mathcal{T}} x(t)\beta(t)dt\right)} + \varepsilon \tag{1.3}$$

with $y \in \{0, 1\}$ and additive error is an example for a 2-class functional classification approach (James, 2002; Escabias et al., 2004).
Models (1.1) to (1.3) give a small impression of the potential of functional modeling approaches. They are usually estimated via maximization of the (penalized) log-likelihood, yet model estimation can be advanced in several ways. Some examples are time series analysis (see e.g. Bosq, 2000), functional boosting (Brockhaus et al., 2017), or functional mixed models (Cederbaum et al., 2016). Of course, the choice of an appropriate model and estimation approach largely depends on the data situation.

Some concepts of classical multivariate analysis can be adapted to functional data, cf. also Gonzalez Manteiga and Vieu (2007) and references therein. Nonetheless, the points mentioned above emphasize that functional data analysis is an original area of research, and there is ample room for the development of new modeling methods as well as the extension of existing approaches. This thesis is inspired by three data sets, which are introduced in Section 1.2. The main methodological advances presented in this thesis are, on the one hand, the extension of generalized models for scalar responses and functional covariates to models including linear functional interaction terms. On the other hand, a novel functional k-nearest-neighbor classification technique is developed, and different estimation approaches are evaluated. Further details are given in the guideline for the thesis in Section 1.3.

1.2 Motivating Data Sets

This thesis is motivated by three data sets of very different type, namely cell chip data, gas sensor data and spectroscopic data of fossil fuels. For all three data sets, a value at a single signal data point depends on the actual system status, which evolves from the previous states. Thus, the order of observation points plays a major role, confirming the functional character of the data sets.
By courtesy of the Siemens AG, the cell chip and spectra data sets were made publicly available. Please visit the websites given in the respective publications to access them. For more details on the materials used, the data itself as well as its acquisition, common evaluation techniques, and respective references, please refer to Appendix D.

1.2.1 Cell Chip Data

The cell chip sensors used in this study consist of a chip with three incorporated sensor types and a layer of living cells (see e.g. Lagarde and Jaffrezie-Renault, 2011; Eltzov and

Marks, 2011, for an overview of biosensors). The cells are contained in medium supplying them with nutrients. This medium can be mixed with test substances to observe the cells' reactions to these substances. The main interest in biosensors of this kind lies in environmental monitoring, as for example water quality monitoring (see e.g. Kubisch et al., 2012; Guijarro et al., 2015). Research also includes the use of cell based sensors for gas sensing (as for example in Bohrn et al., 2011).
The conclusions that can be drawn from biosensor measurements depend on the sensor. The three sensor types used here are ion sensitive field effect transistors (ISFET), an interdigitated electrode structure (IDES) and oxygen sensitive electrodes based on CLARK-electrodes. Each type is a proxy for a certain parameter resulting from the cell metabolism. ISFET-signals relate to the acidification rate of the nutrient medium, which is due to the excretion of acidic metabolites. IDES-signals measure the cellular impedance and can be used to draw conclusions about the cell morphology and cell adhesion on the surface of the sensor chip. CLARK-electrodes measure the oxygen contained in the medium as a proxy for the respiration activity of the cells (Thedinga et al., 2007; Ceriotti et al., 2007).
In the cells' habitual environment, the nutrient medium, the acquired sensor signals are stable. When the nutrient medium changes its composition, for example due to being polluted by a test substance, the cells react on this change in their environment, and the signals alter. Since these processes base on the cells' metabolism, which is unquestionable a continuous process over time, such cell chip data can be taken as functional data, with the sensor signals being functional covariates. As metric response, the values of the concentration of our test substance, which is paracetamol, will be used.

1.2.2 Gas Sensor Data

Often, the ultimate objective of evaluating gas sensor data is the identification of a gas in a mixture of gases. The applications for algorithms providing this are multifaceted. There are many studies concerned with environmental monitoring, such as (indoor) air quality monitoring (see e.g. Piedrahita et al., 2014; Masson et al., 2015). A lot of work is also done on e-noses (Peveler et al., 2013; Dymerski et al., 2013; Li et al., 2015, among others) and in medical technology, when searching for markers or patterns relating gas sensor signals to the health status of a patient (Makisimovich et al., 1996; Kim et al., 2014). For example, Bajtarevic et al. (2009) and Millonig et al. (2010) found evidence that pentanal and acetaldehyde in breathing air might to be correlated to lung cancer and liver diseases.
In this thesis, four AS-MLV metal oxid (MOX) gas sensors with a tin dioxid based sensitive layer are used. The sensitive layer was deposited on a miniaturized hotplate, such that the layer's temperature can be controlled to some extent. Temperature modulated MOX gas sensors are used to simulate several sensitive layers instead of a single one: reactions between a sensitive layer and the atmosphere depend, among other things, on the composition of the ambient air, the type of sensitive material as well as on the layers' temperature (see e.g. Lee and Reedy, 1999). Thus, applying a certain gas to the gas sensor at four

different temperatures simulates four different sensitive layers.
The gas sensors measure the electrical resistance of the sensitive layer as a signal. This resistance, on the one hand, changes with the temperature applied to the sensitive layer. On the other hand, it is influenced by numerous de- and adsorption processes between the layer and its surrounding atmosphere. Such processes do not take place instantaneously. The change of the resistance, i.e. the gas sensor signals, can thus be taken to be functional data over time.

1.2.3 Spectroscopic Data

Two data sets containing spectra of fossil fuels were provided for examination. They were recorded by two spectrometers which operate on different spectral domains. The first spectrometer, the multi purpose analyzer (MPA) by Bruker, is a near infrared (NIR) spectrometer providing a measurement range between 800nm and 2780nm. The second set of data was measured with the Compass X by BWTek, measuring in the ultraviolet-visible (UV-VIS) range between 250nm and 880nm.
Spectroscopy is a very old field of research (for example, first measurements in the infrared range go back to Herschel, 1800). In physics, spectroscopy is a widely ramified topic. The required instrumentation as well as possible applications largely depend on fundamental physical laws. For example, it is not possible to determine the concentration of inorganic compounds, such as cooking salt in water, directly by NIR spectroscopy. Inorganic compounds do not absorb in this wavelength range, and thus a NIR spectrum does not contain information concerning the substance.
Nonetheless, NIR and UV-VIS spectroscopy are interesting for a wide range of applications such as on-line process monitoring due to stable spectrometers that can be built cheaply and on a small scale. Especially NIR spectra are very smooth curves. Specific materials do not show specific features, but their overtones and combinations of the molecular vibrations superimpose. In UV-VIS spectroscopy, (valence or non-bonding) electrons are excited to higher energy levels, i.e. orbitals. For both types of spectroscopy, a more elaborated evaluation of spectra from complex samples is necessary, and often pursued in the field of chemometry. Chemometricians study spectroscopic data exensively with various methods, as in Blanco et al. (2000), Balabin and Smirnov (2012), and Prevolnik et al. (2014), to name a few. In statistics, NIR spectra became popular in the field of functional data analysis. Taking spectra to be functional data is natural, since they are defined on a wavelength range, which represents a respective range of energy and is inherently continuous. Also, the relative and mutual order of data points has a physical meaning.

1.3 Guideline for the Thesis and Contributing Manuscripts

In **Chapter 2**, generalized models for scalar responses with functional covariates are extended to include linear functional interaction terms that are interaction effects between functional variables. The coefficient functions are estimated by using basis expansions and maximization of a log-likelihood, which is penalized to impose smoothness upon the coefficient functions. The respective smoothing parameters for the penalties are estimated from the data, e.g. via generalized cross-validation. Further functional or scalar terms as well as functional interactions of higher order can be added within the same framework. The performance of the introduced approach is tested in simulations. Additionally, the model is applied to two of the motivating data sets, the spectroscopic data and the cell chip sensor data. The main aim is to predict the respective responses.

Another model class discussed in this thesis is scalar-on-function discrimination. Although much research has been done on functional regression and clustering approaches for chemometric data few classification methods exist so far. **Chapter 3** introduces an ensemble method for the classification of functional data that inherently provides automatic and interpretable feature selection. It is designed for single as well as multiple functional (and non-functional) covariates. The ensemble members are posterior probability estimates that are based on a k-nearest-neighbor approach. The ensemble allows for feature selection by including members that are calculated from various semi-metrics used in the k nearest neighbor approach, where a particular semi-metric represents a specific curve feature. Each ensemble member, and thus each curve feature, is weighted by an unknown coefficient. The coefficients of all semi-metrics are estimated using a proper scoring rule with implicit Lasso-type penalty, such that some coefficients can be estimated to be zero. Thus, the ensemble automatically provides feature selection, and also, in the case of multiple functional (and non-functional) covariates, variable selection. The selection performance and the interpretability of the coefficients are investigated in simulation studies. The cell chip and gas sensor data as well as an established functional data set, the phoneme data introduced by Hastie et al. (1995), are examined. Here, the relevance of the feature selection aspect of the ensemble is illustrated.

The above k-nearest-neighbor ensemble is modeled via minimizing the Brier score, and certain constraints have to be put on the ensemble coefficients to ensure predictions to be on a valid scale. In **Chapter 4**, the functional ensemble is combined with a penalized and constrained multinomial logit model (MLM). It is shown that this synthesis yields a powerful classification tool for functional data (possibly mixed with non-functional predictors), which again provides automatic variable selection. The choice of an appropriate, sparsity-inducing penalty allows to estimate most model coefficients to zero, and permits

class-specific coefficients in multiclass problems, so that feature selection is obtained. An additional constraint within the multinomial logit model ensures that the model coefficients can be considered as weights. Thus the estimation results become interpretable with respect to the discriminative importance of the selected features, which is rated by a feature importance measure. The simulation study of the previous chapter is re-estimated via the penalized and constrained MLM, and results are compared. In two application examples, namely the cell chip and the phoneme data, the interpretability as well as the selection results are examined. The classification performance is compared to various other functional and non-functional classification approaches which are in common use. All findings are compared to those of the previous chapter.

The thesis closes with a discussion on potential aspects for future research concerning the presented methods.

Due to the close interdependence especially of the concerted Chapters 3 and 4, some paragraphs contain a certain degree of overlap with regard to content. These overlaps are consciously retained to enhance comprehensibility and allow for a seperate reading of the single chapters.

This thesis has been published in parts in peer reviewed journals or as pre-prints at the Cornell University Library's open access archive `arXiv.org`. All manuscripts have been written in cooperation with (supervising) coauthors. The manuscripts and the personal contributions of the authors to the respective collaborations are listed below.

Chapter 2 bases on
K. Fuchs, F. Scheipl, and S. Greven (2015b) – Penalized scalar-on-functions regression with interaction term. *Computational Statistics & Data Analysis* 81, 38 – 51.
Sonja Greven initialized the project. Karen Fuchs provided parts of the cell chip data, performed the data cleaning, implemented the interaction effect weights and set up the simulation study in R (R Core Team, 2017). She performed the data analyses in close cooperation with Sonja Greven and Fabian Scheipl. The manuscript was drafted in close collaboration of all coauthors. Chapter 2 contains several sections that are not part of Fuchs et al. (2015b). This includes sections on the influence of preprocessing the data (Sections 2.5.2 and 2.6.2), on identifiability issues (Section 2.7), a section concerned with functional covariate interaction of higher orders (Section 2.8) and some additional paragraphs. Preliminary results linked to the topic of Chapter 2 have been presented at several conferences in the talks

> *K. Fuchs*, F. Scheipl, S. Greven, and E. Stütz (2013) – Penalisierte funktionale Regression mit skalarer Zielgrösse unter Einführung eines Kovariablen -Interaktionsterms als Sensorauswertestrategie. DPG Frühjahrstagung der Deutschen Physikalischen

Gesellschaft e.V., Jena

K. Fuchs, F. Scheipl, S. Greven, and E. Stütz (2012) – Penalized scalar on function regression with interaction term. 5th International Conference of the ERCIM Working Group on Computing & Statistics, Oviedo

and in the poster presentations

K. Fuchs, S. Greven, F. Scheipl, and E. Stütz (2013) – Penalized scalar on function regression with interaction term as sensor signal evaluation technique. DAGStat 2013, Freiburg (DAGStat poster prize)

K. Fuchs, S. Greven, F. Scheipl, and E. Stütz (2013) – A stochastic sensor signal evaluation technique using penalized scalar on function regression with interaction term. 17th European Conference on Analytical Chemistry, Warsaw.

Chapter 3 bases on
K. Fuchs, J. Gertheiss and G. Tutz (2015a) – Nearest neighbor ensembles for functional data with interpretable feature selection. *Chemometrics and Intelligent Laboratory Systems* 146, 186 – 197.
Karen Fuchs and Jan Gertheiss closely collaborated in developing the conceptual framework of the model. Karen Fuchs implemented the method in R (R Core Team, 2017) as well as implemented and conducted the numerical experiments and the data analyses. She also wrote the first version of the paper. Chapter 3 includes various data analyses and explanatory paragraphs beyond the results given in Fuchs et al. (2015a). This includes an additional simulation study (Section 3.3.2) and exemplifying data analyses (Sections 3.5 and 3.6). Preliminary work on Chapter 3 has been presented in the talk

K. Fuchs, J. Gertheiss, G. Tutz, R. Pohle, K. Wiesner, and M. Fleischer (2013) – Functional Nearest Neighbour Ensemble for Discrimination of Different Gas Species Using Metal Oxide Gas Sensors. 17th European Conference on Analytical Chemistry, Warsaw.

Chapter 4 bases on
K. Fuchs, W. Pößnecker and G. Tutz (2016) – Classification of Functional Data with k-Nearest-Neighbor Ensembles by Fitting Constrained Multinomial Logit Models.
`arXiv:1612.04710v2 [stat.ME]`.
All three authors initialized the project. Karen Fuchs provided parts of the cell chip data and performed the data cleaning. The conceptual extension of their previous work as well as the setup of the numerical experiments and the data analyses were conducted by Karen Fuchs and Wolfgang Pößnecker in close collaboration. Karen Fuchs and Wolfgang

Pößnecker prepared the manuscript. Gerhard Tutz added valuable remarks and complemental notes which improved the manuscript. Chapter 4 includes simulation studies and comparative discussions of results beyond the method given in Fuchs et al. (2016).

Chapter 2

Penalized Scalar-on-Functions Regression with Interaction Term

2.1 Introduction to Functional Generalized Linear Models

Functional generalized linear models have their seeds in the classical generalized linear models (GLM), which were introduced by Nelder and Wedderburn (1972), and generalized additive models (GAM) introduced by Hastie and Tibshirani (1986). The term "generalized" refers to the distribution of the response, which is expected to follow an exponential family distribution, thus relaxing the restrictive assumption of Gaussian responses.
Let $g(\cdot)$ denote a known monotonic link function and $\mu_i = E(y_i)$ the expected mean of the response y_i. y_i are assumed to be (conditionally) mutually independent $i = 1, \dots n$ observations and to follow an exponential family distribution. Further, let β_0 denote the intercept term, $\boldsymbol{X}_i$ the ith row of the matrix of explanatory variables and $\boldsymbol{\beta}$ the vector of parameters that has to be estimated. With that, a GLM has the general form

$$g(\mu_i) = \beta_0 + \boldsymbol{X}_i\boldsymbol{\beta}. \tag{2.1}$$

Identifying the response and covariates, and defining the (exponential family) distribution as well as the link function, GLMs are completely specified. In general, their estimation and further inference base on the maximization of the (log-) likelihood. GLMs are a well-studied, widespread class of models, and many modifications and extensions have been developed, for example Bayesian GLMs, GLMs using a quasi-likelihood approach, or GLMs that handle covariates with measurement error.
Hastie and Tibshirani (1986) modified Model (2.1) by incorporating (metric) covariates x_q, $q = 1, \dots Q$, by means of smooth functions $f_q(x_{iq})$,

$$g(\mu_i) = \beta_0 + \sum_{q=1}^{Q} f_q(x_{iq}),$$

yielding a GAM without linear effects. Often, GAMs include additional strictly linear (or "parametric") covariates $\tilde{\boldsymbol{X}}_i$,

$$g(\mu_i) = \beta_0 + \tilde{\boldsymbol{X}}_i\tilde{\boldsymbol{\beta}} + \sum_{q=1}^{Q} f_q(x_{iq}). \tag{2.2}$$

If each function $f_q(x_{iq})$ is expanded in a linear basis expansion $f_q(x_{iq}) = \sum_{k=1}^{K_q} b_{qk}B_{qk}(x_{iq})$ with known basis functions $B_{qk}(x_{iq})$ and unknown coefficients b_{qk}, the nonlinear terms in (2.2) can be summarized in matrix notation $\sum_{k=1}^{K_q} b_{qk}B_{qk}(x_{iq}) = \boldsymbol{B}_q^T\boldsymbol{b}_q$, with vectors $\boldsymbol{B}_q = (B_{qk}(x_{iq}))_{k=1,\dots K_q}$ of basis function evaluations and $\boldsymbol{b}_q = (b_{qk})_{k=1,\dots K_q}$ of unknown coefficients. Now let $\boldsymbol{X}_i = [\tilde{\boldsymbol{X}}_i, \boldsymbol{B}_1^T, \dots \boldsymbol{B}_Q^T]$ denote the concatenated matrix of covariates and basis functions, and $\boldsymbol{\beta} = [\tilde{\boldsymbol{\beta}}, \boldsymbol{b}_1, \dots, \boldsymbol{b}_Q]$ the concatenated vector of coefficients that are to be estimated. Then a GAM can also be written as a GLM of the form

$$g(\mu_i) = \beta_0 + \boldsymbol{X}_i\boldsymbol{\beta}.$$

Due to the smooth functions included in GAMs, the estimation procedure of GLMs has to be adapted to avoid overfitting, and to account for the smooth character of the functions. Thus, penalized maximum (log-) likelihood estimation can, for example, be used in GAM estimation, with appropriately chosen penalty and a smoothing parameter which is estimated from the data.
An important issue in additive models is the uniqueness of the estimated functions $\hat{f}_q(\cdot)$: subtracting a constant offset from, say, $\hat{f}_1(\cdot)$, $\hat{f}_1(\cdot) = f_1(\cdot)-$const., and adding the same offset to, say, $\hat{f}_2(\cdot)$, $\hat{f}_2(\cdot) = f_2(\cdot)+$const., does not change the model outcome. The offset could also be incorporated in the intercept β_0. This identifiability problem is usually circumvented by constraining the functions $\hat{f}_q(\cdot)$ to have zero mean.
As for GLMs, there exists a vast literatur on GAMs and their extensions, including GAMs with multivariate and interaction terms.

The general form (2.2) of GAMs is very similar to the general form of functional generalized linear models (FGLM) for scalar responses. Let y_i again denote a (scalar) (conditionally) mutually independent response following an exponential family distribution. In contrast to above, the covariates $x_{iq}(t)$ now are random functions rather than random variables, defined on a certain domain $t \in \mathbb{D}$. A simple FGLM with scalar response is then given by

$$g(\mu_i) = \beta_0 + \sum_{q=1}^{Q} \int_{\mathbb{D}} x_{iq}(t)f_q(t)dt,$$

where $f_q(t)$ are functional coefficients that have to be estimated. Similar to GAMs, functional coefficients are affected by identifiability issues. In Section 2.7, we will consider aspects of this problem in more detail.

Various estimation approaches, as e.g. penalized likelihood maximization, and extensions of the model, as e.g. a FGLM including a functional response, have been examined. The monograph by Ramsay and Silverman (2005) deals with the most established functional model types and analysing techniques. The review by Morris (2015) compares well-known as well as up-to-date functional regression approaches, while Greven and Scheipl (2017) discuss a framework for functional regression.

For a scalar response and functional covariates, many regression models include only a single functional covariate, such as the non-parametric functional regression models of Burba et al. (2009), Wang et al. (2012) and Kudraszow and Vieu (2013). The work of Ferraty and Vieu (2009) introduces a non-parametric additive model including two or more functional covariates.

The most common parametric model is the generalized functional linear model, for which several methods for estimation have been proposed. One strain of research expands both the functional covariate and the coefficient function in a principal component basis (e.g. Müller and Stadtmüller, 2005; Reiss and Ogden, 2007). Other approaches use a spline basis expansion of the coefficient function or the functional covariate and a smoothness penalty approach (e.g. James, 2002; Wood, 2011; Goldsmith et al., 2011).

We have extended the (generalized) functional linear model by interaction terms, relaxing the common additivity assumption of covariate effects. Other recent work in functional regression has focused on relaxing other assumptions such as the linearity assumption. While it seems feasible to extend many of these models to more than one functional covariate, as for example (semi-) functional partial linear models (see e.g. Aneiros-Perez and Vieu, 2008; Lian, 2011), the smoothed functional inverse regression (see e.g. Ferre and Yao, 2005), functional sufficient dimension reduction (see e.g. Lian and Li, 2014), functional projection pursuit regression (see e.g. Ferraty et al., 2013), or single and multiple index functional regression (see e.g. Chen et al., 2011), it is unclear how interaction effects of multiple functional covariates, as well as inference on interpretable interaction effects, can be implemented in these frameworks.

Although some of the above methods include effects of more than one functional covariate, the estimation of interaction effects between functional covariates does not seem to have received much attention until now. If the assumption of additivity of the effects of multiple functional covariates is questionable, a sensible way to extend the generalized functional linear model is to add covariate interaction effects. This chapter introduces a functional interaction term $\int\int x_{i1}(s)x_{i2}(t)\beta(s,t)dsdt$ of functional covariates $x_{i1}(s)$ and $x_{i2}(t)$ with bivariate parameter function $\beta(s,t)$, extending the model with only main effects in Wood (2011).

Our bivariate parameter function $\beta(s,t)$ for the interaction term is represented in terms of a tensor product spline basis. A similar representation of a bivariate coefficient function can be found in Marx and Eilers (2005) in the context of scalar-on-image regression. Marx and Eilers (2005) also examine a generalized linear model and use a penalized log-likelihood approach for estimation. The main difference lies in the fact that Marx and Eilers (2005)

consider a single image covariate $x_i(s,t)$, while we have two covariates $x_{i1}(s)$ and $x_{i2}(t)$ and consider their main effects as well as their interaction. Our model is also related to Yao and Müller (2010), who consider a pth-order polynomial model, where the scalar mean response depends on two-way up to p-way interaction effects of the centered predictor process with itself. Our approach, on the other hand, allows for interaction effects between different functional covariates. Yao and Müller (2010) expand the functional regression parameters as well as the centered functional covariate in the empirical eigenfunction basis of the functional covariate. By contrast, we do not assume the interaction effect surface to lie in the space spanned by the eigenfunctions of the two covariate processes, but to be smooth, and use penalized splines for estimation. Recently, Usset et al. (2016) presented an identical model and estimation approach, resulting in similar findings in the congruent parts of their simulation study, as the one in Fuchs et al. (2015b), which is related to this chapter. Another recent modeling approach concerning bivariate coefficient functions can be found in Yang et al. (2013), who examine an interaction term similar to ours. Yang et al. (2013) approximate the covariate and coefficient functions by a truncated Karhunen-Loève decomposition. The coefficients of the coefficient functions' decomposition are specified as priors of a mixture of Dirac functions, and estimation is fulfilled via stochastic search variable selection and a joint Bayesian analysis. Bivariate parameter functions can also be found for example in Antoch et al. (2010) or Ivanescu et al. (2015) in the context of function-on-function regression.

Along with the progress in computer sciences, a lot of functional data of different fields of research as well as applications became available. The works of Ramsay and Silverman (2005), Sorensen et al. (2013) and Goldsmith and Scheipl (2014) give an impression thereof, while Ullah and Finch (2013) reviewed a selection of functional analyses.
Our method, although general, is motivated by two data sets. The first contains spectra of fossil fuel samples measured at the ultraviolet-visible (UV-VIS) and near infrared (NIR) range. The main goal here is the prediction of the heat value of a sample based on its spectrum. The second data set consists of cell chip data, where three different and concurrently measured sensor signal types reflect the metabolism of a layer of living cells growing on the chip surface. Especially the prediction of the concentration of probably bioactive substances contained in the cell nutrient medium is of interest.
In Section 2.2, we present our model and the estimation method used. Since later results are compared to those of other functional methods, these methods are introduced briefly in Section 2.3. Section 2.4 presents an extensive simulation study. Our method is applied to the two motivating data sets in Sections 2.5 and 2.6. Identifiability in our context is discussed in Section 2.7. Section 2.8 comments on scalar-on-functions regression models including a three-way interaction term. We close with a short discussion and outlook in Section 2.9.

2.2 Scalar-on-Functions Regression with Interaction Term

We extend the generalized functional linear model to include interactions for functional covariates. We assume the scalar responses y_i, $i = 1, \dots n$, to be (conditionally) mutually independent and to follow an exponential family distribution with a known link function $g(\cdot)$ linking the expected value μ_i of y_i to the linear predictor η_i,

$$\begin{aligned} g(\mu_i) &= \eta_i = \beta_0 + \int x_{i1}(s)\xi_1(s)ds + \\ &\quad \int x_{i2}(t)\xi_2(t)dt + \int\int x_{i1}(s)x_{i2}(t)\beta(s,t)dsdt. \end{aligned} \tag{2.3}$$

Here, β_0 is the intercept term, and $x_{i1}(s)$ and $x_{i2}(t)$ are two functional covariates that are expected to influence y_i. The covariate values $x_{i1}(s)$ are observed without error in the interval $\mathbb{D}$ with discrete observation points $\{s_1, \dots s_J\} \subset \mathbb{D}$. Likewise, $x_{i2}(t)$ is observed without error in the interval $\mathbb{E}$ with discrete observation points $\{t_1, \dots t_K\} \subset \mathbb{E}$. $\xi_1(s)$, $\xi_2(t)$ and $\beta(s,t)$ are unknown functional coefficients corresponding to the main and interaction terms. In the linear case $y_i = \mu_i + \varepsilon_i$, we assume ε_i to be independent and identically distributed (iid) normal errors with zero mean and variance σ^2. Following Wood (2011) in approximating the integrals of Model (2.3) by quadrature sums, the model can be expressed as

$$\begin{aligned} g(\mu_i) &\approx \beta_0 + h_1 \sum_{j=1}^{J} x_{i1}(s_j)\xi_1(s_j) + h_2 \sum_{k=1}^{K} x_{i2}(t_k)\xi_2(t_k) + \\ &\quad h_1 h_2 \sum_{j=1}^{J}\sum_{k=1}^{K} x_{i1}(s_j)x_{i2}(t_k)\beta(s_j,t_k), \end{aligned}$$

with h_1, h_2 being the lengths of the intervals between two observation points in $\mathbb{D}$ and $\mathbb{E}$, respectively, assuming a regular grid of observations on both intervals. In case of unequal spacing, the sums could be replaced by appropriate weighted sums from quadrature rules. Both main effects of Model (2.3) can be expanded in a spline basis (Wood, 2011), and the interaction term can be represented in a tensor product basis of two univariate spline bases,

$$\begin{aligned} g(\mu_i) &\approx \beta_0 + h_1 \sum_{j=1}^{J}\sum_{f=1}^{F} x_{i1}(s_j)b_{1f}\phi_{1f}(s_j) + h_2 \sum_{k=1}^{K}\sum_{g=1}^{G} x_{i2}(t_k)b_{2g}\phi_{2g}(t_k) \\ &\quad + h_1 h_2 \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{l=1}^{L}\sum_{m=1}^{M} x_{i1}(s_j)x_{i2}(t_k)c_{lm}\phi_{3l}(s_j)\phi_{4m}(t_k) \\ &= \beta_0 + h_1 \boldsymbol{x}_{i1}^T \boldsymbol{\Phi}_1 \boldsymbol{b}_1 + h_2 \boldsymbol{x}_{i2}^T \boldsymbol{\Phi}_2 \boldsymbol{b}_2 + \\ &\quad h_1 h_2 (\boldsymbol{x}_{i2}^T \otimes \boldsymbol{x}_{i1}^T)\,(\boldsymbol{\Phi}_4 \otimes \boldsymbol{\Phi}_3)\,\mathrm{vec}(\boldsymbol{C}). \end{aligned} \tag{2.4}$$

Here, $\xi_1(s)$ and $\xi_2(t)$ are approximated by basis expansions $\xi_1(s) \approx \sum_{f=1}^{F} b_{1f}\phi_{1f}(s)$ and $\xi_2(t) \approx \sum_{g=1}^{G} b_{2g}\phi_{2g}(t)$ with coefficients $\boldsymbol{b}_1 = (b_{11}, \dots b_{1F})^T$ and $\boldsymbol{b}_2 = (b_{21}, \dots b_{2G})^T$ and suitable basis functions $\phi_{1f}(s)$ and $\phi_{2g}(t)$. $\beta(s,t)$ is approximated using a tensor product of basis functions $\sum_{l=1}^{L}\sum_{m=1}^{M} c_{lm}\phi_{3l}(s)\phi_{4m}(t)$ with coefficients $\boldsymbol{C} = (c_{lm})_{l=1,\dots,L;m=1,\dots,M}$. The vectors $\boldsymbol{x}_{i1} = (x_{i1}(s_1), \dots x_{i1}(s_J))^T$ and $\boldsymbol{x}_{i2} = (x_{i2}(t_1), \dots x_{i2}(t_K))^T$ contain the observed covariate values and $\boldsymbol{\Phi}_1 = (\phi_{1f}(s_j))_{j=1,\dots,J;f=1,\dots,F}$, $\boldsymbol{\Phi}_2$, $\boldsymbol{\Phi}_3$, $\boldsymbol{\Phi}_4$ (set up analogously) are matrices of basis function evaluations. The symbol $\otimes$ denotes the Kronecker product, the operation vec($\boldsymbol{C}$) converts its argument into a column vector $\boldsymbol{c} = (c_{11}, \dots\ c_{L1}, \dots\ c_{1M}, \dots, c_{LM})^T$.
Quadratic roughness penalties are added to the log-likelihood of Equation (2.4). The penalized log-likelihood function $l_p(\boldsymbol{\theta})$ for given smoothing parameters λ_a, $a = 1, \dots 4$, is

$$l_p(\boldsymbol{\theta}) = l(\boldsymbol{\theta}) - \frac{1}{2}\boldsymbol{\theta}^T\boldsymbol{\mathcal{S}}\boldsymbol{\theta}. \tag{2.5}$$

Here, $\boldsymbol{\theta} = \left(\beta_0, \boldsymbol{b}_1^T, \boldsymbol{b}_2^T, \text{vec}(\boldsymbol{C})^T\right)^T$ is the coefficient parameter vector. Function $l(\boldsymbol{\theta}) = \sum_{i=1}^{n}\Big(\left(y_i\Delta_i - b(\Delta_i)\right)/a(\nu)\ + c(\nu, y_i)\Big)$ is the ordinary log-likelihood of Equation (2.4), where y_i is expected to follow an exponential family distribution, with $\Delta_i = \mu_i$, $b(\Delta_i) = \frac{\Delta_i^2}{2}$, $\nu = \sigma^2$, $a(\nu) = \nu$, and $c(\nu, y_i) = -\frac{y_i^2 + \nu\log(2\pi\nu)}{2\nu}$ for a normally distributed and $\Delta_i = \log\left(\frac{\mu_i}{1-\mu_i}\right)$, $b(\Delta_i) = \log\left(1 + e^{\Delta_i}\right)$, $\nu = 1$, $a(\nu) = \nu$, and $c(\nu, y_i) = 0$ for a Bernoulli distributed response. $\boldsymbol{\mathcal{S}}$ is a block-diagonal matrix of four blocks 0, $\lambda_1\boldsymbol{\mathcal{P}}_1$, $\lambda_2\boldsymbol{\mathcal{P}}_2$, $\lambda_3\boldsymbol{\mathcal{P}}_3$ + $\lambda_4\boldsymbol{\mathcal{P}}_4$ with known and fixed (semi-) positive definite penalty matrices $\boldsymbol{\mathcal{P}}_a, a = 1, \dots, 4$. These penalty matrices $\boldsymbol{\mathcal{P}}_a$ penalize each of the coefficient vectors $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ and both directions of the surface coefficients $\boldsymbol{C}$ separately. The penalty matrices have to be chosen appropriately, for example, when using B-Splines, as first order difference matrices for penalizing $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$. For the penalization of $\boldsymbol{C}$, Kronecker products $\boldsymbol{\mathcal{P}}_3 = \boldsymbol{I}_M \otimes \tilde{\boldsymbol{\mathcal{P}}}_1$ and $\boldsymbol{\mathcal{P}}_4 = \tilde{\boldsymbol{\mathcal{P}}}_2 \otimes \boldsymbol{I}_L$ of an appropriate penalty matrix for each direction and a suitable identity matrix can be used (cf. Marx and Eilers, 2005). Penalizing the two directions of $\boldsymbol{C}$ separately allows for anisotropy in the coefficient surface. If this is not necessary, a simpler isotropic penalty matrix with a single smoothing parameter can be used. The implementation we present allows for most common choices of spline bases and penalties. In practice, the estimation of $\boldsymbol{\theta}$ is performed conditional on estimated smoothing parameters λ_a via a penalized iteratively re-weighted least squares scheme.
Since the smoothing parameters λ_a heavily affect the estimation of the functional parameters, care has to be taken concerning their choice. One possibility is to minimize the generalized crossvalidation (GCV) score, another is to use restricted maximum likelihood estimation (REML) within a mixed model framework (Wood, 2006; Ruppert et al., 2003). REML may sometimes be preferable to GCV for smoothing parameter selection (Wood, 2011; Reiss and Ogden, 2009) in the sense of reducing the small number of undersmoothing failures and reducing the occurences of multiple optima of the criterion. On the

other hand, GCV is computationally more efficient. In our simulations, the quality of both the estimates and the relative mean squared errors of prediction were comparable for these two methods, but GCV was faster to compute. We thus recommend GCV where computational time is an important issue, as is the case in e.g. online-monitoring for which the applications in Sections 2.5 and 2.6 are ultimately intended. In other cases, where time is of less importance, we recommend to follow the findings in Reiss and Ogden (2009) and Wood (2011) in using REML or in comparing the two criteria. The number of knots, while it could be chosen data-driven by either GCV (Ruppert, 2002) or maximum likelihood (Kauermann and Opsomer, 2011), is of less importance than the smoothing parameter, as long as it is chosen large enough to capture the main characteristics of the data. In practice, we recommend to conduct a sensitivity analysis regarding this choice. For example, in our applications we compared predictive capability of the model across a range of knot numbers and chose the smallest number yielding a low mean squared error of prediction. Pointwise 95% confidence bands can be based on the estimated coefficient functions $\pm$ two times their standard error estimates. For example, the estimate $\hat{\xi}_1(s_j) = \sum_{f=1}^{F} \hat{b}_{1f}\phi_{1f}(s_j)$ at s_j yields a standard error estimate $\hat{\text{sd}}\left(\hat{\xi}_1(s_j)\right) = \sqrt{\boldsymbol{\Phi}_1(s_j)\hat{\boldsymbol{\Sigma}}_{\xi,1}\boldsymbol{\Phi}_1^T(s_j)}$ at s_j, with $\hat{\boldsymbol{\Sigma}}_{\xi,1}$ being the Bayesian posterior covariance matrix of the estimated $\hat{\boldsymbol{b}}_1$ (see e.g. Ruppert et al., 2003), and $\boldsymbol{\Phi}_1(s_j) = (\phi_{11}(s_j), \ldots, \phi_{1F}(s_j))^T$. We can use these pointwise standard errors for all $s_j \in \{s_1, \ldots s_J\} \subset \mathbb{D}$ to construct pointwise confidence bands, and in an analogous way for coefficient estimate $\hat{\xi}_2(t)$. For the bivariate surface estimate, pointwise confidence bands are computed as the estimated surface $\pm$ two times its standard error estimates $\hat{\text{sd}}\left(\hat{\beta}(s_j, t_k)\right) = \sqrt{(\boldsymbol{\Phi}_4(t_k) \otimes \boldsymbol{\Phi}_3(s_j))\,\hat{\boldsymbol{\Sigma}}_\beta\left(\boldsymbol{\Phi}_4^T(t_k) \otimes \boldsymbol{\Phi}_3^T(s_j)\right)}$, for all $s_j \in \{s_1, \ldots s_J\} \subset \mathbb{D}$, $t_k \in \{t_1, \ldots t_K\} \subset \mathbb{E}$, $\hat{\boldsymbol{\Sigma}}_\beta$ being the Bayesian posterior covariance matrix of $\hat{\boldsymbol{C}}$, and $\boldsymbol{\Phi}_3(s_j)$ and $\boldsymbol{\Phi}_4(t_k)$ being defined analogously to $\boldsymbol{\Phi}_1(s_j)$.

2.2.1 Possible Extensions

Our scalar-on-functions regression model (2.3) is not limited to one simple interaction. It can be extended by adding further main effects, random effects and strictly parametric as well as smooth effects of scalar covariates, see Wood (2004). Further two-way or higher order interaction effects can be added to the model analogously to the interaction effect introduced above. For implementation, we can make use of the robust, highly versatile and well-developed inference algorithms implemented in `mgcv` (Wood, 2013) in `R` (R Core Team, 2017).

2.2.2 Implementation

For maximization of the penalized log-likelihood in Equation (2.5) we use the `gam`-function of the `mgcv` package, which is tailored towards penalized regression problems with splines. In the following sections we use cubic P-splines with a first order difference penalty, pe-

nalizing deviations from constant coefficient functions and surfaces, which can be achieved using the function call

```
model <- gam(y ~ 1 +
                 s(s, by = h1*x1, bs = "ps", m = c(2,1)) +
                 s(t, by = h2*x2, bs = "ps", m = c(2,1)) +
                 te(s, t, by = h1*h2*weights, bs = "ps",
                       m = list(c(2,1),c(2,1))))
```

with response vector `y`. The first term `1` accounts for the intercept, the first main effect $h_1 \sum_{j=1}^{J} x_{i1}(s_j)\xi_1(s_j)$, $i = 1, \dots n$, is called by the expression `s(s, by = h1*x1, bs = "ps", m = c(2,1))`. Here, `s` $= (s_1, \dots s_J)^T$ is the vector of grid points where covariate $x_{i1}(s)$ is observed. The matrix of covariate values $\boldsymbol{x}_1 \in \mathbb{R}^{n \times J}$, stored in `x1`, is treated as a multiplicand to `s` and is itself multiplied with the interval constant `h1` $= h_1$ of the Riemann sum. The term `bs = "ps"` chooses P-splines as penalized splines. The first number in `m = c(2,1)` gives the P-splines' order, the second gives the order of the difference penalty. The second main effect is called analogously. For the interaction effect, the multiplicand `weights` is a matrix of size $n \times JK$ consisting of pairwise products of the covariate values. The ith row in `weights` is a vector equal to the Kronecker product of the two covariate vectors for the ith observation.
We choose to estimate the smoothing parameters λ_a by minimizing the GCV score (method `"GCV.Cp"` in the `mgcv` package). As mentioned, other selection criteria, such as REML, could also be used. We provide R-code implementing our approach in the Web-supplement of Fuchs et al. (2015b).

2.3 Alternative Scalar-on-Functions Regression Methods

Additionally to the proposed method, we tested the performance of other estimation methods on a simulation setting example as well as on the spectra data. These methods are able to include one or more functional covariates, but can not deal with functional covariate interaction. The methods we compare our results with are first a functional linear model of an analogous form $y_i = \beta_0 + \frac{1}{|T_1|}\int_{T_1} x_{i1}(s)\xi_1(s)ds + \frac{1}{|T_2|}\int_{T_2} x_{i2}(t)\xi_2(t)dt + \varepsilon_i$ (Ramsay and Silverman, 2005), calculated with the `fregre.glm` function. Here, y_i, $i = 1, \dots n$, is a scalar response, β_0 is the intercept, $x_{i1}(s)$ and $x_{i2}(t)$ are two functional covariates, T_1, T_2 are the supports of the respective functions, ε_i is an iid normal error and $\xi_1(s)$, $\xi_2(t)$ are unknown coefficient functions. They are estimated via likelihood maximization.
Second, we calculate a functional spectral additive model
$y_i = \beta_0 + \sum_{q=1}^{Q} \left(f_{1q}(\iota_{1q}) + f_{2q}(\iota_{2q}) \right) + \varepsilon_i$, an extension of Müller and Stadtmüller (2005),

with the `fregre.gsam` function. Here, y_i, $i = 1, \dots n$, is a scalar response, β_0 is the intercept, ε_i is an iid normal error, and f_{1q}, f_{2q}, $q = 1, \dots Q$, are unknown smooth functions which have to be estimated. The ι_{1q}, ι_{2q}, $q = 1, \dots Q$, are the coefficients of basis function expansions of the covariates $x_{i1}(s) \approx \sum_{q=1}^{Q} \iota_{1q}\phi_{1q}(s)$ and $x_{i2}(t) \approx \sum_{q=1}^{Q} \iota_{2q}\phi_{2q}(t)$, with $\phi_{1q}(s)$, $\phi_{2q}(t)$, $q = 1, \dots Q$, being known bases functions, e.g. splines or eigenfunctions. Estimation of f_{1q} and f_{2q} is performed via penalized likelihood maximization.
Third, the nonparametric functional generalized kernel additive model
$\hat{y}_i = g^{-1}\Big(\hat{f}_1\left(x_{i1}(s)\right) + \hat{f}_2\left(x_{i2}(t)\right)\Big)$ (Febrero-Bande and Gonzalez-Manteiga, 2013) is calculated with the `fregre.gkam` function. Here, $\hat{y}_i$, $i = 1, \dots n$, is an estimated scalar response, $g^{-1}(\cdot)$ is the inverse of a known link function, $x_{i1}(s)$ and $x_{i2}(t)$ are two functional covariates and f_1, f_2 are unknown smooth functions which have to be estimated. The estimation of them is done via a backfitting algorithm. In each step, they are fitted nonparametrically using an iterative local scoring algorithm by applying Nadaraya-Watson weighted kernel smoothers, with a semi-metric in the latter's argument.
The last alternative method is a penalized functional model of the form
$y_i = \beta_0 + \int_{T_1} x_{i1}(s)\xi_1(s)ds + \int_{T_2} x_{i2}(t)\xi_2(t)dt + \varepsilon_i$ (Goldsmith et al., 2011), calculated via the `pfr` function. Here, y_i, $i = 1, \dots n$, is a scalar response, β_0 is the intercept, $x_{i1}(s)$ and $x_{i2}(t)$ are two functional covariates defined on domains of definition T_1, T_2, ε_i is an iid normal error and $\xi_1(s)$, $\xi_2(t)$ are unknown coefficient functions which have to be estimated. Estimation is based on penalized spline regression, after expanding the functional covariates in a large number of smooth eigenvectors.
The first three functions are available in the `fda.usc` package (Febrero-Bande et al., 2013) for R, the last is implemented in the `refund` package (Crainiceanu et al., 2013). As far as possible with the respective implementations, parameters as for example the number of basis functions used for the coefficients' spline bases are chosen identical to our approach. Otherwise, the default settings of the respective function calls were used.

2.4 Simulation Study

We evaluate the performance of our approach in an extensive simulation study. Its setup and main results are discussed in the following.

2.4.1 Simulation Study Setup

Three different generating processes are used for the two functional covariates $x_1(s)$ and $x_2(t)$. The first is taken from Wood (2013) and corresponds to a sum of up to five normal densities $x_1(s) = \sum_{w=1}^{W} f_w(s; \mu_w, \sigma_w^2)$, where variances σ_w^2 and means μ_w are drawn from uniform distributions, with $W = \llbracket \kappa \rrbracket \in \mathbb{Z}$, $\kappa \sim U(0, 5)$, $\mu_w \sim U(s_1, s_J)$, $\sigma_w^2 = \frac{s_J - s_1}{10} u_w$, $u_w \sim U(0.5, 1.5)$. $U(\tau_1, \tau_2)$ indicates a uniform distribution on the interval $[\tau_1, \tau_2]$. The second generating process is a linear combination of a constant, a linear function and

sine-functions, $x_1(s) = \nu_1 + \nu_2 s + \sum_{z=3}^{18} \nu_z \sin\left(\frac{\pi}{2}(2z-3)s\right)$, with $\nu_1, \nu_2 \sim N(0,1)$ and $\nu_z \sim N\left(0, \left(\frac{\pi(2z-3)}{2}\right)^{-1}\right)$, $z = 3, \dots 18$. Process three is a linear combination of B-spline functions, $x_1(s) = \sum_{v=1}^{V} \omega_v b_v(s)$ with $\omega_v \sim N(0,1)$ and $b_v(s)$ being B-splines of degree three. $x_2(t)$ is simulated analogously.
For the true parameter functions, two main effect functions $\alpha(s)$ and $\gamma(t)$ and an interaction function $\rho(s,t)$,

$$\begin{aligned}
&\alpha(s) = 0.02s^{11}(10(1-s))^6 + \frac{1}{10}(10s)^3(1-s)^{10},\\
&\gamma(t) = 5t - 0.0001\exp(t) + 10\sin(t), \text{ and}\\
&\rho(s,t) = \sin\left(\frac{st}{20}\right)
\end{aligned}$$

are chosen, where $\alpha(s)$ is also taken from Wood (2013). $J = 100$ equidistant grid points in $\mathbb{D} = [0,1]$ and $\xi_1(s) = \alpha(s)$ are kept unchanged. The conditional distribution of the responses y_i is taken to be either normal or binomial. We also vary which effects are included in the model as well as the number of observations n, the generating processes for $x_1(s)$ and $x_2(t)$ and the number of B-spline basis functions V where applicable. Additionally, the number of grid points K for covariate $x_2(t)$ and the true functions $\xi_2(t)$ and $\beta(s,t)$ are varied. All possible combinations of the parameter choices listed in Table 2.1 are considered.
In the logistic model, the covariates and true functions are rescaled in order to simulate probabilities π_i over the whole range of $[0,1]$.

We simulate $R = 200$ data sets and obtain estimates $\hat{\xi}_1(s)$, $\hat{\xi}_2(t)$ and/ or $\hat{\beta}(s,t)$ for each setting. The relative mean squared error of estimation for the interaction is computed as

$$\text{relMSE}_\beta = \frac{1}{R}\sum_{r=1}^{R} \frac{\sum_{j=1}^{J}\sum_{k=1}^{K}\left(\beta(s_j,t_k) - \hat{\beta}_r(s_j,t_k)\right)^2}{\sum_{j=1}^{J}\sum_{k=1}^{K}\left(\beta(s_j,t_k) - \bar{\beta}\right)^2},$$

with $\bar{\beta} = \frac{1}{JK}\sum_{j=1}^{J}\sum_{k=1}^{K}\beta(s_j,t_k)$ and $\bar{\beta} = 0$ when the true coefficient surface is constant. The index $r = 1, \dots R$ represents the replicates and $\hat{\beta}_r(s_j,t_k)$ the corresponding estimates. relMSE_{ξ_1} and relMSE_{ξ_2} are defined analogously.
The relative mean squared error of the response is computed as

$$\text{relMSE}_y = \frac{1}{R}\sum_{r=1}^{R} \frac{\sum_{i=1}^{n}\left(E(y_i) - \hat{y}_i\right)_r^2}{\sum_{i=1}^{n}\left(E(y_i) - \bar{y}\right)_r^2},$$

with $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$, $\hat{y}_i$ being the estimate for y_i and r indexing the replicate.
The advantage of considering a relative MSE lies in the comparability of values irrespective of e.g. differing scales.

I) model specified as	a) linear, $y_i \sim N(\mu_i, 1)$ normally distributed with $\mu_i = \eta_i$ b) logistic, $y_i \sim B(1, \pi_i)$ Bernoulli distributed with $\pi_i = \frac{\exp(\eta_i)}{1+\exp(\eta_i)}$
II) effects included in the model	a) $\eta_i = \beta_0 + \int x_{i1}(s)\xi_1(s)ds + \int x_{i2}(t)\xi_2(t)dt$ b) $\eta_i = \beta_0 + \int\int x_{i1}(s)x_{i2}(t)\beta(s,t)dsdt$ c) $\eta_i = \beta_0 + \int x_{i1}(s)\xi_1(s)ds + \int x_{i2}(t)\xi_2(t)dt + \int\int x_{i1}(s)x_{i2}(t)\beta(s,t)dsdt$
III) sample size	a) $n = 50$ b) $n = 250$ in the case of Bernoulli distributed response: c) $n = 500$ d) $n = 1000$
IV) combinations of the three generating processes normal densities (N), sine-functions (S) and B-splines (B) (first column for $x_1(s)$, second for $x_2(t)$)	a) N - S b) N - B c) S - B d) B - B in the case of using B: i) $V = \max(4, \lfloor\!\lfloor \frac{1}{10}K \rfloor\!\rfloor)$ ii) $V = \lfloor\!\lfloor \frac{3}{20}K \rfloor\!\rfloor$ iii) $V = \lfloor\!\lfloor \frac{1}{2}K \rfloor\!\rfloor$
V) number of grid points K for covariate $x_2(t)$	on the domain $\mathbb{E} = [0, 1]$ we use a) $K = J = 100$ equidistant grid points b) $K = 30$ equidistant grid points
VI) the true parameter function for the second main effect and the interaction term	a) $\xi_2(t) = \alpha(t)$ b) $\xi_2(t) = 5$ constant c) $\xi_2(t) = \gamma(t)$ A) $\beta(s,t) = 5$ constant B) $\beta(s,t) = \rho(s,t)$

Table 2.1: All possible combinations of the different choices are considered, resulting in 1320 different settings.

2.4.2 Results – Linear Model

Figure 2.1 shows results for the linear model with all terms, IIc). As an example, the simulation parameters here were chosen to be $n = 50$ observations, processes with linear combinations of sine-functions for generating $x_1(s)$ and a B-spline basis of $V = 50$ basis functions for $x_2(t)$, $K = J = 100$ equidistant grid points for both covariates, and the true parameter functions $\xi_2(t) = \alpha(t)$ and $\beta(s,t) = \rho(s,t)$. Results for the models with main effects or interaction effect only were similar, but slightly better due to the smaller number of parameters, see also the online appendix of Fuchs et al. (2015b).

In Figure 2.1, the true, univariate parameter functions are depicted as black lines, the $R = 200$ estimates as gray lines, their means as red lines, and the 2.5% and 97.5% pointwise quantiles as dashed blue lines. For clarity, the bivariate parameter function of the interaction term is depicted as a surface plot, with color coding as before. The chosen setting yields some of the worst results. In settings with other parameter choices, the estimates are even closer to their respective true functions and the relMSE values are up to a factor 10^3 smaller than for this setting (cf. Figure 2.2). Means are very close to the respective true functions. Variability around the mean is smaller for $\xi_2(t)$ than for $\xi_1(s)$ due to the larger information content in the covariate $x_2(t)$, please see our discussion below. The relative mean squared errors of estimation relMSE_{ξ_1} and relMSE_{ξ_2} of orders 10^{-1} to 10^{-3} are reasonably small compared with the sample size. Estimation is even better for the interaction term as measured by relMSE_β, with values of order 10^{-2} and less.
The boxplots of these relative mean squared errors of estimation for the chosen setting can be found in Figure 2.2 as the third box per panel. The relMSEs compare these results with the remaining three generating processes IVa) (first boxes), IVb) iii) (second boxes) and IVd) iii) (fourth boxes). The comparison shows that both the relMSE_{ξ_1} and relMSE_{ξ_2} are slightly higher when the covariates are generated by sine-functions. For relMSE_β, the highest values occur when a B-spline basis is used for both covariates. This is consistent with the following general results.
Comparing across parameter settings, the relMSE values decrease with an increasing number of observations n, as expected. relMSE_{ξ_1} and relMSE_{ξ_2} are higher for a small number of $K = 30$ grid points for $x_2(t)$ than for $K = 100$. For $K = 30$, $x_2(t)$ provides less information for the estimation of $\xi_2(t)$ and $\beta(s,t)$, and some estimates miss characteristic features of the true functions. relMSE_{ξ_2} values are up to a factor thousand higher for $\xi_2(t) = \alpha(t)$ than for $\xi_2(t) \equiv 5$ and $\xi_2(t) = \gamma(t)$ due to the higher complexity of $\alpha(t)$ compared to linear or near-linear functions. Even for $\xi_2(t) = \alpha(t)$ values are acceptable and relMSE_{ξ_2} is of order 10^{-1} for settings with $K = 30$ grid points down to 10^{-3} for $K = 100$ grid points. Constant true surfaces [VI A), relMSE_β values of order 10^{-7}] are easier to estimate than non-constant ones [VI B), relMSE_β between 10^{-2} for $n = 50$ down to 10^{-6} for other scenarios]. Identical generating processes for $x_1(s)$ and $x_2(t)$ yield about the same quality of the respective estimates of $\xi_1(s)$ and $\xi_2(t)$. Generating processes with sine-functions or

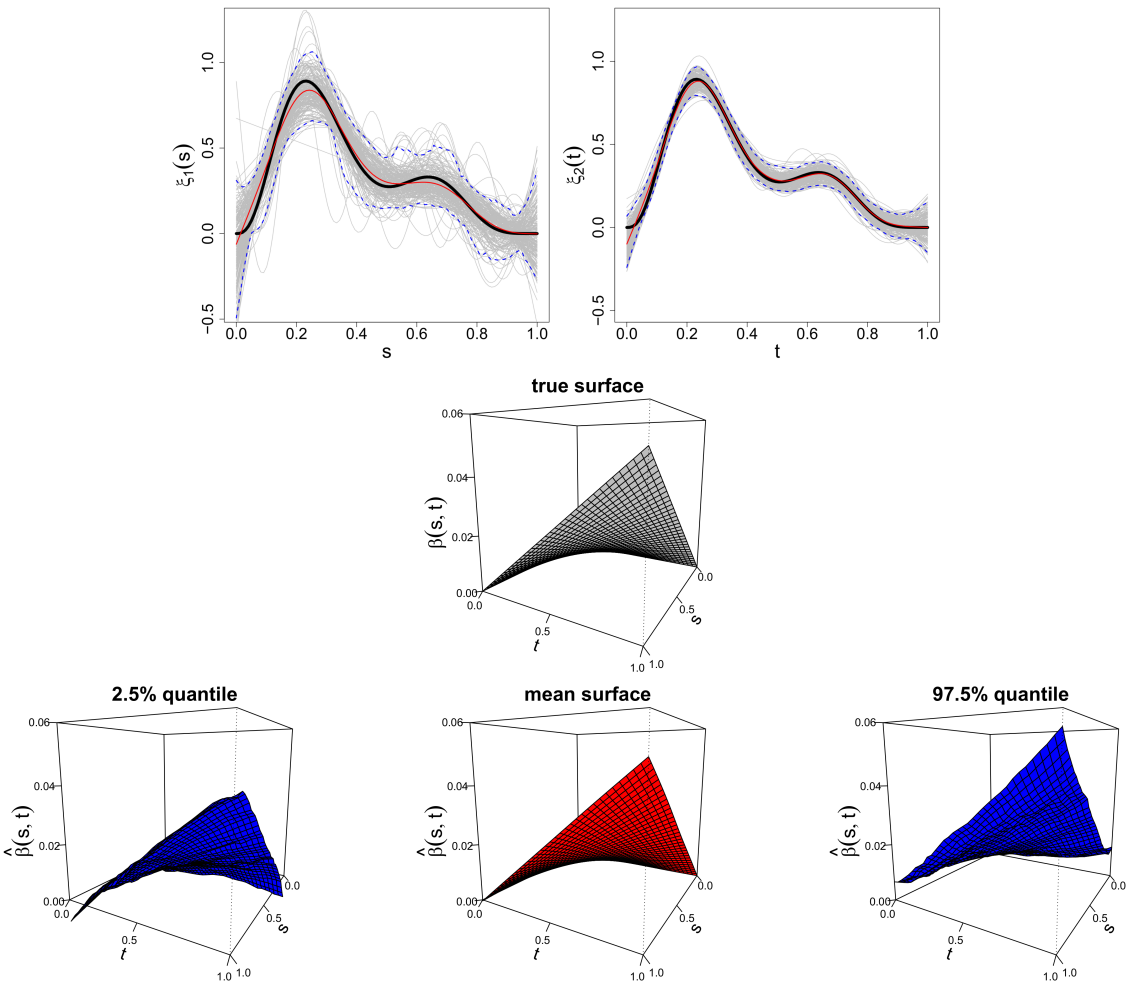


Figure 2.1: Example result for a linear model containing all three effects, $n = 50$ observations, sine- and B-spline generating processes with $K = 100$ grid points for $x_{i1}(s)$ and $x_{i2}(t)$, equal main effect functions and the true interaction function $\beta(s,t) = \rho(s,t)$. The true parameter functions (black), the mean (red) of 200 estimates (gray, for the main effects), and the 2.5% and 97.5% quantiles (dashed blue) are shown. The interaction effect is depicted as a surface plot.

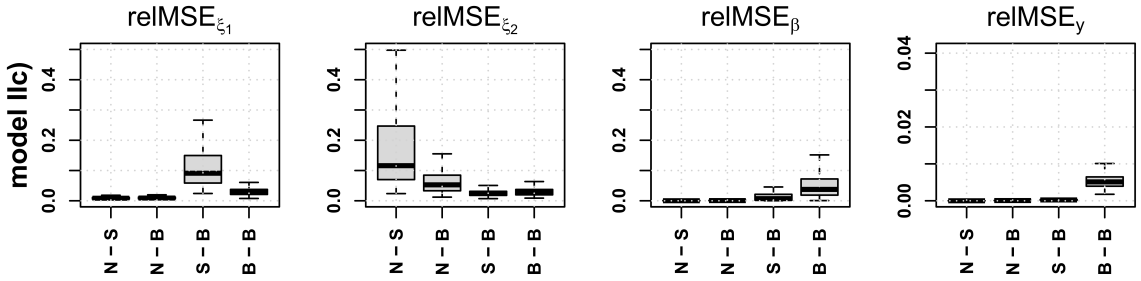


Figure 2.2: Example results for the full model, showing boxplots of relative mean squared errors across 200 simulations for $\xi_1(s)$, $\xi_2(t)$, $\beta(s,t)$ and y, respectively. In each panel, the first box corresponds to results from normal densities/ sine generating functions (38, 16, 149), the second box to normal densities/ B-spline generating functions (38, 12, 133), the third box to sine/ B-spline (16, 12, 50) and the fourth box to B-spline/ B-spline (12, 12, 49) generating functions. The numbers in the brackets give the medians of the ranks of the resulting covariate and weight matrices across all settings. The remaining parameter choices are a small number of $n = 50$ observations, $K = 100$ grid points, $V = 50$ B-spline generating functions where applicable, equal main effect functions and the interaction effect $\beta(s,t) = \rho(s,t)$.

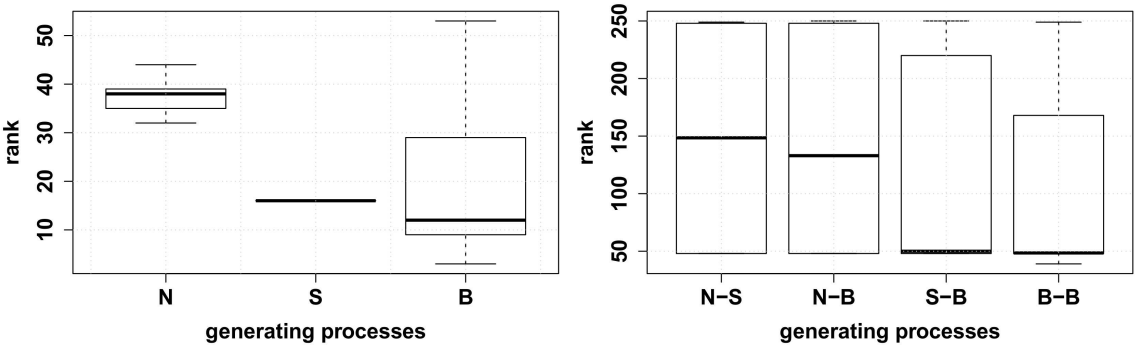


Figure 2.3: Left: Boxplots of the observed ranks of the covariate vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ for the generating processes described in Section 2.4.1. Right: Ranks for the matrix $\boldsymbol{x}_1\boldsymbol{x}_2$ of the interaction term. Both plots show ranks across all settings.

B-splines show higher relMSEs for $\xi_1(s)$ and $\xi_2(t)$, with the sine generating process tending to higher overall relMSE, but not above 10^{-1}. The relative mean squared errors for $\beta(s,t)$, being of order 10^{-2} or less, also show lower values if the generating processes include normal densities, and slightly higher relative errors if sine-functions are used. The process with B-splines for both directions shows the highest relMSE_β. These results can be understood when examining the rank of the covariate matrices, with boxplots of these ranks for main effects and interaction given in Figure 2.3. For illustration, example realizations of covariate $x_1(s)$ for all three generating processes can be found in the online appendix, Section 1, of Fuchs et al. (2015b).
Since the normal density generating process draws both the mean and standard deviation of the normal density randomly, it shows high information contents and high ranks. B-spline covariates can be of low rank, which increases with the number of B-spline functions. The rank for sine-function covariates is constant and equal to 16. Thus, estimates of the main effects are best for normal density processes, followed by covariates generated by a rich B-spline basis. The interaction surface is most reliably estimated for N-S and N-B generating process combinations because of their relatively few low rank covariate matrices.
The overall quality of the estimates is very good in most cases. Rare exceptions from this general pattern occur for all three model variants IIa-c) when low-rank covariates occur. The interaction surface is well estimated across scenarios, independent from other terms included in the model.

The quality of prediction is good in all cases with relMSE_y values ranging between 10^{-2} for scenarios with low-rank covariate processes down to 10^{-11} for settings with larger information content. For increasing n and number of grid points K the relMSE_y decreases slightly. The true parameter functions and the number of terms in the model have no noteworthy influence on the relMSE_y. Thus, prediction, which is often of interest in real world applications, is reliable for all settings.

For the simulation study setting used in Figure 2.1, we compared results for our approach to four other regression approaches (cf. Section 2.3) with scalar responses and two functional covariates. Namely, we use the functional linear model based on Ramsay and Silverman (2005), the functional spectral additive model, which is a nonparametric extension of Müller and Stadtmüller (2005), the nonparametric functional generalized kernel additive model based on Febrero-Bande and Gonzalez-Manteiga (2013) and the penalized functional model based on Goldsmith et al. (2011), for which implementations were available (cf. `R`-packages `fda.usc` (Febrero-Bande et al., 2013) and `refund` (Crainiceanu et al., 2013)). These four methods were chosen as they are to our knowledge the only methods where implementations are available that can deal with more than one functional covariate. However, none of them considers functional interactions.
Figure 2.4 shows the comparison of the results of a simulated data set where parameter

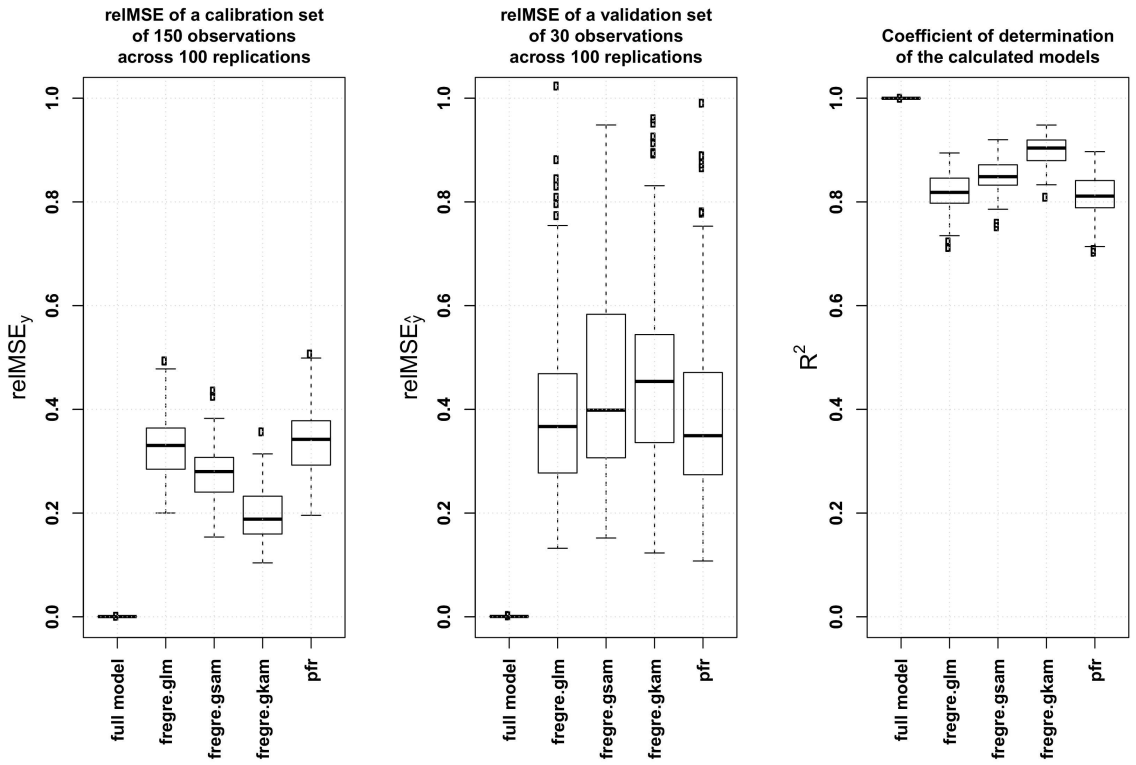


Figure 2.4: Results for a data set of a Gaussian distributed scalar response, based on two main effects and a covariate interaction. The setting here includes a medium number of $n = 150$ observations for the calibration data set and a validation data set of size $n_{val} = 30$. The covariates are generated by the sine-function and B-spline generating processes, with $V = 50$ generating functions for the B-splines, $K = 100$ grid points for $x_{i2}(t)$ and true functions $\xi_1(s) = \alpha(s)$, $\xi_2(t) = \alpha(t)$ and $\beta(s,t) = \rho(s,t)$. The first panel shows the boxplots of the relMSE across 100 replications of the model. The second panel shows the same for the relMSE of prediction. The last panel shows the R^2 of the models.

choices are the same as in Figure 2.1 of Section 2.4.2. We have a Gaussian distributed scalar response, based on two main effects and a covariate interaction. The models were calculated on the basis of a calibration data set of 150 observations. The boxplots of the relative mean squared errors across 100 replicates can be seen in the first panel. The second panel shows the boxplots of the relative mean squared errors of prediction (cf. Section 2.5.1) across 100 replicates of a validation data set of size 30. Additionally, the boxplots of the R^2 across 100 replicates of the models can be seen in the third panel.
The results show clearly that, if the data contains information from a functional covariate interaction, our approach including interaction is the best suited of all implemented methods applicable to more than one functional covariate and a scalar response.

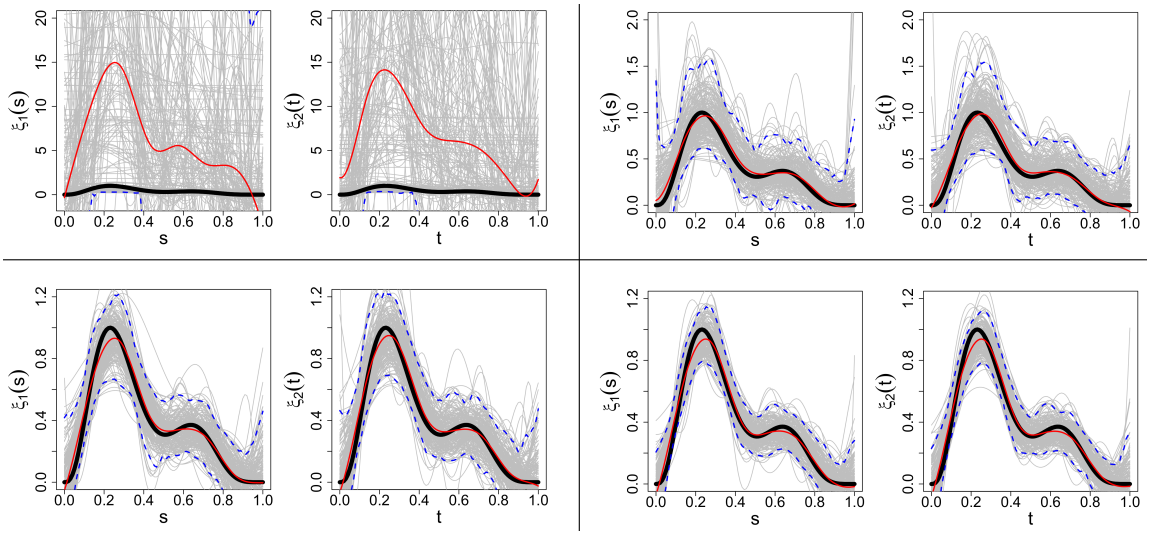


Figure 2.5: Example results for the logistic model with main effects only, $K = 100$ grid points for $x_{i2}(t)$, B-spline generating processes of $V = 15$ bases functions and equal true coefficient functions. Results are shown row-wise for $n = 50$, 250, 500 and 1000 observations. The true parameter functions (black), the mean functions (red) and the 2.5% and 97.5% quantiles (dashed blue) of 200 estimates (gray) are shown.

2.4.3 Results – Logistic Model

For a model with Bernoulli distributed response, the information content for the estimation of the functional parameters is naturally less than for a normally distributed response. Our simulations show that the number of observations required for a good estimation increases substantially. Figure 2.5 shows a typical example for the main effects model. With $n = 50$ observations, there is too little information in the data and most estimates fail to capture the essential features of the true functions. With higher n estimation improves. While estimates seem unbiased for $n \geq 500$, variability decreases slowly and seems to be acceptable for $n \geq 1000$ observations. The interaction-only and full model show similar results.

To give an overview of the effect on estimation when changing single parameters, some more chosen scenarios will be found in the following. All results here reflect typical results for the respective parameter choice for very many observations $n = 1000$.
In Figure 2.6(a), the setting includes B-spline generating processes of $V = 50$ generating functions, $K = 100$ grid points for $x_{i1}(s)$ and $x_{i2}(t)$ and respective true functions $\xi_1(s) = \alpha(s)$ and $\xi_2(t) \equiv 0.5$.
In Figure 2.6(b), the setting includes the normal density generating process and B-spline generating process of $V = 15$ generating functions, $K = 30$ grid points for $x_{i2}(t)$ and respective true functions $\xi_1(s) = \alpha(s)$ and $\xi_2(t) = \alpha(t)$.
In Figure 2.6(c), the setting includes the normal density generating process and B-spline

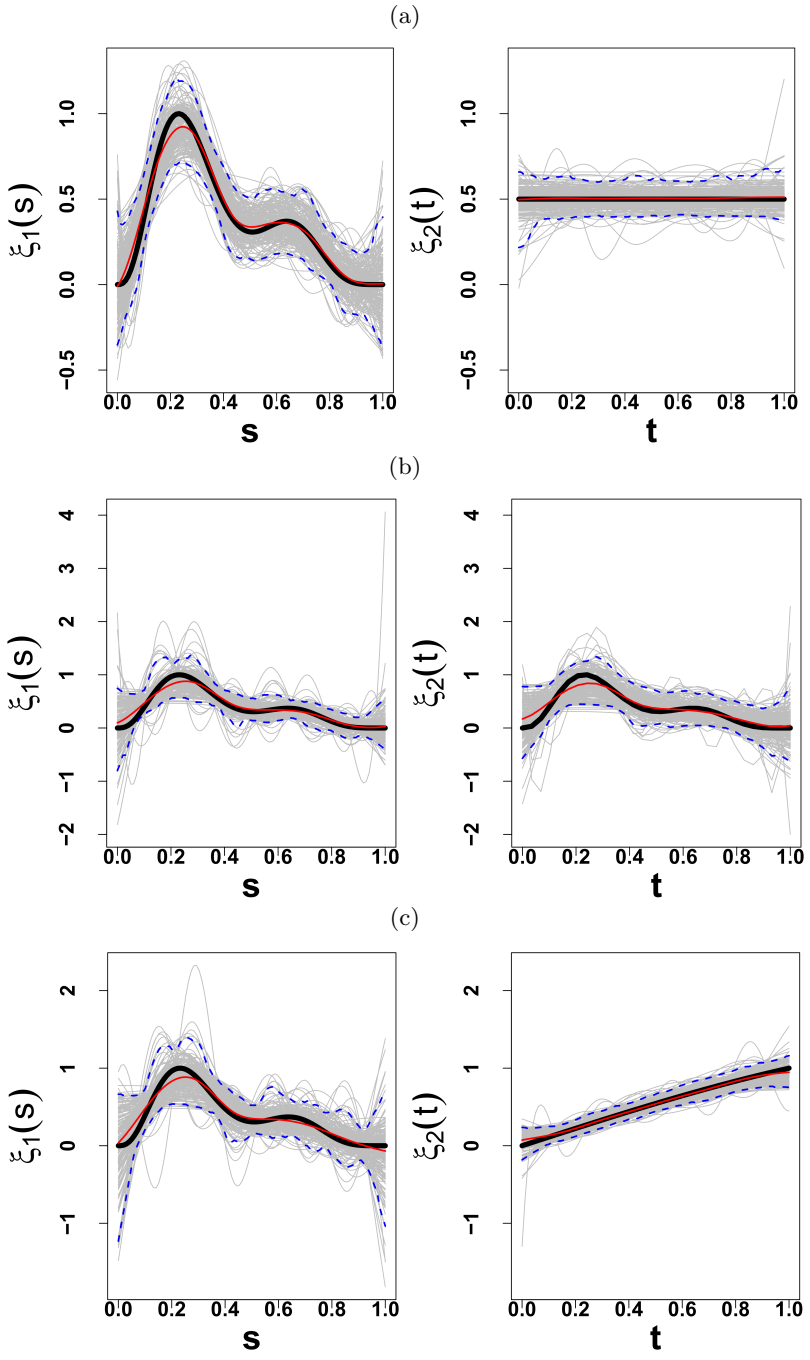


Figure 2.6: Example results for the logistic model with main effects only.

generating process of $V = 15$ generating functions, $K = 100$ grid points for both $x_{i1}(s)$ and $x_{i2}(t)$ and respective true functions $\xi_1(s) = \alpha(s)$ and $\xi_2(t) = \gamma(t)$.
In all three cases, the means of the estimates are acceptable close to the true functions. It turns out that results for $n = 1000$ observations are similar to those for the linear model with smaller sample sizes with respect to the effects of the parameters, and a discussion is omitted for brevity. Note that the absolute level of a constant true function ($\xi_2(t) = 5$ in the linear and $\xi_2(t) = 0.5$ in the binomial model) has no influence on the estimation quality, as expected.

2.5 Application to Spectroscopic Data

Functional data analysis has previously been used to analyse spectroscopic data, especially near infrared spectra, of various materials (see for example Ramsay and Silverman, 2005). In our study, we obtained two spectra types with different wavelength ranges for each of $n = 129$ fossil fuel samples with the goal to infer the respective heat values. The latter is directly inferable from the calorific value, which depends on the chemical composition of the sample. Also characteristic for the chemical composition of a sample are the vibrations and harmonics of excited molecules when measuring a spectrum, and thus a correlation can be expected. For model calibration and validation, the heat values were determined by laboratory analyses.
The two spectra types are near infrared spectra, measured at 2307 equidistant wavelengths, and ultraviolet-visible spectra, measured at 1335 equidistant wavelengths. We use spectra referenced to a reference spectrum by a suitable modified Lambert-Beer law to eliminate any dependency on the optical setup. Before modeling, the spectra have been smoothed using B-splines. For dimension reduction, the smoothed signals were evaluated at $J = 134$ equidistant points for the UV-VIS and at $K = 231$ equidistant points for the NIR spectra. The smoothed spectra $x_i(s)$, $i = 1, \dots n$, were then centered by subtracting the mean $\frac{1}{J}\sum_{j=1}^{J} x_i(s_j)$ (curve-wise centering) to remove the spectrum offsets resulting from the optical setup and carrying no relevant information. Figure 2.7 shows the smoothed and centered UV-VIS (upper left panel) and NIR spectra (upper right panel) as well as a histogram of the heat values (lower panel).
Other, less scientifically adequate preprocessing options we experimented with yielded similar or slightly inferior results, see also Section 2.5.2.

2.5.1 Results

We fit models with main effects only, interaction effect only as well as the full model to assess the interactions' effect on prediction. We use six basis functions for each marginal basis. Models with up to ten marginal basis functions showed no improvement with respect to the interaction models' mean squared error of prediction in a sensitivity analysis.
In each of 25 replications, $n_{val} = 13$ curves are drawn randomly as validation data, the

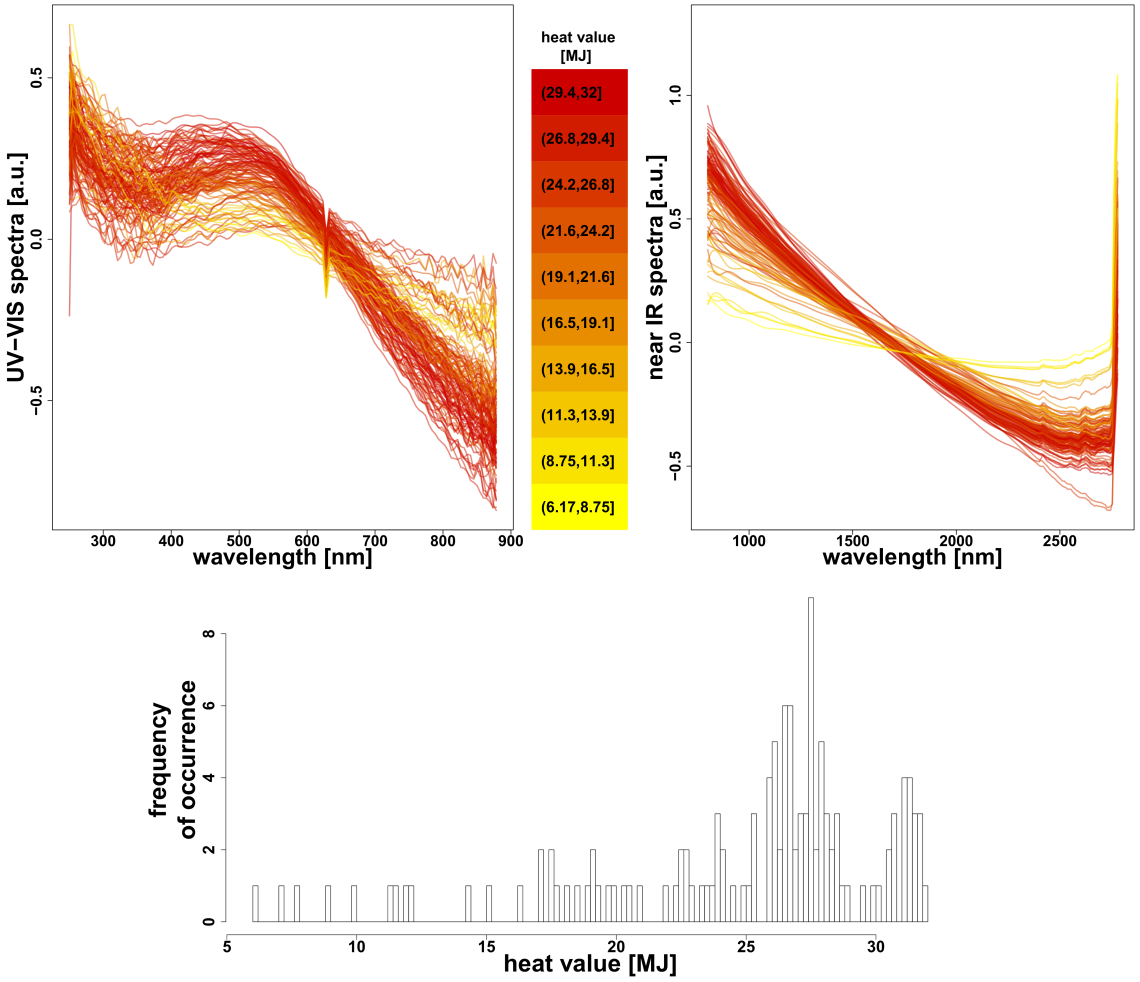


Figure 2.7: The upper left panel shows the smoothed ultraviolet-visible spectra, evaluated at $J = 134$ equidistant points distributed over the whole original wavelength range. The upper right panel shows the smoothed near infrared spectra, evaluated at $K = 231$ equidistant points distributed over the whole original range. Both spectra types have been centered after referencing and smoothing. The corresponding heat values range between 6.17 MegaJoule (MJ) and 31.96 MJ, with a histogram of all values given in the lower panel.

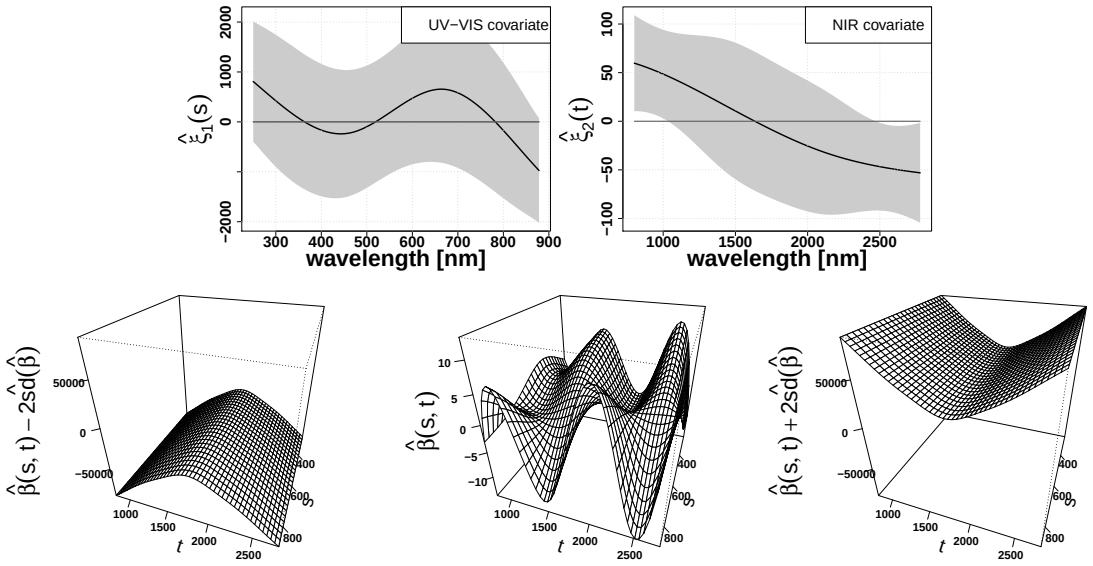


Figure 2.8: Estimates of the full model fitted on the full data set. The upper row shows the main effect estimates with pointwise confidence bands. In the lower row, the interaction surface estimate (middle) ± two times the estimated standard errors (left and right) are given.

remaining $n_{cal} = 116$ curves are used to fit the model.
Figure 2.8 shows the estimates for the full model fitted to the full spectra data. The first main effect estimate $\hat{\xi}_1(s)$ belonging to the UV-VIS spectra implies that UV-VIS spectra with high values around 300nm and 650nm and negative values from around 800nm correspond to high heat values. This result is consistent with most of the spectra. The NIR spectra coefficient estimate $\hat{\xi}_2(t)$ is nearly linear and slightly decreasing. Thus, the higher the spectra values are in the beginning and the lower they are in the end, the higher are the corresponding heat values. The values of the $\hat{\xi}_2(t)$ estimate are small compared to $\hat{\xi}_1(s)$. Most information contained in the NIR spectra seems to be included in the estimated interaction surface, which is of a complex, sine-like form.

We compare the predictive performance of our approach for this data set to that of the four other methods introduced in Section 2.3.
The four methods are denoted by their function call names `fregre.glm`, `fregre.gsam`, `fregre.gkam`, implemented in the `fda.usc` package (Febrero-Bande et al., 2013), and function `pfr` implemented in the `refund` package (Crainiceanu et al., 2013). In Figure 2.9, the R^2_{adj} for our approach can be found on the left, where the full model shows the highest

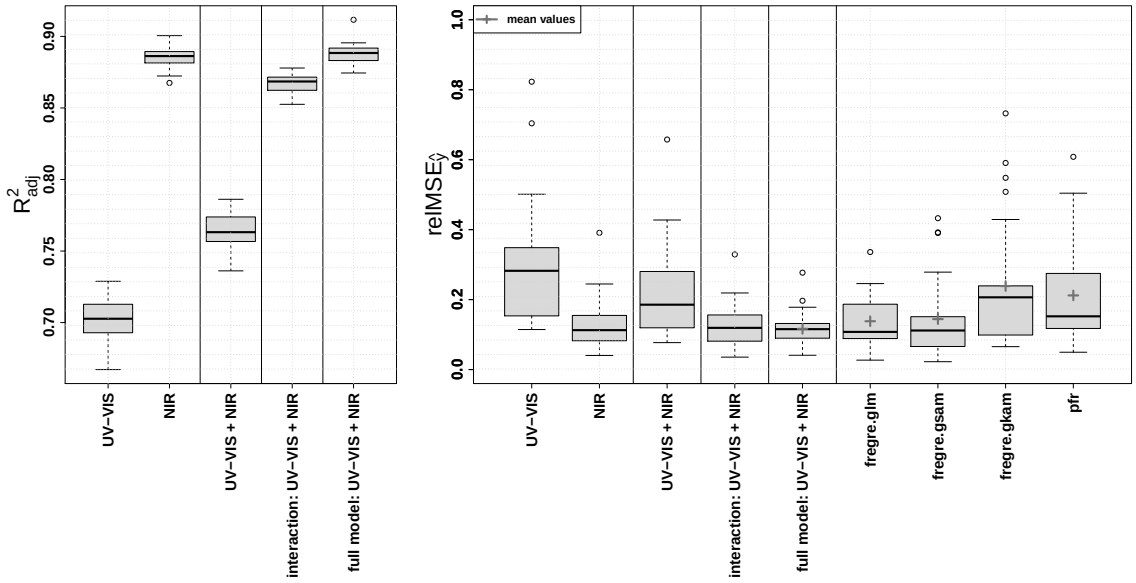


Figure 2.9: Adjusted R^2_{adj} (left) and relative mean squared error of prediction in the validation data (right) for different models including up to two main effects and up to one interaction effect. For the relMSE$_{\hat{y}}$, the results of four alternative estimation methods "fregre.glm", "fregre.gsam", "fregre.gkam" and "pfr" are added. The gray crosses give the means of the respective relMSE$_{\hat{y}}$. The boxplots show variation of the two quantities across 25 splits into calibration and validation data.

R^2_{adj}. The relative mean squared errors of prediction

$$\text{relMSE}_{\hat{y}} \quad = \quad \frac{1}{25} \sum_{o=1}^{25} \frac{\sum_{i=1}^{n_{val}} (y_i - \hat{y}_i)_o^2}{\sum_{i=1}^{n_{val}} (y_i - \bar{y})_o^2},$$

with i ranging through the validation data and $\bar{y}$ the mean of the y_i in the calibration data, for all methods are shown on the right. While the medians of methods `fregre.glm`, `fregre.gsam` and of our full model are very similar, the latter yields the smallest variation across splits as well as the smallest mean relMSE$_{\hat{y}}$ across all estimation methods. Thus, prediction for the full model seems most stable and reliable and in this application, the interaction of the two spectra contains information and improves prediction.

2.5.2 Influence of Preprocessing

Besides the preprocessing presented above, including smoothing and centering, we tested several alternative preprocessing options, which yielded similar or slightly inferior results.

To evaluate how the estimates differ depending on whether the original covariates $x_i(s)$ or the curve-wise centered covariates $\tilde{x}_i(s) = x_i(s) - \frac{1}{J} \sum_{j=1}^{J} x_i(s_j)$ are used for modeling, we

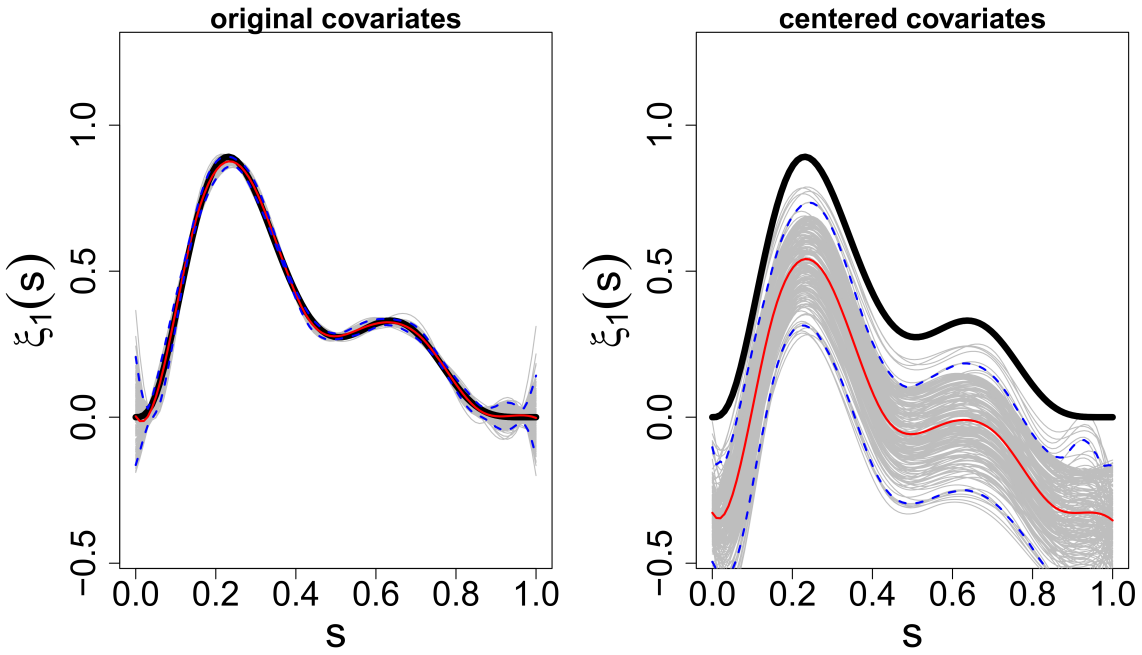


Figure 2.10: Main effect model estimates when using the original (left) and the curve-wise centered (right) covariates. The true parameter functions (black), the mean (red) of 200 estimates (gray), and the 2.5% and 97.5% quantiles (dashed blue) are shown.

first conducted a small simulation study. Two models are examined: one with an intercept and a single main effect,

$$g(\mu_i) \quad = \quad \beta_0 + \int x_{i1}(s)\xi_1(s)ds, \tag{2.6}$$

and an interaction effect model IIb) (cf. Table 2.1). For both, the modeling uses the generating process with normal densities for the covariates $x_1(s)$, and the sine-function generating process for $x_2(t)$, each with $J = K = 100$ equidistant grid points in $\mathbb{D} = \mathbb{E} = [0, 1]$. The number of observations is differing, with $n \in \{50, 250, 500, 1000\}$. The true parameter functions are $\xi_1(s) = \alpha(s)$ and $\beta(s, t) = \rho(s, t)$, the conditional distribution of the responses y_i is taken to be normal. Thus, apart from using either $x_i(s)$ or $\tilde{x}_i(s)$, the modeling is identical, following model definitions (2.6) or IIb).

Figure 2.10 shows the estimates of the single main effect model. The first panel gives the estimates of the original generated covariates, the second panel the estimates of the identical, but curve-wise centered covariates. The number of observations here is $n = 50$. The true, univariate parameter functions are depicted as black lines, the $R = 200$ estimates as gray lines, their means as red lines, and the 2.5% and 97.5% pointwise quantiles as dashed

blue lines.
The absolute values of the main effect estimates resulting from curve-wise centered covariates are lower than those from the uncentered covariates. This is due to the `gam`-function that is used for model estimation. To ensure identifiability of the estimates, the latter are centered if the covariates' means are constant (see Wood, 2013, entry "linear.functional.terms"). Despite this shift, the estimates' means exhibit a very similar progression for both data situations. The covariates' means seem to carry information, since the estimates from the curve-wise centered covariates show more variation around their mean than the estimates resulting from the orignal covariates. This variability decreases with increasing numbers of observations n.

Figure 2.11 shows the estimates for the interaction effect model when using the original covariates to build the covariate interaction matrix (upper panels), and using the curve-wise centered covariates (lower panels). Color coding is as before. The interaction effect estimates which base on the curve-wise centered covariates show an obvious scale shift and shape deviation compared to the true bivariate parameter function. When building the covariate interaction matrix on the original covariates, and centering it afterwards, the deformation disappears, while, analogously to the main effect model, a scale shift remains (see Figure 2.12).
To understand this one has to take a closer look at the interaction matrix building. Let $\boldsymbol{\omega}_i = (\boldsymbol{x}_{i2}^T \otimes \boldsymbol{x}_{i1}^T)$ denote the ith row of the covariate interaction matrix build from the original covariates, $\tilde{\boldsymbol{\omega}}_i = (\tilde{\boldsymbol{x}}_{i2}^T \otimes \tilde{\boldsymbol{x}}_{i1}^T)$ the ith row of the matrix build from the curve-wise centered covariates, and $\tilde{\boldsymbol{\omega}}'_i = \boldsymbol{\omega}_i - \frac{1}{JK}\sum_{j=1}^{J}\sum_{k=1}^{K}\boldsymbol{\omega}_i$ the ith row of the matrix build from the original covariates with subsequent centering. Further, define $\boldsymbol{I}_J = (1, \dots 1)^T \in \mathbb{R}^{J\times 1}$, $\boldsymbol{I}_K$ analogously, and let $\boldsymbol{a}_{i1} = \frac{1}{J}\sum_{j=1}^{J} x_{i1}(s_j)\boldsymbol{I}_J$ and $\boldsymbol{a}_{i2} = \frac{1}{K}\sum_{k=1}^{K} x_{i2}(t_k)\boldsymbol{I}_K$ be vectors containing the curve-wise means. Then it can be shown that

$$\begin{aligned} \tilde{\boldsymbol{\omega}}_i &= \boldsymbol{\omega}_i - const_i - \\ &\quad \boldsymbol{a}_{i2} \otimes \boldsymbol{x}_{i1}^T - \boldsymbol{x}_{i2}^T \otimes \boldsymbol{a}_{i1}, \text{ while} \\ \tilde{\boldsymbol{\omega}}'_i &= \boldsymbol{\omega}_i - const'_i. \end{aligned}$$

The dependence of $\tilde{\boldsymbol{\omega}}_i$ on the Kronecker products of the covariates and the covariates' means results in deformed estimates, since this term alters the model matrix relative to that of model IIb), i.e. $\boldsymbol{\omega}_i$.

From Figure 2.13, it can be seen that the relMSE$_\beta$ (cf. Section 2.4.1) decreases, even for small numbers of observations, when using the original covariates to build the covariate interaction matrix with subsequent centering instead of simple covariate centering. This is natural, since the estimates' shapes are similar to the true function. The shift due to the `gam`-function estimate centering, ensuring identifiability, remains. Nonetheless, the effect of the simple curve-wise centering on the prediction performance, represented by relMSE$_y$,

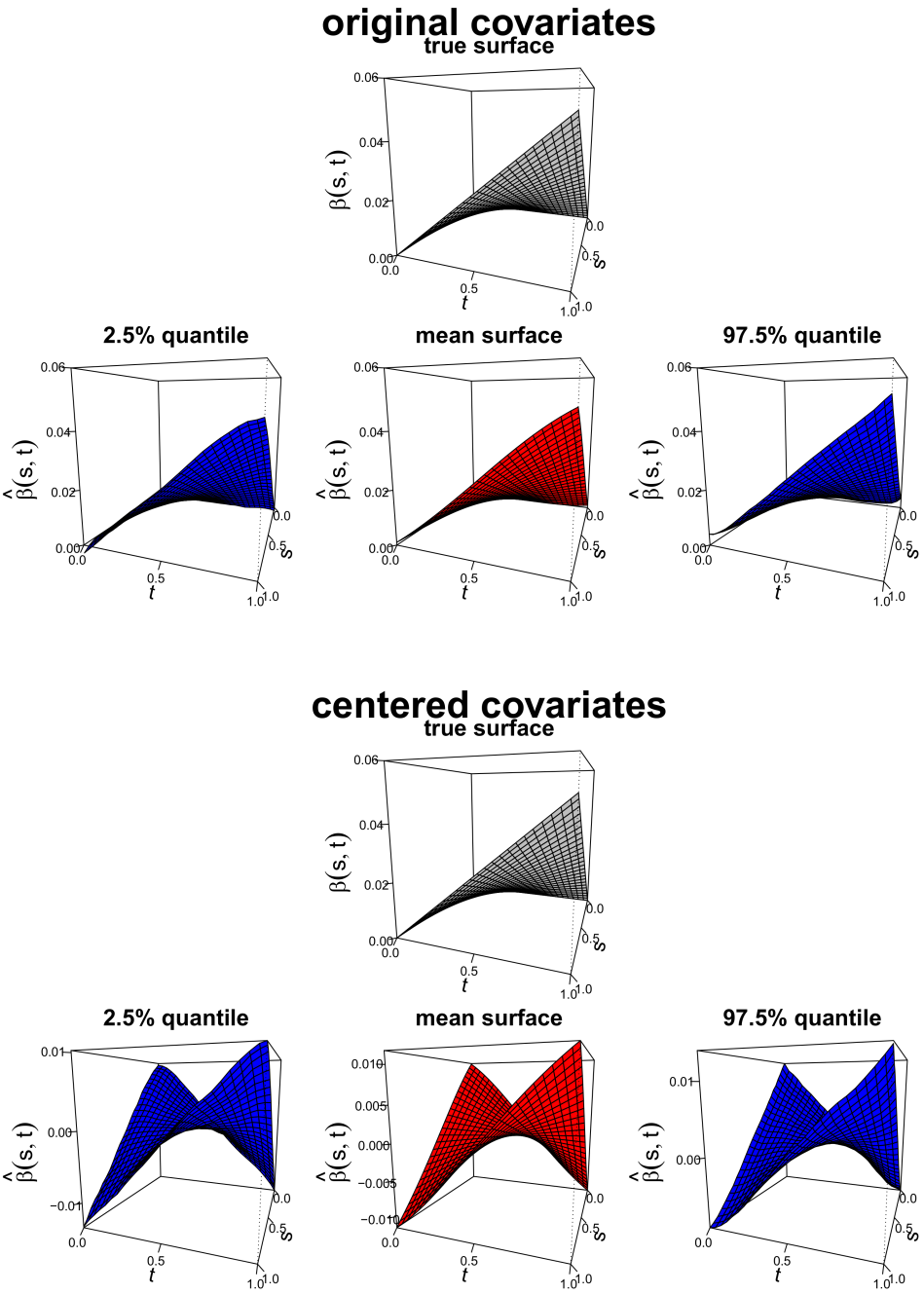


Figure 2.11: Interaction effect model estimates basing on the original (upper panels) and the curve-wise centered (lower panels) covariates. The true parameter function (black), the mean (red) of 200 estimates, and the 2.5% and 97.5% quantiles (blue) are shown.

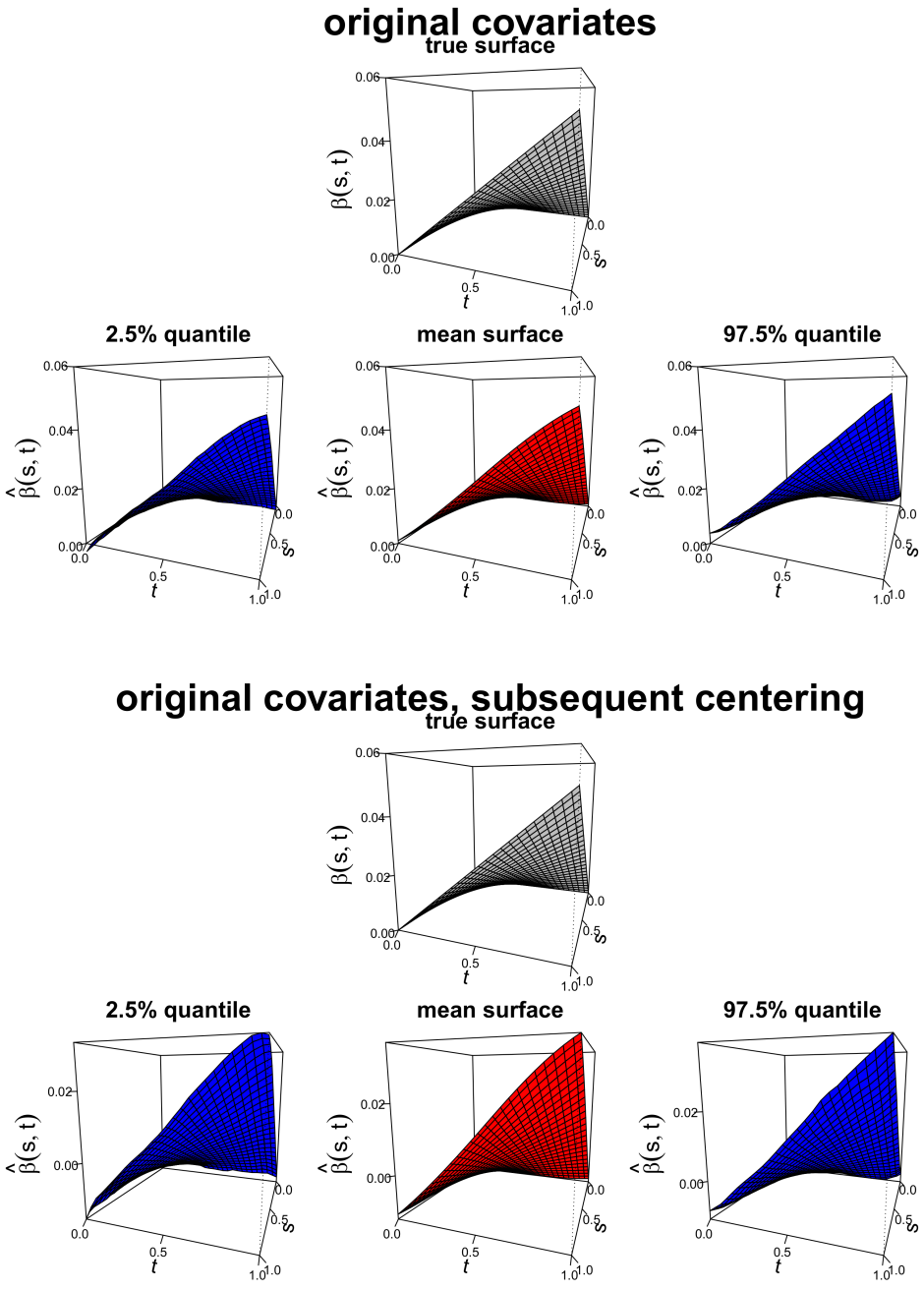


Figure 2.12: Interaction effect model estimates when using the original covariates to build the covariate interaction matrix (upper panels), and when using the original covariates, followed by row-wise centering of the covariate interaction matrix (lower panels). The true parameter function (black), the mean (red) of 200 estimates, and the 2.5% and 97.5% quantiles (blue) are shown.

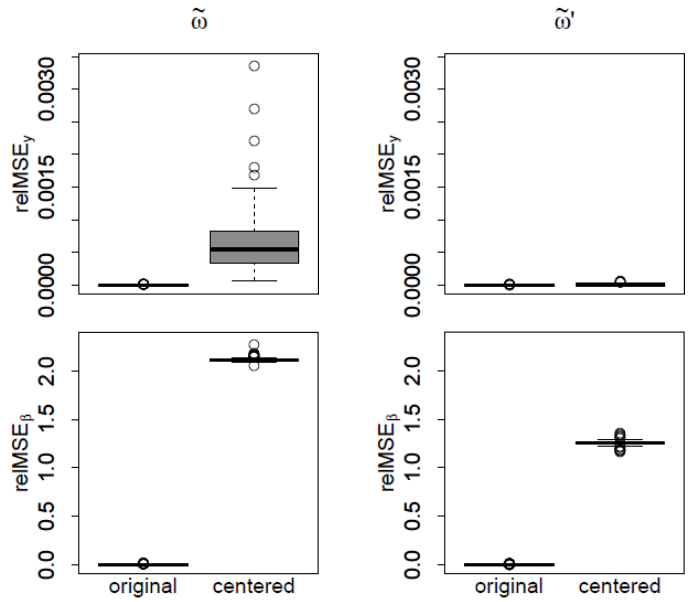


Figure 2.13: relMSE$_y$ and relMSE$_\beta$ across the 200 modeling replications for the interaction effect model comprising $n = 50$ observations. The left column shows results when using the curve-wise centered covariates to build the covariate interaction matrix. The right column shows results using the original covariates with subsequent centering. In each panel, the first boxes give results when using the original covariates, the second boxes the results of the respective centering.

is of orders 10^{-3} to 10^{-4} and thus marginal. As a consequence, one should take care when building the covariate interaction matrix. If prediction performance is of main interest, the possibly scientifical adequate use of the curve-wise centered covariates to build the covariate interaction matrix can be applied. If interpretation of the estimates is of importance, the covariate interaction matrix should be build from the original covariates and be row-wise centered afterwards.

The above results suggest that centering has only a mild influence on the prediction results of the spectra. Various preprocessings have been applied to the spectra, Table 2.2 gives respective details. In preprocessing 7, the default is used for penalization, resulting in cubic P-splines with a second order difference penalty (`m = c(2,2)`, see Wood, 2013, entry "smooth.construct.ps.smooth. spec"). Else, the modeling uses cubic P-splines with a first order difference penalty. Analogously to the previous section, each model was built 25 times on $n_{cal} = 116$ randomly drawn calibration spectra, and validated on the remaining data.
Figure 2.14 compares the model results per preprocessing for each possible covariate combination, constituting models with up to two main effects and an interaction term. The first row of panels shows R^2_{adj} for the lowest ($nk = 4$, wide boxes) and the highest ($nk = 10$,

Option	Preprocessing steps
Preprocessing 1	1. order differentiation, smoothing*, dimension reduction*
Preprocessing 2	smoothing*, dimension reduction*, centering†, modeling assumes a Gamma-distributed response
Preprocessing 3	smoothing*, dimension reduction*, centering† (identical to the preprocessing examined in Section 2.5.1)
Preprocessing 4	dimension reduction*
Preprocessing 5	smoothing*, dimension reduction*
Preprocessing 6	dimension reduction*, centering†
Preprocessing 7	smoothing*, dimension reduction*, using the "`default`" penalization

Table 2.2: All preprocessing options that have been applied to the spectra data.

superimposed narrow boxes) tested numbers of marginal bases functions nk. The second and third rows of panels show the same for the $\text{relMSE}_{\hat{y}}$ and relMSE_y. Color coding is with respect to the preprocessing options. For results of other values of nk, please refer to Appendix A.1.

Across all preprocessing options, results for $nk = 4$ are worse than for $nk = 10$, although this is not a monotone trend, as can be seen from the figures in Appendix A.1. The modeling of the interaction term seems to work quite well for all preprocessings, reflected in values of R^2_{adj} above 0.8. The relMSE_y behaves analogously to the R^2_{adj}. Preprocessing 2, including the assumption of a Gamma-distributed response, shows some variation across the splits for especially the NIR main effect and the two main effects models. Here, the addition of an interaction term seems to stabilize the model, since the variation across splits decreases.

With respect to the prediction performance, preprocessings 2 and 6 (unsmoothed, curve-wise centered signals) show the highest $\text{relMSE}_{\hat{y}}$ values throughout all models. Preprocessing 1 using the smoothed first derivatives of the signals shows smallest median $\text{relMSE}_{\hat{y}}$ values for small nk, whereas its mean $\text{relMSE}_{\hat{y}}$ values are comparable to other preprocessings.

The only difference between preprocessings 3 and 5, and 4 and 6, is the curve-wise centering in 3 and 6. A closer look at Figure 2.14 suggests that the curve-wise centering results in a bit lower R^2_{adj}, as well as in higher relMSE_y and $\text{relMSE}_{\hat{y}}$ values. Additionally, comparing preprocessings 3 to 6 and 4 to 5, preprocessings including smoothing tend to better results, especially in terms of prediction performance. Preprocessing 3, which had been chosen to be examined further in the above Sections 2.5 and 2.5.1, performs among the best of all options in terms of the $\text{relMSE}_{\hat{y}}$, depending on the model type.

* analogously to the description at the beginning of Section 2.5

† by subtracting the mean $\frac{1}{J}\sum_{j=1}^{J} x_i(s_j)$ (curve-wise centering)

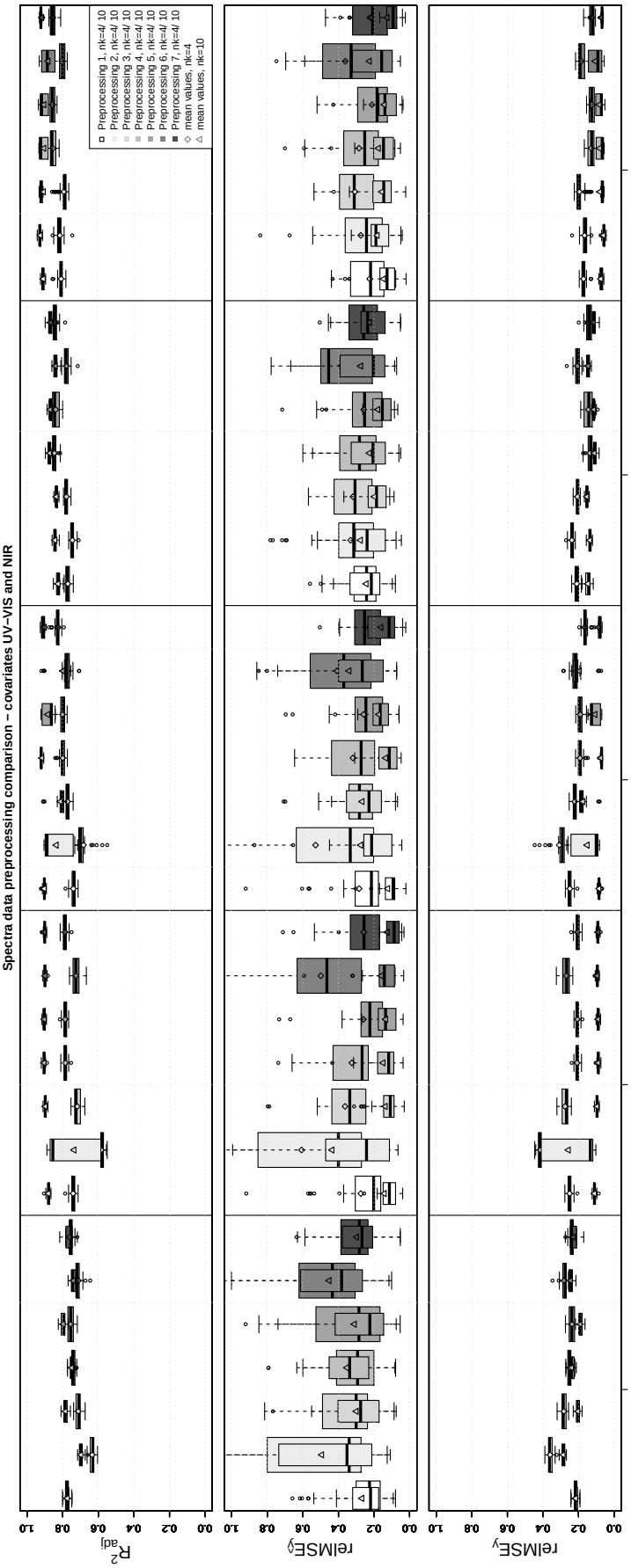


Figure 2.14: First row: R^2_{adj} for the lowest ($nk = 4$, wide boxes) and the highest ($nk = 10$, superimposed narrow boxes) tested numbers of marginal bases functions, per preprocessing and model type. The second and third rows of panels show the same for the relMSE$_{\hat{y}}$ and relMSE$_y$.

In summary, $nk = 4$ seem not enough marginal bases functions to build a well-performing model, whereas $nk = 6$ or $nk = 7$ yield good results for all preprocessing options. Smoothing of the UV-VIS and NIR spectra data seems advisable, while curve-wise centering has a small negative impact on the model quality and prediction performance. Preprocessing 3 remains the best choice, since its performance is comparable to the other preprocessing options, and it is the scientifically most adequate preprocessing for the spectra data. Nonetheless, the results of the small simulation at the beginning of this section suggest that with preprocessing 3, the interpretation of the estimated bivariate functions has to be handled with care. Although the interpretation especially of the main effect estimates in Section 2.5.1, i.e. Figure 2.8, seems to be in accordance with the data, the validity of the bivariate estimate can not be guaranteed.

2.6 Application to Cell Chip Data

Cell chip sensors are used for example to monitor the quality of drinking water or ambient air, see e.g. Bohrn et al. (2012). They consist of a silicon chip providing electrical signals. Their surfaces are covered with a monolayer of a living cell population, which reacts to pollutants in the cell culture medium supplying them with nutrients. Ion sensitive field effect transistor (ISFET) signals relate to the acidification rate of the medium in which the cells are contained, due to the excretion of acidic metabolites. Interdigitated electrode structure (IDES) signals can be used to draw conclusions about the cell morphology and cell adhesion on the surface of the sensor chip. CLARK-electrodes measure the oxygen contained in the medium, a proxy for the respiration activity of the cells (Thedinga et al., 2007; Ceriotti et al., 2007). Figure 2.15 shows a cell chip in its housing (black cavity and socket) and its surface with the three sensor types in the upper row. Our data was recorded after applying nutrient medium, polluted with different concentrations of paracetamol (chem.: AAP), on chinese hamster lung fibroblast cells. It is possible that, upon AAP treatment, the pH-value of the nutrient medium changes, such that the ISFET-signals should vary according to the AAP concentration. A morphological change of cells being under stress is supposed to be reflected in the IDES-signals. The respiration activity of the cells might change, and with that the CLARK-signals. Thus, it is reasonable to expect a correlation between the AAP concentration and the sensor signals.
Our data set consists of $n = 280$ measurements. There were seven concentrations of AAP, namely 0mM, 0.5mM, 1.5mM, 2.5mM, 3.5mM, 5mM and 6mM. The first 170 minutes of the measurement correspond to an acclimatisation phase with medium (no AAP) flowing over the cells. As this part is not informative, we do not include it in the analysis. After this first phase, measurements for each of the three signals are recorded at 36 time points on an equidistant grid from 170 minutes to 381 minutes. Before modeling, the signal curves have been smoothed by B-splines. Since the measuring apparatus preprocesses the measured flow rates, the measurement curves are in arbitrary units. Figure 2.15 shows the

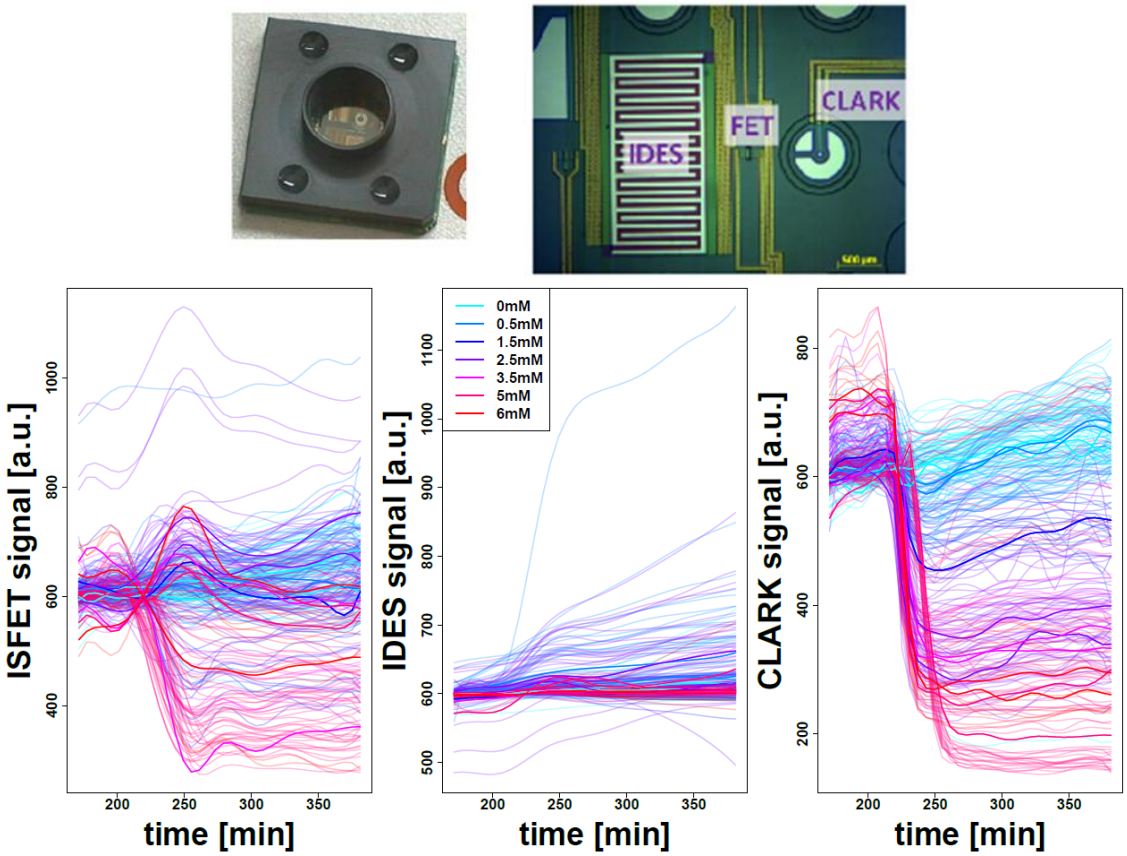


Figure 2.15: Upper row: A cell chip of the Micronas company in its housing (left) and its surface with the three sensor types (right). Lower row: The $n = 280$ signals for each of the three sensor types, measured at 36 equidistant time points in arbitrary units ([a.u.]). They have been smoothed using B-splines. The colors represent the seven AAP concentrations.

preprocessed measurement curves in the lower panels.
As small differences in the level at the beginning can be assumed to be due to random variation we considered shifting the smoothed measurements to start at 100 [a.u.], both by subtraction and multiplication. As results were comparable, we present results only for the non-shifted curves, which were most stable under subsampling.

2.6.1 Results

We fit models with up to three main effects, corresponding to the ISFET-, IDES- and CLARK-signals, as well as models including a two-way interaction. Each of the models with two covariates is calculated without and with interaction in order to assess the interactions' effect on prediction. The number of basis functions is four for each marginal basis.

Four basis functions are sufficient in this application with relatively smooth signals, and sensitivity analyses with up to ten marginal bases functions showed no relevant improvement with respect to the relative mean squared error of prediction.
For each of 25 replications, we have randomly drawn $n_{val} = 56$ observations as validation data, the remaining $n_{cal} = 224$ observations are used to fit the models. Figure 2.16 shows the adjusted R^2_{adj} across all replications. The relative mean squared error of prediction (cf. Section 2.5.1) for the AAP concentrations of the validation data is given on the right. The model including IDES-signals only yields the worst results with an adjusted R^2_{adj} very near to zero and relative mean squared errors above 0.9 for every replication. This is likely due to the low rank of the IDES measurements, which indicates very limited information content. The IDES-signals also do not improve fit and prediction in the other models examined.
We therefore consider models using ISFET- and CLARK-signals. The model with main effects only and the full model yield the best results, with comparable relMSE$_{\hat{y}}$ and R^2_{adj} values. The estimates of the full Model (2.3) applied to the full data, being typical also for the single replicates, are shown in Figure 2.17. For ISFET-signals, the linearly, slightly decreasing $\hat{\xi}_1(s)$ implies that lower values at the end of the measurement relate to a high AAP concentration. CLARK curves with high values in the beginning and low values especially around 300 minutes correspond to the highest amounts of AAP. The interaction surface estimate is nearly constant and negative. Thus, ISFET- and CLARK-signals with high values and similar curve progression correspond to lower AAP values than expected from the main effects alone. Despite the small negative values of the interaction surface estimate, the high values of the products $x_{i1}(s)x_{i2}(t)$ mean that the interaction effect is comparable in magnitude to the main effects.
In this application, the interaction only slightly contributes to prediction, if at all, and is not significant. However, our approach allowed us to check whether the additivity assumption of linear covariate effects was adequate.

2.6.2 Influence of Preprocessing

Besides the preprocessing presented above, including smoothing, we tested several preprocessing options, which yielded similar or slightly inferior results. The details on the preprocessing steps can be taken from Table 2.3. The models use cubic P-splines with a first order difference penalty. Analogously to the previous section, each model was built 25 times on $n_{cal} = 224$ randomly drawn calibration curves, and validated on the remaining data.

Figure 2.18 compares results per preprocessing for each possible covariate combination of the ISFET- and CLARK-signals, constituting models with up to two main effects and an interaction term. The first row of panels shows R^2_{adj} for the lowest ($nk = 4$, wide boxes) and the highest ($nk = 10$, superimposed narrow boxes) tested numbers of marginal bases

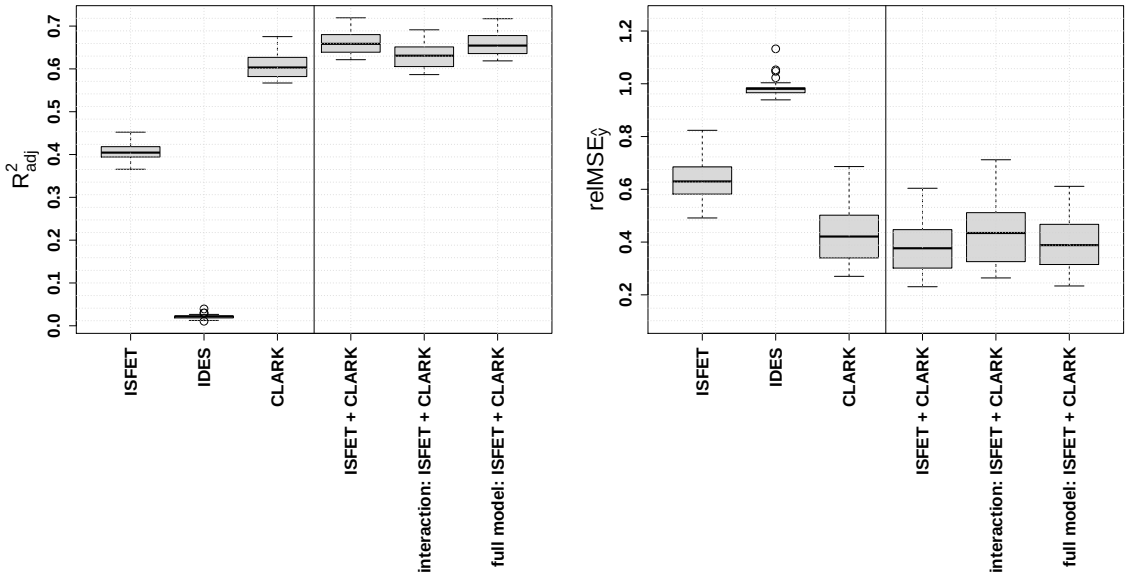


Figure 2.16: Adjusted R^2_{adj} (left) and relative mean squared error of prediction in the validation data (right) for different models including up to two main effects and up to one interaction effect. The boxplots show variation in both quantities across 25 splits into calibration and validation data.

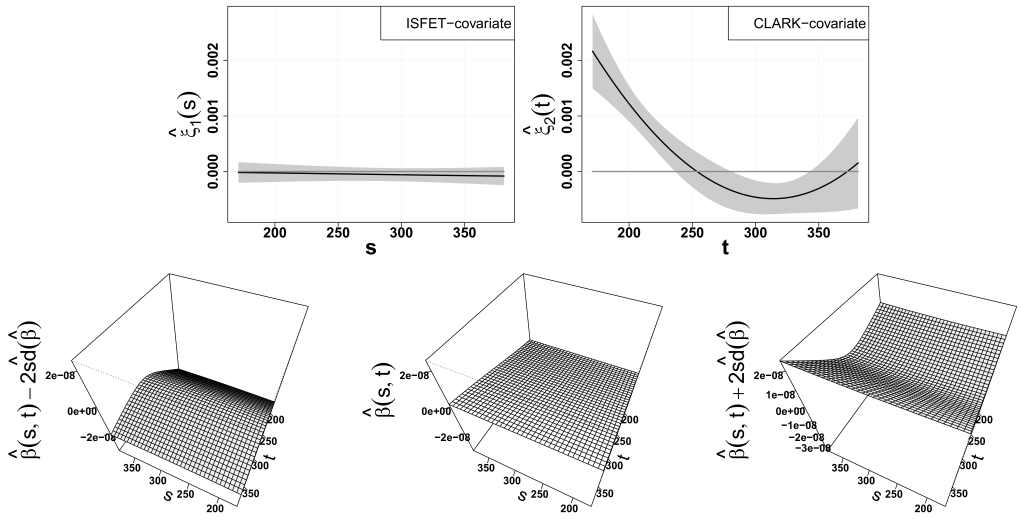


Figure 2.17: Estimates of the full model fitted on the full data set. The sensor signals used as covariates are the ISFET- and CLARK-signals. The upper row shows the main effect estimates with pointwise confidence bands. The lower row shows the interaction surface estimate (middle) ± two times the estimated standard errors (left and right).

Option	Preprocessing steps
Preprocessing 1	centering†
Preprocessing 2	curve shifting such that each curve has an initial value of 100 [a.u.] (by subtraction)
Preprocessing 3	raw curves
Preprocessing 4	smoothing* (identical to the preprocessing examined in Section 2.6.1)
Preprocessing 5	smoothing*, centering†
Preprocessing 6	smoothing*, centering (by subtracting the overall mean $\frac{1}{NJ}\sum_{i=1}^{N}\sum_{j=1}^{J} x_i(s_j)$)
Preprocessing 7	smoothing*, curve shifting such that each curve has an initial value of 100 [a.u.] (by subtraction)
Preprocessing 8	smoothing*, curve shifting such that each curve has an initial value of 100 [a.u.] (by multiplication)

Table 2.3: All preprocessing options that have been applied to the cell chip data.

functions nk. The second and third rows of panels show the same for the relMSE$_{\hat{y}}$ and relMSE$_y$. Color coding is with respect to the preprocessing options. As before, we will focus on the ISFET and CLARK covariates. Analogous plots for models based on the ISFET- and IDES-signals, or the IDES- and CLARK-signals, respectively, can be found in Appendix A.1. Overall, results are similar to those of the ISFET and CLARK models discussed in the following.

For models containing the ISFET or CLARK covariates, the relMSE$_y$ behaves analogously to the R^2_{adj}. In general, models containing more marginal bases functions tend to give better results, although this is not a monotone trend, as can be seen from the figures in Appendix A.1. The results across preprocessings, for either $nk = 4$ or $nk = 10$, are very similar, except for the interaction effect model. Here, it seems that preprocessing 8, including smoothing and a shift of the signals via multiplication, shows slightly more variation across the splits than preprocessing 7, in which the curves are smoothed and shifted by subtraction. Comparing preprocessings that either include smoothing or not, or curve-wise centering or not, both preprocessing steps do not seem to have any noteworthy influence on the prediction.

It is especially interesting that centered covariates (both by curve-wise centering or subtracting the curves' overall mean) result in obviously worse interaction effect models. In the full Model (2.3), the additional main effects seem to compensate the negative impact of centering. This is in contrast with the results from the spectra data, where centering had only a small influence on the model performance. To summarize, the choice of the number of marginal bases functions depends both on the covariates used and the chosen model, with $nk = 4$ yielding good results for most variants. Except for the interaction

* analogously to the description at the beginning of Section 2.6

† by subtracting the mean $\frac{1}{J}\sum_{j=1}^{J} x_i(s_j)$ (curve-wise centering)

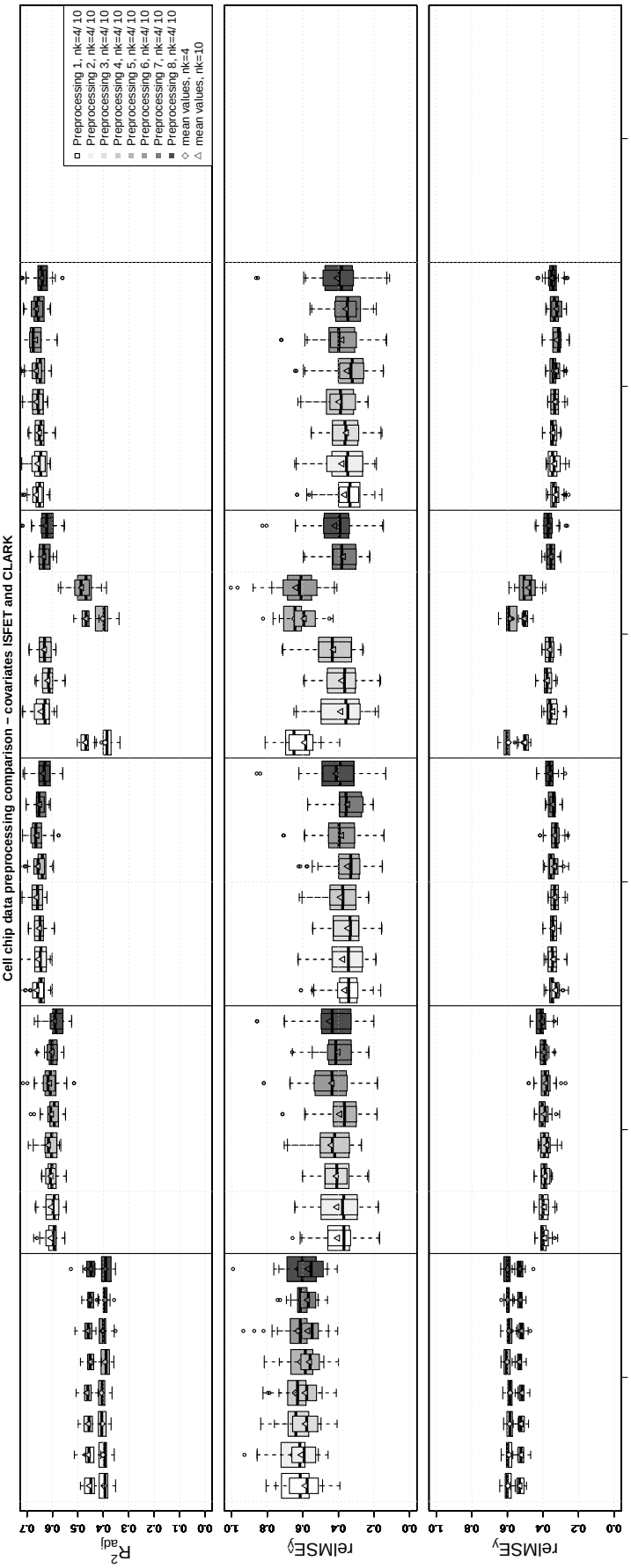


Figure 2.18: First row: R^2_{adj} for the lowest ($nk = 4$, wide boxes) and the highest ($nk = 10$, superimposed narrow boxes) tested numbers of marginal bases functions, per preprocessing and model type. The second and third rows of panels show the same for the relMSE$_{\hat{y}}$ and relMSE$_y$.

model, neither smoothing nor centering or shifting has much impact on the model quality and prediction performance. Preprocessing 4 remains the most stable variant with respect to the resulting estimates.
The identifiability issue discussed in Section 2.5.2 certainly affects the estimates of the cell chip data as well as those of the spectra data. Estimate interpretations, especially concerning the bivariate estimate, in Section 2.6.1 hence have to be considered with caution, too. But since, by definition, identifiability does not alter the prediction results, the deterioration of interaction models caused by centering has to be understood as an independent effect. Thus, the large negative effect of centering the data on the interaction model performance not only stresses the importance of choosing models being appropriate for the data at hand, but also underlines that care should be taken when manipulating data.

2.7 Identifiability in the Context of Scalar-on-Functions Regression

In regression models including more than one unknown function, as for example in GAMs as (2.2), the uniqueness of the respective estimated functions is an essential topic, see e.g. Buja et al. (1989). Identifiability is also important in regression models with functional covariates, especially when interpreting coefficient functions. Theoretical results relevant to our context can be found for example in Cardot et al. (2003), Cardot and Sarda (2005) and Prchal and Sarda (2007). Practical results on the issue of identifiability using a penalized spline approach in the context of function-on-function regression can be found in Scheipl and Greven (2016).

Let the functional covariates $x_{i1}(s)$ and $x_{i2}(t)$ be stochastic processes, decomposed in the Karhunen-Loève expansion as $x_{i1}(s) = \sum_{q_1=1}^{\infty} \iota_{i1q_1}\chi_{1q_1}(s)$, with $\int \chi_{1a}(s)\chi_{1b}(s)ds = \delta_{ab}$, δ_{ab} being Kronecker's delta. The ι_{i1q_1} are uncorrelated random variables with mean zero and variance $\text{var}(\iota_{i1q_1}) \neq 0$, and $\text{var}(\iota_{i1q_1})$ and $\chi_{1q_1}(s)$ are eigenvalues and eigenfunctions of the covariance of $x_{i1}(s)$, respectively. $x_{i2}(t)$ is decomposed analogously. Let us now assume a single main effect model of the form

$$g(\mu_i) \quad = \quad \eta_i = \beta_0 + \int x_{i1}(s)\xi_1(s)ds.$$

If the covariance of $x_{i1}(s)$ can be well represented using only a finite number Q_1 of eigenfunctions, functions in the null space of the covariance, spanned by eigenfunctions with $q_1 > Q_1$, can be added to $\xi_1(s)$ without changing the fit to the data (see e.g. He et al., 2000; Prchal and Sarda, 2007). This is counteracted by the penalty, which typically ensures a unique solution by using a smoothness assumption, unless the null-space of the penalty overlaps the null-space of the covariance of $x_{i1}(s)$ (Happ, 2013; Scheipl and Greven, 2016). This implies that care should be taken when interpreting coefficient functions with

low-rank covariates, while prediction is not affected. The issue can similarly occur for a main effect model including the second covariate $x_{i2}(t)$.
For an interaction effect model

$$g(\mu_i) \quad = \quad \eta_i = \beta_0 + \int\int x_{i1}(s)x_{i2}(t)\beta(s,t)dsdt, \tag{2.7}$$

results are analogue. Assume now $x_{i1}(s)$, $x_{i2}(t)$ to be realizations of two stochastic processes $X_1(s) \in L^2(\mathcal{S})$, $X_2(t) \in L^2(\mathcal{T})$ that are independent and square-integrable with zero mean each, defined on finite domains $\mathcal{S}$ and $\mathcal{T}$, respectively. $X_1(s)$ can be decomposed, analogously to above, in the Karhunen-Loève expansion $X_1(s) = \sum_{q_1=1}^{Q_1} \iota_{q_1}\chi_{q_1}(s)$. Then, $\{\chi_{q_1}(s), q_1 \in \mathbb{N}\}$ is an orthonormal basis of the space spanned by the eigenfunctions with non-zero eigenvalues of the covariance operator V_{X_1}. If $Q_1 < \infty$, this basis can be completed to form an orthonormal basis of $L^2(\mathcal{S})$. For the uncorrelated random variables ι_{q_1}, the expected value again is $\mathbb{E}(\iota_{q_1}) = 0 \; \forall \; q_1 \in \mathbb{N}$. Assume analogous definitions for $X_2(t)$. Further, let a function $o(s) \in L^2(\mathcal{S})$ be expanded by $o(s) = \sum_{q_s=1}^{\infty} \kappa_{q_s}\chi_{q_s}(s)$, with $\kappa_{q_s} \in \mathbb{R}$, $q_s \in \mathbb{N}$. Analogously, $o(t) \in L^2(\mathcal{T})$. Then, one can use a tensor product basis to expand the bivariate function $o(s,t) \in L^2(\mathcal{S}) \times L^2(\mathcal{T})$ by assuming κ_{q_s} to vary smoothly over $t \in \mathcal{T}$,

$$\begin{aligned}
o(s) \quad &= \quad \sum_{q_s=1}^{\infty} \kappa_{q_s}\chi_{q_s}(s) \text{ and } o(t) = \sum_{q_t=1}^{\infty} \kappa_{q_t}\chi_{q_t}(t), \text{ thus} \\
\kappa_{q_s}(t) \quad &= \quad \sum_{q_t=1}^{\infty} \kappa_{q_s q_t}\chi_{q_t}(t), \text{ concluding} \\
o(s,t) \quad &= \quad \sum_{q_s=1}^{\infty}\sum_{q_t=1}^{\infty} \kappa_{q_s q_t}\chi_{q_s}(s)\chi_{q_t}(t).
\end{aligned}$$

With these assumptions, it is

$$\begin{aligned}
\int_{\mathcal{T}}\int_{\mathcal{S}} X_1(s)X_2(t)o(s,t)dsdt \quad &= \quad \int_{\mathcal{T}}\int_{\mathcal{S}} \sum_{q_1=1}^{Q_1} \iota_{q_1}\chi_{q_1}(s) \sum_{q_2=1}^{Q_2} \iota_{q_2}\chi_{q_2}(t) \cdot \\
&\qquad \sum_{q_s=1}^{\infty}\sum_{q_t=1}^{\infty} \kappa_{q_s q_t}\chi_{q_s}(s)\chi_{q_t}(t)dsdt \\
&= \quad \sum_{q_1=1}^{Q_1}\sum_{q_2=1}^{Q_2} \iota_{q_1}\iota_{q_2}\kappa_{q_1 q_2}.
\end{aligned}$$

For detailed derivations of this and the following equations in this sections, please refer to Appendix A.2.

Since for the processes $X_1(s)$ and $X_2(t)$ it is $\mathbb{E}(\iota_{q_1}) = 0$, $\mathbb{E}(\iota_{q_2}) = 0$, and $\text{var}(\iota_{q_1}) =: \nu_{q_1} \neq 0$, $\text{var}(\iota_{q_2}) =: \nu_{q_2} \neq 0 \; \forall \; q_1, \, q_2 \in \mathbb{N}$ by definition, it is

$$\begin{aligned} \int_{\mathcal{T}} \int_{\mathcal{S}} X_1(s) X_2(t) o(s,t) ds dt &= 0 \\ \Leftrightarrow \kappa_{q_1 q_2} &= 0 \; \forall \; \big(q_1 \leq Q_1 \vee q_2 \leq Q_2\big). \end{aligned} \tag{2.8}$$

At the same time, the function $o(s,t)$ can also be written as

$$\begin{aligned} o(s,t) &= \sum_{q_s=1}^{\infty} \sum_{q_t=1}^{\infty} \kappa_{q_s q_t} \chi_{q_s}(s) \chi_{q_t}(t) \\ &\quad \textit{see Appendix A.2} \\ &= \sum_{q_s=1}^{Q_1} \sum_{q_t=1}^{Q_2} \kappa_{q_s q_t} \chi_{q_s}(s) \chi_{q_t}(t) + \sum_{q_s=Q_1+1}^{\infty} \sum_{q_t=1}^{Q_2} \kappa_{q_s q_t} \chi_{q_s}(s) \chi_{q_t}(t) + \\ &\quad \sum_{q_s=1}^{Q_1} \sum_{q_t=Q_2+1}^{\infty} \kappa_{q_s q_t} \chi_{q_s}(s) \chi_{q_t}(t) + \sum_{q_s=Q_1+1}^{\infty} \sum_{q_t=Q_2+1}^{\infty} \kappa_{q_s q_t} \chi_{q_s}(s) \chi_{q_t}(t). \end{aligned} \tag{2.9}$$

Now define $\beta(s,t) \in L^2(\mathcal{S}) \times L^2(\mathcal{T})$ in Model (2.7) to be identifiable if

$$\int_{\mathcal{T}} \int_{\mathcal{S}} X_1(s) X_2(t) o(s,t) ds dt = 0 \Leftrightarrow o(s,t) \equiv 0 \; \forall \; s \in \mathcal{S}, t \in \mathcal{T}. \tag{2.10}$$

Combining Equations (2.8) and (2.9), and using the rule for sums, $\sum_{i=m}^{n} a_i = 0 \Leftrightarrow m > n$, it follows that $\beta(s,t)$ is identifiable if both Q_1 and Q_2 are infinite.
Else if we assume, without loss of generality, $Q_2 < \infty$ (or both Q_1, $Q_2 < \infty$), and Equation (2.8) is taken to be true, i.e. $\kappa_{q_1 q_2} = 0 \; \forall \; q_1 \leq Q_1, q_2 \leq Q_2$. Then, from (2.9), it follows that for all functions $o(s,t)$ but the null-function it is
$o(s,t) = \sum_{q_s=1}^{\infty} \sum_{q_t=Q_2+1}^{\infty} \kappa_{q_s q_t} \chi_{q_s}(s) \chi_{q_t}(t) \neq 0$
$\Big($or $o(s,t) = \sum_{q_s=Q_1+1}^{\infty} \sum_{q_t=Q_2+1}^{\infty} \kappa_{q_s q_t} \chi_{q_s}(s) \chi_{q_t}(t) \neq 0$, respectively$\Big)$.
This conclusion is equivalent to the proposition that the kernel of the covariance of $X_1(s)X_2(t)$ is empty if $\beta(s,t)$ is identifiable, as we will show in the following.

The covariance operator of $X_1(s)$, applied to a function $o(s) \in L^2(\mathcal{S})$, is defined by

$$\begin{aligned} (V_{X_1} o)(s) &= \int_{\mathcal{T}} \mathbb{E}\left\{ \left[X_1(s) - \mathbb{E}(X_1(s)) \right] \left[X_1(t) - \mathbb{E}(X_1(t)) \right] \right\} o(t) dt \\ &\quad \textit{by definition, } \mathbb{E}(X_1(s)) = 0 \\ &= \int_{\mathcal{T}} \mathbb{E}\left\{ X_1(s) X_1(t) \right\} o(t) dt. \end{aligned}$$

Let us define a new covariate function $X(s,t) \in L^2(\mathcal{S}) \times L^2(\mathcal{T})$ by

$$X(s,t) := X_1(s)X_2(t).$$

Then for $X(s,t)$, it is $\mathbb{E}(X(s,t)) = \mathbb{E}(X_1(s)X_2(t)) = 0$, since $X_1(s)$, $X_2(t)$ are assumed to be independent. Now let function $o(s,t) \in L^2(\mathcal{S}) \times L^2(\mathcal{T})$ be expanded as above. For the covariance operator of $X(s,t)$ applied to $o(s,t)$, with $u \in \mathcal{S}$, $v \in \mathcal{T}$, it follows

$$\begin{aligned}(V_X o)(s,t) &= \int_{\mathcal{T}}\int_{\mathcal{S}} \mathbb{E}\left\{X(s,t)X(u,v)\right\} o(u,v)dudv \\ &\quad \textit{see Appendix A.2} \\ &= \sum_{q_s=1}^{Q_s}\sum_{q_t=1}^{Q_t} \nu_{q_s}\nu_{q_t}\kappa_{q_sq_t}\chi_{q_s}(s)\chi_{q_t}(t).\end{aligned}$$

Thus, function $o(s,t)$ lies in the kernel of V_X

$$\begin{aligned}Ker(V_X) &= \text{span}\{\chi_{q_s}(s), q_s \le Q_s\}^{\perp} \circledcirc \text{span}\{\chi_{q_t}(t), q_t \le Q_t\}^{\perp} \\ &:= \text{span}\{\chi_1(s)\chi_1(t), \chi_2(s)\chi_1(t), \dots \chi_{Q_s}(s)\chi_1(t), \chi_1(s)\chi_2(t) \dots \chi_{Q_s}(s)\chi_{Q_t}(t)\}^{\perp}\end{aligned}$$

if

$$\begin{aligned}o(s,t) \in Ker(V_X) &\Leftrightarrow (V_X o)(s,t) = 0 \\ &\Leftrightarrow \kappa_{q_sq_t} = 0 \;\forall\; (q_s \le Q_s \vee q_t \le Q_t)\,.\end{aligned}$$

Since $Ker(V_X) \neq \{\emptyset\} \Leftrightarrow Q_s \wedge Q_t < \infty$, $\beta(s,t)$ is identifiable only up to the functions $o(s,t) \in Ker(V_X)$.

This theoretical result can be transferred to real data situations, where the stochastic processes $X_1(s)$, $X_2(t)$ become N observed curves $x_{i1}(s)$, $x_{i2}(t)$, $i = 1, \dots N$, on finite grids $s \in \{s_1, \dots s_J\} \subset \mathcal{S}$, $t \in \{t_1, \dots t_K\} \subset \mathcal{T}$, typically with $\mathcal{S}$, $\mathcal{T} \subseteq \mathbb{R}$. Analogously to the previous sections, the integrals of Model (2.7) can be approximated via quadrature sums. Using the Karhunen-Loève expansion for $x_{i1}(s)$ and $x_{i2}(t)$, and representing the coefficient function $\beta(s,t)$ in a tensor product of univariate spline bases yields

$$\begin{aligned}\eta_i &= \beta_0 + \int\int x_{i1}(s)x_{i2}(t)\beta(s,t)dsdt \\ &\quad \textit{assume equidistant grids for both domains} \\ &\approx \beta_0 + h_1h_2 \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{q_1=1}^{Q_1} \iota_{i1q_1}\chi_{1q_1}(s_j) \sum_{q_2=1}^{Q_2} \iota_{i2q_2}\chi_{2q_2}(t_k) \sum_{l=1}^{L}\sum_{m=1}^{M} c_{lm}\phi_{3l}(s_j)\phi_{4m}(t_k) \\ &=: \underset{1\times 1}{\beta_0} + \underset{1\times Q_1}{\boldsymbol{\Xi}_1^T}\underset{Q_1\times J}{\boldsymbol{\chi}_1^T}\underset{J\times J}{\boldsymbol{W}_1}\underset{J\times L}{\boldsymbol{\Phi}_3}\underset{L\times M}{\boldsymbol{C}}\underset{M\times K}{\boldsymbol{\Phi}_4^T}\underset{K\times K}{\boldsymbol{W}_2^T}\underset{K\times Q_2}{\boldsymbol{\chi}_2}\underset{Q_2\times 1}{\boldsymbol{\Xi}_2} \\ &\quad \textit{vectorize, } vec(ABC) = (C^T \otimes A)vec(B) \\ &= \underset{1\times 1}{\beta_0} + (\underset{1\times Q_2}{\boldsymbol{\Xi}_2^T}\underset{Q_2\times K}{\boldsymbol{\chi}_2^T}\underset{K\times K}{\boldsymbol{W}_2}\underset{K\times M}{\boldsymbol{\Phi}_4} \otimes \underset{1\times Q_1}{\boldsymbol{\Xi}_1^T}\underset{Q_1\times J}{\boldsymbol{\chi}_1^T}\underset{J\times J}{\boldsymbol{W}_1}\underset{J\times L}{\boldsymbol{\Phi}_3})\underset{LM\times 1}{\text{vec}(\boldsymbol{C})}. \qquad (2.11)\end{aligned}$$

Here, h_1, h_2 are the lengths of the intervals between two observation points in the respective domains $\mathcal{S}$, $\mathcal{T}$. The matrices are defined by

$$\begin{array}{ll}
\boldsymbol{\Xi}_1 = (\iota_{i11}, \dots \iota_{i1Q_1})^T & \boldsymbol{\Xi}_2 = (\iota_{i21}, \dots \iota_{i2Q_2})^T \\
\mathbb{R}^{J\times Q_1} \ni \boldsymbol{\chi}_1 = (\chi_{1q_1}(s_j))_{j=1,\dots J;q_1=1,\dots Q_1} & \mathbb{R}^{K\times Q_2} \ni \boldsymbol{\chi}_2 = (\chi_{2q_2}(t_k))_{k=1,\dots K;q_2=1,\dots Q_2} \\
\mathbb{R}^{J\times J} \ni \boldsymbol{W}_1 = \operatorname{diag}(h_1, \dots h_1)_{j=1,\dots J;j=1,\dots J} & \mathbb{R}^{K\times K} \ni \boldsymbol{W}_2 = \operatorname{diag}(h_2, \dots h_2)_{k=1,\dots K;k=1,\dots K} \\
\mathbb{R}^{J\times L} \ni \boldsymbol{\Phi}_3 = (\phi_{3l}(s_j))_{j=1,\dots J;l=1,\dots L} & \mathbb{R}^{K\times M} \ni \boldsymbol{\Phi}_4 = (\phi_{4m}(t_k))_{k=1,\dots K;m=1,\dots M} \\
\mathbb{R}^{L\times M} \ni \boldsymbol{C} = (c_{lm})_{l=1,\dots L;m=1,\dots M} &
\end{array}$$

When examining all N observations instead of observation i, Equation (2.11) becomes more complicated, including a Hadamard-Schur product, which for arbitrary matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C} \in \mathbb{R}^{n\times m}$ defines an elementwise matrix multiplication $\boldsymbol{C} = (\boldsymbol{A} \circ \boldsymbol{B}) = (a_{nm}b_{nm})$. The interaction model for finite solution data thus becomes

$$\begin{aligned}
\underset{N\times 1}{\boldsymbol{\eta}} &= \underset{N\times 1}{\boldsymbol{\beta_0}} + (\underset{N\times Q_2}{\boldsymbol{\Xi}_2}\ \underset{Q_2\times K}{\boldsymbol{\chi}_2^T}\ \underset{K\times K}{\boldsymbol{W}_2}\ \underset{K\times M}{\boldsymbol{\Phi}_4} \otimes \underset{1\times L}{\boldsymbol{I}_1}) \circ (\underset{1\times M}{\boldsymbol{I}_2} \otimes \underset{N\times Q_1}{\boldsymbol{\Xi}_1}\ \underset{Q_1\times J}{\boldsymbol{\chi}_1^T}\ \underset{J\times J}{\boldsymbol{W}_1}\ \underset{J\times L}{\boldsymbol{\Phi}_3})\underset{LM\times 1}{\operatorname{vec}(\boldsymbol{C})} \\
&=: \boldsymbol{\beta_0} + (\boldsymbol{B} \circ \boldsymbol{A})\boldsymbol{\Theta} \\
&=: \boldsymbol{\beta_0} + \boldsymbol{D}\boldsymbol{\Theta}, \qquad (2.12)
\end{aligned}$$

with

$$\begin{array}{ll}
\mathbb{R}^{N\times LM} \ni \boldsymbol{D} = \boldsymbol{B} \circ \boldsymbol{A} & \boldsymbol{\Theta} = \operatorname{vec}(\boldsymbol{C}) \\
\mathbb{R}^{N\times LM} \ni \boldsymbol{A} = \boldsymbol{I}_2 \otimes \boldsymbol{\Xi}_1\boldsymbol{\chi}_1^T\boldsymbol{W}_1\boldsymbol{\Phi}_3 & \mathbb{R}^{N\times LM} \ni \boldsymbol{B} = \boldsymbol{\Xi}_2\boldsymbol{\chi}_2^T\boldsymbol{W}_2\boldsymbol{\Phi}_4 \otimes \boldsymbol{I}_1 \\
\boldsymbol{I}_1 = (1, \dots 1)^T \in \mathbb{R}^L & \boldsymbol{I}_2 = (1, \dots 1)^T \in \mathbb{R}^M \\
\mathbb{R}^{N\times Q_1} \ni \boldsymbol{\Xi}_1 = (\iota_{i1q_1})_{i=1,\dots N;q_1=1,\dots Q_1} & \mathbb{R}^{N\times Q_2} \ni \boldsymbol{\Xi}_2 = (\iota_{i2q_2})_{i=1,\dots N;q_2=1,\dots Q_2}
\end{array}$$

and else as above.
The finite data analogon to the identifiability definition (2.10) is

$$\boldsymbol{D}\boldsymbol{\Theta} = 0 \ \Leftrightarrow\ \boldsymbol{\Theta} \equiv 0,$$

which is equivalent to $Ker(\boldsymbol{D}) = \{0\}$. Considering a response following an exponential family distribution,

$$\boldsymbol{Y} \sim EF(\mu, \eta),$$

the uniqueness of the corresponding solution of maximizing the log-likelihood function at least requires $\boldsymbol{D}$ to be of full column rank, i.e. rank($\boldsymbol{D}$)=LM, or $Ker(\boldsymbol{D}) = \{0\}$ (see Fahrmeir et al., 2009; Happ, 2013; Scheipl and Greven, 2016). Since the resulting Hadamard-Schur product of two matrices is part of the Kronecker product of those matrices, the rank fulfills

$$\begin{aligned}
\operatorname{rank}(\boldsymbol{D}) &= \operatorname{rank}(\boldsymbol{B} \circ \boldsymbol{A}) \\
&\leq \operatorname{rank}(\boldsymbol{B} \otimes \boldsymbol{A}) \\
&= \operatorname{rank}(\boldsymbol{B})\operatorname{rank}(\boldsymbol{A}), \qquad (2.13)
\end{aligned}$$

see Styan (1973). For the rank of $\boldsymbol{A}$, it is

$$\begin{aligned}
\text{rank}(\boldsymbol{A}) &= \text{rank}(\boldsymbol{I}_2 \otimes \boldsymbol{\Xi}_1 \boldsymbol{\chi}_1^T \boldsymbol{W}_1 \boldsymbol{\Phi}_3) \\
&= \text{rank}(\boldsymbol{I}_2)\text{rank}(\boldsymbol{\Xi}_1 \boldsymbol{\chi}_1^T \boldsymbol{W}_1 \boldsymbol{\Phi}_3) \\
&= \text{rank}(\boldsymbol{\Xi}_1 \boldsymbol{\chi}_1^T \boldsymbol{W}_1 \boldsymbol{\Phi}_3) \\
&\quad \textit{by construction, } \boldsymbol{\Xi}_1 \textit{ and } \boldsymbol{W}_1 \textit{ are of full column rank; then} \\
&\quad \textit{rank}(\boldsymbol{\Xi}_1 \boldsymbol{\chi}_1^T \boldsymbol{W}_1 \boldsymbol{\Phi}_3) = \textit{rank}(\boldsymbol{\chi}_1^T \boldsymbol{\Phi}_3) \textit{ follows from Harville (2000), Lemma 8.3.2} \\
&= \text{rank}(\boldsymbol{\chi}_1^T \boldsymbol{\Phi}_3) \\
&\leq \min(Q_1, J, L) \\
&\quad \textit{since rank}(\boldsymbol{X}_1) = \textit{rank}(\boldsymbol{\Xi}_1 \boldsymbol{\chi}_1^T \boldsymbol{W}_1) \textit{ it is min}(N, J) = \textit{min}(Q_1, J), \\
&\quad \textit{and with rank}(\boldsymbol{\Xi}_1) = Q_1 \leq N \textit{ it follows } Q_1 \leq \textit{min}(N, J) \\
&\leq \min(Q_1, L).
\end{aligned}$$

Analogously, $\text{rank}(\boldsymbol{B}) \leq \min(Q_2, M)$.

With the results for $\text{rank}(\boldsymbol{A})$ and $\text{rank}(\boldsymbol{B})$, we can conclude that

$$\text{rank}(\boldsymbol{D}) \leq \min(Q_2, M) \cdot \min(Q_1, L).$$

This means that the identifiability of Model (2.12) depends on both covariates through Q_1 and Q_2.

In conclusion, the above discussion shows that the uniqueness of an estimate $\hat{\beta}(s,t)$ of a scalar-on-functions regression model (2.7) with covariate interaction effect depends on the ranks of both functional covariates. It remains to be shown that if $Q_1 < L$ or $Q_2 < M$ holds, the model matrix $\boldsymbol{D}$ is not of full column rank and Model (2.7) is not identifiable anymore. Thus, care has to be taken when defining a spline basis for the coefficient function $\beta(s,t)$, especially in data situations with functional covariates of low ranks Q_1, Q_2.
In fact, the estimation process used for Model (2.3) uses a penalized maximum likelihood. The respective penalty term also influences the identifiability issue. The results in Happ (2013) and Scheipl and Greven (2016) indicate that the penalty slighty attenuates the problem, probably in such a way that the coefficient function estimate might be identifiable unless it is an element of the overlap of the covariates covariance's kernel and the kernel of the penalty. Furthermore, if a model includes more than one term containing the same covariate (as is the case in Model (2.3)), additional multicollinearity can occur in functional regression models analogously to multivariate statistics. It can be expected that the identifiability issue is exacerbated in the presence of multicollinearity. Also, presumably, penalization and multicollinearity affect each other, and the theoretical examination thereof as well as the impact on the estimation performance in a real data situation would be worth examining in detail.

2.8 Covariate Interaction of Higher Orders

The Model (2.3) can be extended to higher-order interactions, such as a three-way interaction

$$\begin{aligned} g(\mu_i) &= \beta_0 + \int x_{i1}(s)\xi_1(s)ds + \int x_{i2}(t)\xi_2(t)dt + \int x_{i3}(u)\xi_3(u)du + \\ &\quad \int\int x_{i1}(s)x_{i2}(t)\beta_1(s,t)dsdt + \\ &\quad \int\int x_{i1}(s)x_{i3}(u)\beta_2(s,u)dsdu + \\ &\quad \int\int x_{i2}(t)x_{i3}(u)\beta_3(t,u)dtdu + \\ &\quad \int\int\int x_{i1}(s)x_{i2}(t)x_{i3}(u)\beta(s,t,u)dsdtdu. \end{aligned} \tag{2.14}$$

The implementation is analogous to the two-way interaction effect. The interpretation and visualization of especially the three-dimensional interaction surface $\beta(s,t,u)$ becomes more challenging and sample sizes likely need to be large to estimate the interaction terms well. Since the cell chip data examined in Section 2.6 offers three signal types, Model (2.14) will in the following be applied to it as an exemplarily data set. Here, the ISFET-signals are taken to be $x_{i1}(s)$, the IDES-signals $x_{i2}(t)$, and the CLARK-signals $x_{i3}(u)$.

2.8.1 Functional Linear Model with Second Order Covariate Interaction Applied to Cell Based Sensor Chips

To compare the prediction results of Models (2.3) and (2.14), the cell chip data is used, with again $n = 280$ measurements per signal type, being smoothed by B-splines (see also Figure 2.15, lower panels). The number of basis functions is four for each marginal basis. For each of 25 replications, a number of $n_{val} = 56$ observations was drawn randomly from the data as validation data set, the remaining $n_{cal} = 224$ observations are used to fit the models.

Figure 2.19 compares the adjusted R^2 and the relative mean squared error of prediction (cf. Section 2.5.1) for the AAP concentrations across all replications of the validation data for different models. The first six models are the same that were already compared in Section 2.6.1, with results being analogous to the respective section, and the sixth boxes presenting the full two-way interaction Model (2.3). The seventh boxes yield the results for the three-way interaction Model (2.14). Both, the adjusted R^2 as well as the relMSE$_{\hat{y}}$ are very similar to the two-way interaction model. Results from Section 2.6.1 indicated that the two-way interaction effect of the ISFET- and CLARK-signals contributes at most slightly to prediction. It thus shows that the inclusion of higher order interaction terms that use data of low rank (as is the IDES data) will not improve the prediction results of a model.

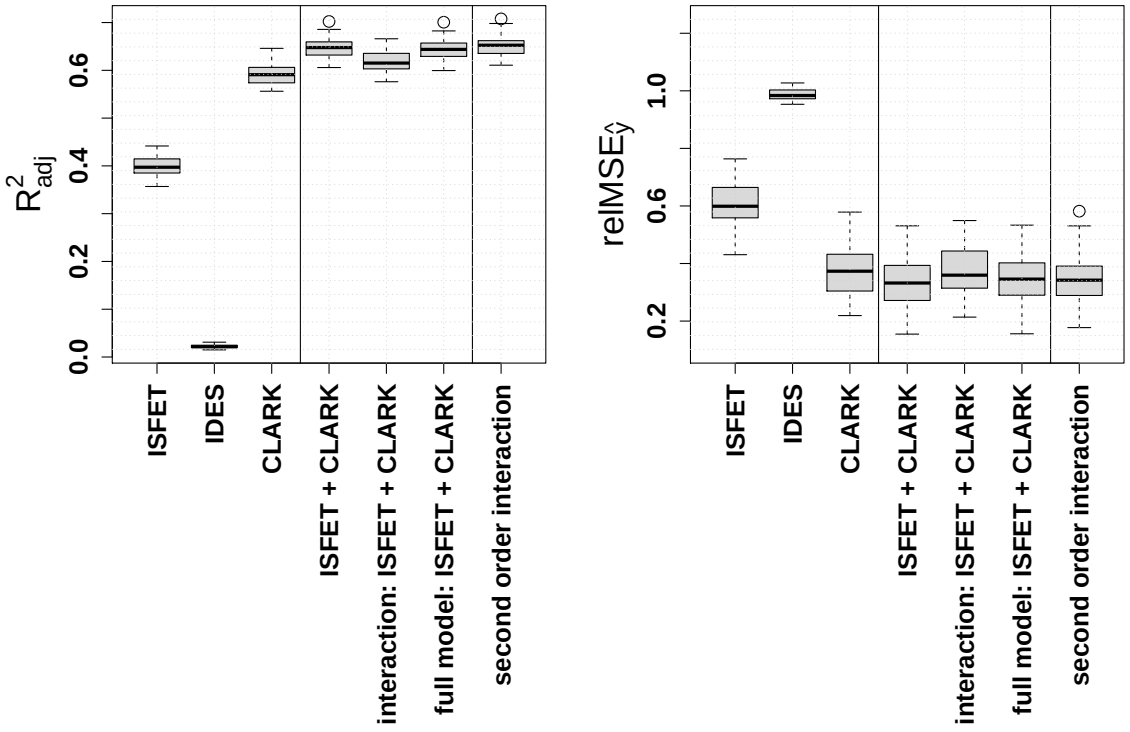


Figure 2.19: R^2_{adj} (left) and relative mean squared error of prediction in the validation data (right) for different models including up to three main effects and up to one three-way interaction effect. The boxplots show variation in both quantities across 25 splits into calibration and validation data.

The estimates of the three-way interaction model (2.14) applied to the full data, being typical also for the single replicates, are shown in Appendix A.3. For ISFET-signals, the almost linearly, decreasing main effect estimate $\hat{\xi}_1(s)$ implies that the higher values at the beginning of a measurement are, the higher the AAP concentration should be. The same holds for the CLARK based estimate $\hat{\xi}_3(u)$. Estimate $\hat{\xi}_2(t)$ forms a slightly curved arch, increasing until about 275 minutes, and decreasing afterwards. Thus, high IDES-signal values around 275 minutes should correspond to the highest amounts of AAP. The first interaction surface estimate $\hat{\beta}_1(s,t)$ using the ISFET- and IDES-signals is nearly constant and negative. Thus, ISFET- and IDES-signals with high values and similar curve progression correspond to lower AAP values than expected from the main effects. The second interaction surface estimate $\hat{\beta}_2(s,u)$, based on the ISFET- and CLARK-signals, is also nearly constant, but positive, such that ISFET- and CLARK-signals with high values and similar curve progression correspond to higher AAP values. The last two-way interaction surface estimate $\hat{\beta}_3(t,u)$ is a nearly linearly decreasing in the u-direction until about 300 minutes, increasing slightly afterwards, and becoming negative around 225 minutes. In

the t-direction, it is approximately constant. This implies that CLARK curves with high values in the beginning and low values especially around 300 minutes correspond to the highest amounts of AAP.
To interpret the three-way interaction effect shape $\hat{\beta}(s,t,u)$, we fix the value of s at the respective observed points $s_z \in \{s_1, \dots s_{36}\}$. For every s_z, the estimated surface $\hat{\beta}(s_z,t,u)$ is a plane, tilted downwards to decreasing values of u crossing zero between 270 - 380 minutes (dependings on z), and being constant in t. Its absolute level increases roughly linearly with increasing s_z. This represents those CLARK-signals with low values at the beginning and high values at the end of a measurement that correspond to high AAP concentrations.

Note that, compared to the number of coefficients that have to be estimated, the number of observations $n_{cal} = 224$ used for modeling is quite small. Additionally, the results concerning identifiability found in Section 2.7 will probably exacerbate in models including several or higher order interaction terms. Thus, the above interpretation of the coefficient functions should be affirmed carefully.

2.9 Perspectives

Our proposed scalar-on-functions regression with interaction term can offer relevant improvement concerning the predictive power and part of explained variance of real world data compared to functional models without interaction term. In addition, it can offer enhanced insights into the underlying structure of covariate effects. Interaction effects between covariates can be detected and our model can thus also be used to check additivity assumptions in scalar-on-functions regression models. In cases with low information content in the covariates, coefficient estimates have to be interpreted with care due to possible identifiability issues, but predictive performance of the model is not affected.
Our simulation studies show that reliable estimates of covariate interactions can be achieved on relatively small data sets in the Gaussian response case, while binary responses require much larger data sets. Prediction of an outcome is reliable for small data sets in both cases. Simulations and the spectra application indicate that including a functional interaction term can improve fit and prediction for data where an interaction is truly present.
We found that data preprocessing has a non-negligible influence on the estimated coefficient functions as well as on the model outcome. The amount of this influence depends on the data at hand. It is advisable to do a careful comparison if different preprocessings are meant to be applied to a particular data set.
The proposed model can be extended to higher-order interactions, such as three-way interactions. The number of coefficients to be estimated and thus the computational effort naturally increase. For example, on a machine with four AMD Opteron CPUs of 12 cores and 2.2 GHz RAM, with software R version 3.0.1 (2013-05-16) and the add-on package `mgcv` version 1.7-28, the estimation of Model (2.3) for the cell chip data takes only sec-

onds, while a three-way interaction model takes about a quarter of an hour to fit. Further research is needed to make the information contained in higher-order interactions usable and the implementation of corresponding models more efficient.

It is to be expected that data with multiple functional variables will become more and more frequent due to advances in engineering and other fields in the next decades. Thus, ample room remains for further development of flexible regression models with multiple functional covariates and corresponding inference.

Chapter 3

Nearest Neighbor Ensembles for Functional Data with Interpretable Feature Selection

3.1 Nearest Neighbors and Ensemble Methods

In this chapter, the classification of functional covariates is achieved by combining posterior probabilities – calculated from semi-metrics, a specific number of nearest neighbors, and the leave-one-out technique – to an ensemble. There are two main types of ensembles in data analysis. The first uses the results of various models as ensemble members. One of the first approaches setting up such an ensemble for functional data is the method by Goldsmith and Scheipl (2014). Here, scalar-on-function regression is done by building an ensemble, for example a linear model, with the fits of many candidate estimators constituting the ensemble members. The ensemble introduced in this chapter, however, does not deal with regression but classification problems. The far more common type of ensembles, especially widespread in the machine learning community, uses scores or features of some kind as ensemble members and a linear combination as the ensemble. Examples for such ensembles can be found, for example, in Athitsos and Sclaroff (2005), Preda et al. (2007), or Araki et al. (2009). Alonso et al. (2012) use a pre-defined semi-metric on (derivatives of) functional observations to generate multivariate data points that then are used as input in a classification algorithm.

Nearest neighbors are discrimination techniques belonging to the group of supervised learning methods. Originally proposed by Fix and Hodges (1951), they have become popular in chemometrics, see, e.g., Alonso-Salces et al. (2005), Japon-Lujan et al. (2006), Lukasiak et al. (2007), Kruzlicova et al. (2008), Fdez-Ortiz de Vallejuelo et al. (2011), or Berrueta et al. (2007); but have also been used in other fields of application (see for example Melvin et al., 2008; Wong et al., 2010; Przewozniczek et al., 2011; Nava et al., 2014). Nearest

neighbor approaches are nonparametric and memory based (see also Hastie et al., 2011). For multivariate data, the basic principle of k-nearest-neighbors is as follows:
Let a learning sample be given by $(y_i, \boldsymbol{x}_i)$, $i = 1, \dots N$, where $\boldsymbol{x}_i = (x_{i1}, \dots, x_{iq})^T$ is a vector of predictors and $y_i \in \{1, \dots G\}$ denotes the class membership of observation i. Moreover, let $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ denote a distance measure in the feature space. For a new observation $\boldsymbol{x^*} = (x_1^*, \dots, x_q^*)^T$, one determines the k observations which are closest to $\boldsymbol{x^*}$. This means one seeks the k-nearest-neighbors of $\boldsymbol{x^*}$, denoted by $\boldsymbol{x}_{(1)}, \dots \boldsymbol{x}_{(k)}$, with nearness defined by the distance measure $d(\cdot, \cdot)$. The k nearest neighbors fulfill

$$d(\boldsymbol{x^*}, \boldsymbol{x}_{(1)}) \leq \dots \leq d(\boldsymbol{x^*}, \boldsymbol{x}_{(k)}).$$

Let $y_{(i)}$ denote the observed class linked to the neighbor $x_{(i)}$. For the assignment of $\boldsymbol{x^*}$ to a class y^*, one uses the majority rule,

$$y^* = g \Leftrightarrow g \text{ is most frequent in observations } \{y_{(1)}, \dots y_{(k)}\}.$$

The resulting classifier is called the $\underline{k}$-$\underline{n}$earest-$\underline{n}$eighbor (kNN) classifier.
The basic k-nearest-neighbor approach can be modified in various ways. For example, Gertheiss and Tutz (2009) extended it to an ensemble of weighted nearest neighbor posterior probability estimates. Tutz and Koch (2016) include the labels of different types of nearest neighbors as covariates in a (multinomial) logit model. Further extensions can be found in Ji and Zhao (2013) and Hayat et al. (2014); see also Bischl et al. (2013) for a comparison of the k-nearest-neighbor approach with other local discrimination techniques. In the present chapter we introduce a k-nearest-neighbor classification ensemble for functional data. The general ensemble methodology used here is related to "model stacking" and "super learning" (see, e.g., Wolpert, 1992; LeBlanc and Tibshirani, 1996; van der Laan and Dudoit, 2003).

The two data sets motivating our approach are the measurements of cell based sensor chips and gas sensor data (see also Chapter 1 and Appendix D). For the temperature-cycle operated gas sensors, depicted in Figure 3.1, the classification task implies the discrimination of seven gas species. Here, physicochemical considerations lead to the assumption that especially the jumps in the signals contain information about the curves' classes (see also Section 3.5).
The second data set refers to the cell chip sensor measurements used in this classification approach, depicted in Figure 3.7. In the experiment two classes are considered, one class with 2.5mM paracetamol (short: AAP) applied to the cells, and one class without the use of paracetamol. One approach to two-class discrimination problems with functional covariates is to use a logistic functional model (Müller and Stadtmüller, 2005). Such generalized functional linear models use the whole observation, i.e., the curve $x_i(t)$ across the entire domain $\mathbb{D}$. In many applications, however, it is reasonable to assume that only parts of the signal contain discriminative information. For example, biological considerations

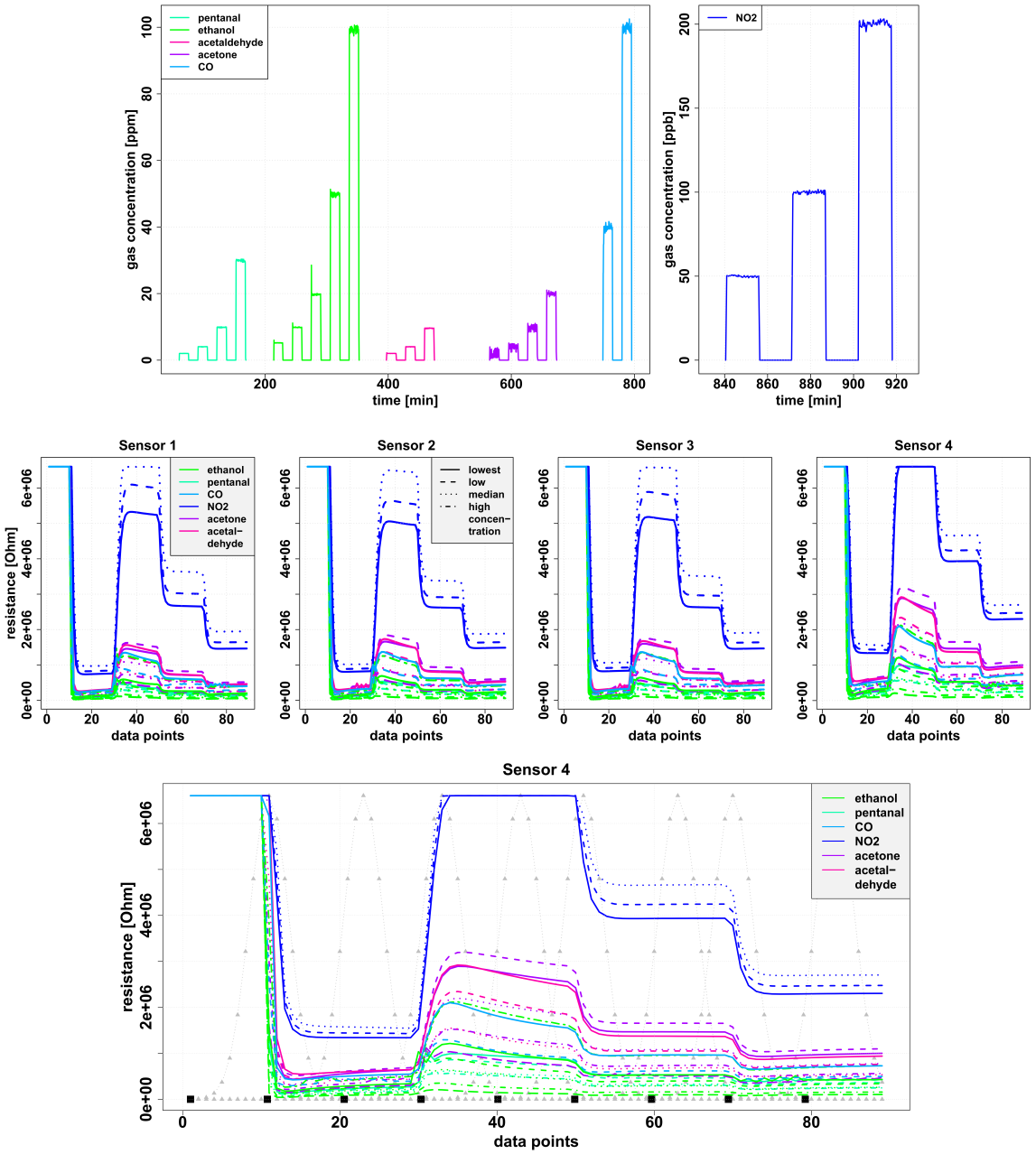


Figure 3.1: The upper panel shows the gas species and concentrations applied to the gas sensors over time. Pure synthetic air is applied where no other gases are shown. The second row of panels shows the mean gas sensor signals for each sensor per gas species and respective concentrations. The third panel again shows the mean signals of the fourth gas sensor, where the light gray, dotted lines depict the function $\phi_\tau(t)$ in semi-metric $d_{a\tau}^{Scan}(\cdot,\cdot)$, and the impact points t_q used in $d_a^{Points}(\cdot,\cdot)$ are depicted as black boxes (for further details on semi-metrics and their parameters, see Section 3.2.1).

concerning the cell chip data suggest that especially the range around 220 minutes is of importance (see also Section 3.4).

The k-nearest-neighbor ensemble approach presented here is especially designed to perform automated and interpretable feature selection on functional covariates. Ferraty and Vieu (2006) showed that the concept of nearness in functional data analysis is adequately met by so-called semi-metrics in the space of the functional predictors. The idea of our nearest neighbor ensemble is not to use a single semi-metric, but a set of semi-metrics, where each semi-metric focuses on a certain feature of the curve. For example, we use a semi-metric that focuses on the absolute distance of two curves on a limited range $\mathbb{D}_{small} \subset \mathbb{D}$ of the domain of definition $\mathbb{D}$ of the covariates, or one that focuses on jump heights at specific points from $\mathbb{D}$. The basic concept is to select from the set of potential semi-metrics the best ones and combine them in a smart and data-driven way: By assigning weights to the members of the ensemble, information on the discriminative power of different semi-metrics is obtained. The estimated weights reflect which signal parts, or which forms of data preprocessing, are most relevant for discrimination. Thus, the resulting k-nearest-neighbor ensemble allows for an automated and interpretable selection of curve features.
The remainder of this chapter is organized as follows. In Section 3.2, the semi-metrics and the functional k-nearest-neighbor ensemble are introduced in detail. In Section 3.3, our approach is evaluated by means of simulation studies and compared to alternative classification methods. All classification approaches are applied to the cell chip, the gas sensor, and the phoneme data in Sections 3.4 and 3.6. The chapter ends with a discussion of further developments. In the online supplement of Fuchs et al. (2015a), we provide the cell chip data as well as code reproducing our results. An up-to-date implementation of the functional k-nearest-neighbor ensemble can be found in the R-package `classiFunc` (Maierhofer and Fuchs, 2017), which was developed as part of Maierhofer (2017).

3.2 Construction of Functional Nearest Neighbor Ensembles

3.2.1 Distance Measures

Ferraty and Vieu (2006) postulate that a semi-metric d on space $\mathcal{F}$ fulfills $d(a,a) = 0 \wedge d(a,b) \leq d(a,e) + d(e,b)\ \forall a,b,e \in \mathcal{F}$. They point out that semi-metrics, if chosen appropriately, may override the curse of dimensionality by taking functional features of the functional observations into account. We put an additional constraint on our semi-metrics: they should also fulfill $d(a,b) = d(b,a)\ \forall a,b \in \mathcal{F}$. This ensures that the similarity of two curves $x_i(t)$ and $x_j(t)$ is based on curve characteristics and ignores, for example, the orientation of curve shifts, i.e., whether curve $x_i(t)$ lies above or beneath a curve $x_j(t)$ with identical shape. An important difference between metrics and semi-metrics lies in the implications of a distance $d = 0$. While $d(a,a) = 0$ holds for semi-metrics as well as for metrics, the property $d(a,b) = 0 \Leftrightarrow a \equiv b$ of a metric space does not necessarily hold

Semi-metric	Takes into account...
$d_{a,\mathbb{D}_{small}}^{shortEucl}(x_i(t), x_j(t)) = \sqrt{\int_{\mathbb{D}_{small}} \left(x_i^{(a)}(t) - x_j^{(a)}(t)\right)^2 dt}$	... the absolute distance on a limited part of the domain of definition $\mathbb{D}_{small} \subset \mathbb{D}$ of two curves (or their derivatives).
$d_a^{Mean}(x_i(t), x_j(t)) = \left\lvert \int_{\mathbb{D}} x_i^{(a)}(t)dt - \int_{\mathbb{D}} x_j^{(a)}(t)dt \right\rvert$	... the similarity of mean values of the whole curves (or their derivatives).
$d_a^{relAreas}(x_i(t), x_j(t)) = \left\lvert \left\lvert \frac{\int_{\mathbb{D}_1} x_i^{(a)}(t)dt}{\int_{\mathbb{D}_2} x_i^{(a)}(t)dt} \right\rvert - \left\lvert \frac{\int_{\mathbb{D}_1} x_j^{(a)}(t)dt}{\int_{\mathbb{D}_2} x_j^{(a)}(t)dt} \right\rvert \right\rvert$	... the similarity of the relation of areas on parts of the domain of definition $\mathbb{D}_1$, $\mathbb{D}_2 \subset \mathbb{D}$.
$d_{no}^{Jump}(x_i(t), x_j(t)) = \left\lvert (x_i(t_n) - x_i(t_o)) - (x_j(t_n) - x_j(t_o)) \right\rvert$	... the similarity of jump heights at points t_n, $t_o \in \mathbb{D}$.
$d_a^{Max}(x_i(t), x_j(t)) = \left\lvert \max\left(x_i^{(a)}(t)\right) - \max\left(x_j^{(a)}(t)\right) \right\rvert$	... the difference of the curves' (or their derivatives') global maxima.
$d_a^{Min}(x_i(t), x_j(t)) = \left\lvert \min\left(x_i^{(a)}(t)\right) - \min\left(x_j^{(a)}(t)\right) \right\rvert$	... the difference of the curves' (or their derivatives') global minima.
$d_a^{Points}(x_i(t), x_j(t)) = \frac{1}{S} \sum_{q=1}^{S} \left\lvert x_i^{(a)}(t) - x_j^{(a)}(t) \right\rvert_{t=t_q}$	... the differences at certain observation points (also called "points of impact").

Table 3.1: Further semi-metrics used in the k-nearest-neighbor ensemble.

for semi-metrics, such that $d(a, b) = 0$ can occur for $a \neq b$. In principle, every distance measure operating on curves $x_i(t)$ and fulfilling the above equations is allowed in our approach. Nonetheless, the semi-metrics we will consider are supposed to account for specific characteristics of the functional covariates.

In what follows, let $x_i^{(a)}(t)$ denote the ath order differentiation of $x_i(t)$. We restrict the set of semi-metrics we use to semi-metrics that focus on specific curve characteristics. For example, a measure that focuses on the curve distances is the Euclidian distance

$$d_a^{Eucl}(x_i(t), x_j(t)) = \sqrt{\int_{\mathbb{D}} \left(x_i^{(a)}(t) - x_j^{(a)}(t)\right)^2 dt}.$$

It represents the absolute distance of two curves, or their derivatives, which might contain information concerning the class, for example, if the curves have similar shapes within classes. Instead of such a "static" semi-metric, especially appealing distance measures are adaptive ones that locate points or regions of discriminative power. An example for such a more sophisticated semi-metric is

$$d_{a\tau}^{Scan}(x_i(t), x_j(t)) = \sqrt{\int_{\mathbb{D}} \left(\phi_\tau(t)\left(x_i^{(a)}(t) - x_j^{(a)}(t)\right)\right)^2 dt},$$

with $\tau \in \mathbb{D}$, where the function $\phi_\tau(t)$ is an appropriate scan function, for example a Gaussian kernel

$$\phi_\tau(t) = \frac{1}{\sqrt{2\pi}} \exp^{-\frac{(t-\tau)^2}{2}},$$

giving a weight profile to the variable t that is centered around τ. Further semi-metrics used in our k-nearest-neighbor ensemble are given in Table 3.1. They were chosen because they seemed to be the most appropriate distance measures for the analyses of the application data described in Sections 3.4 and 3.5. Naturally, our approach can be extended to other semi-metrics. In applications, however, it might be difficult to judge which semi-metric is most appropriate for the discrimination task. Therefore our strategy is to combine several semi-metrics in an ensemble. Which semi-metric yields most information for discrimination is reflected in ensemble weights that are estimated using the data at hand.

3.2.2 The Functional Nearest Neighbor Ensemble

Let $(y_i, x_i(t))$, $i = 1, \dots N$, be a learning sample and $(y^*, x^*(t))$ a new observation with unknown class membership y^*, and let $d(\cdot, \cdot)$ denote a semi-metric. Then the observations are ordered such that

$$d\big(x^*(t), x_{(1)}(t)\big) \leq \dots d\big(x^*(t), x_{(k)}(t)\big) \leq \dots d\big(x^*(t), x_{(N)}(t)\big), \tag{3.1}$$

with the $x_{(1)}(t), \dots x_{(N)}(t)$ being observations from the learning sample. Using (3.1), we define the neighborhood $\mathcal{N}\left(x^*(t)\right)$ of the k nearest neighbors of $x^*(t)$,

$$\mathcal{N}\left(x^*(t)\right) = \{x_j(t) : d(x^*(t), x_j(t)) \leq d(x^*(t), x_{(k)}(t))\}. \tag{3.2}$$

With $I(\cdot)$ denoting the indicator function, the estimated probability $\hat{\pi}_g$ that covariate $x^*(t)$ belongs to class g is given by

$$\hat{\pi}_g = \frac{1}{k} \sum_{x_j(t) \in \mathcal{N}(x^*(t))} I(y_j = g).$$

Similar to the k-nearest-neighbor classifier for multivariate data described in the introduction, the unknown y^* is assigned to the class that is most frequent within the neighborhood $\mathcal{N}\left(x^*(t)\right)$, i.e., to the class of highest probability, $y^* = \underset{g}{\operatorname{argmax}}\left(\hat{\pi}_g\right)$.

This simple functional k-nearest-neighbor approach can be extended to a functional ensemble which, in its structure, is similar to the model presented in Gertheiss and Tutz (2009). If we use several semi-metrics $d_l(\cdot, \cdot)$, $l = 1, \dots, p$, instead of one specific semi-metric only, the order (3.1) of the observations relative to the new observation $x^*(t)$ depends on the distance measure $d_l(\cdot, \cdot)$. For distance $d_l(\cdot, \cdot)$, we define the neighborhood $\mathcal{N}_l\left(x^*(t)\right)$ of the k nearest neighbors of $x^*(t)$ analogously to neighborhood (3.2). The corresponding posterior probability estimates are denoted by $\hat{\pi}_{gl}$ and given by

$$\hat{\pi}_{gl} = \frac{1}{k} \sum_{x_j(t) \in \mathcal{N}_l(x^*(t))} I(y_j = g).$$

The overall posterior probability estimate $\hat{\pi}_g$ that function $x^*(t)$ is from class g is set up as an ensemble

$$\hat{\pi}_g = \sum_{l=1}^{p} c_l \hat{\pi}_{gl} \tag{3.3}$$

$$\text{with } c_l \geq 0 \ \forall l, \ \sum_{l=1}^{p} c_l = 1, \tag{3.4}$$

where the coefficients c_l are unknown and have to be estimated. The constraint (3.4) not only yields identifiability of the coefficients (in a least square sense, see Soetaert et al., 2013, and the references therein) but also ensures that the probability estimates $\hat{\pi}_g$ are proper probabilities in the sense that $0 \leq \hat{\pi}_g \leq 1 \ \forall g$ and for all potential, or future, $x^*(t)$; see Proposition (1) in Gertheiss and Tutz (2009). The coefficients give a weight to each estimate $\hat{\pi}_{gl}$, and with that to every semi-metric $d_l(\cdot, \cdot)$. This enables the ensemble to determine which semi-metric $d_l(\cdot, \cdot)$, i.e., which curve characteristic, yields the highest contribution to $\hat{\pi}_g$ and is the most informative concerning the discrimination of the classes. The strength of the method is that feature selection is a built-in feature of the method; in contrast to, for example, functional principal component analysis (FPCA) (Di et al., 2009; Goldsmith et al., 2013). As in the multivariate case, FPCA is a method to project the feature space on the eigenfunction space of the covariates' covariance matrix. The covariance matrices $\mathbf{\Sigma}_i(t; t_0) = \text{cov}(x_i(t); x_i(t_0))$ per curve $x_i(t)$ are estimated in two steps (cf. Appendix B.1). The eigenfunctions $\Phi_e(t)$ and eigenvalues λ_e, $e = 1, \ldots E$, of the corresponding smoothed covariance matrices constitute the functional principal component basis functions and score variances. Here, the final number of scores E has to be chosen. Another popular approach, without automatic feature selection, however, is the method by Ferraty and Vieu (2006), where a fixed kernel function and a semi-metric have to be chosen by the user (see also Section 3.3.1).

The ensemble (3.3) can be extended to include further parameters. For instance, the order of derivation a and the number of nearest neighbors k have to be chosen to calculate the semi-metrics and with that the posterior probability estimates $\hat{\pi}_{gl}$. We include the order a of the derivative of the covariates in the ensemble (3.3) by using it inherently in the semi-metrics, such that $d(\cdot, \cdot) = d\left(x^{*(a)}(t), x_j^{(a)}(t)\right)$. Moreover, the number of nearest neighbors k is no longer assumed to be fixed, but to be from a given set of M numbers of nearest neighbors, $k \in \mathcal{K} = \{k_1, \ldots k_M\}$. This means that the index l now represents a tuple $\{d(\cdot, \cdot), a, k\}$, with $d(\cdot, \cdot)$ denoting the distance measure, a denoting the order of the derivative and k the number of nearest neighbors used. The corresponding neighborhood is denoted by $\mathcal{N}_{(l)}\left(x^{*(a)}(t)\right)$, which is used to calculate the single posterior probability estimate $\hat{\pi}_{g(l)}$. Ensemble (3.3) thus extends to an ensemble including $l = 1, \ldots, p$ ensemble

members, each one characterized by an unique tuple $\{d(\cdot,\cdot), a, k\}$. By assigning weights c_l to every ensemble member, the relevance of a combination of $d(\cdot,\cdot)$, a and k is automatically determined. The weighting of the single k's from the set $\mathcal{K}$ is another important advantage of our ensemble, since only few techniques exist for determining an optimal choice of k, see for example Hall et al. (2008) for the case of multivariate data.
Since one can choose the semi-metrics that are used in the k-nearest-neighbor ensemble, the ensemble can also be applied to functional covariates that are not square integrable simply by adapting the semi-metrics. Also, the approach is quite robust against single outliers because it focuses on various curve characteristics.

In the simulation studies and application sections below, semi-metrics as well as corresponding parameters are chosen with respect to the data at hand. Each of the resulting tuples will be coded by a number, called "coefficient number". Where reasonable, this coefficient number is again decoded and the tuple details are given.

3.2.3 Estimation of Weights

The weights c_l can be estimated from the learning sample by minimizing the global Brier score (Brier, 1950)

$$Q \;=\; \sum_{i=1}^{N}\sum_{g=1}^{G}\left(z_{ig}-\hat{\pi}_{ig}\right)^2, \tag{3.5}$$

where $z_{ig}=1$ if $y_i=g$ and $z_{ig}=0$ otherwise codes the response. The Brier score is a strictly proper scoring rule (Gneiting and Raftery, 2007), and the only one that (up to a positive linear transformation) fulfills the properties Selten (1998) demands of scoring rules. Among others, one advantage of the Brier score over other measures as, for example, the logarithmic score, is that it is neither hypersensitive nor insensitive. Not being hypersensitive means that the score does not react strongly on small differences between small probabilities, especially probabilities of value (around) zero. Not being insensitive means that the expected score loss $\sum_{g=1}^{G}\left(\pi_{ig}-\hat{\pi}_{ig}\right)^2$ corresponding to the Brier score adequately reflects the difference between the underlying true and the predicted distribution of the probabilities (Selten, 1998).

Estimation in Practice

The global Brier score (3.5) is interpreted as a function of the coefficient vector $\boldsymbol{c} = (c_1,\dots c_p)^T$ of the coefficients c_l,

$$Q(\boldsymbol{c}) = \underset{NG\times 1}{\left(\boldsymbol{z}\right)} - \underset{NG\times pp\times 1}{\left(\boldsymbol{P}\,\boldsymbol{c}\right)}^T \left(\underset{NG\times 1}{\boldsymbol{z}} - \underset{NG\times pp\times 1}{\boldsymbol{P}\,\boldsymbol{c}}\right), \tag{3.6}$$

with vector $\boldsymbol{z} = (\boldsymbol{z}_1|\dots|\boldsymbol{z}_N)^T$, $\boldsymbol{z}_i = (z_{i1},\dots z_{iG})^T$, $i = 1,\dots N$, $g = 1,\dots G$, and matrix $\boldsymbol{P} = \left(\boldsymbol{P}_1^T|\dots|\boldsymbol{P}_N^T\right)^T$, where

$$\boldsymbol{P}_i = \begin{pmatrix} \hat{\pi}_{i1(1)} & \dots & \hat{\pi}_{i1(p)} \\ \vdots & \dots & \vdots \\ \hat{\pi}_{iG(1)} & \dots & \hat{\pi}_{iG(p)} \end{pmatrix}$$

merges the estimates of the single posterior probabilities $\hat{\pi}_{ig(l)}$, with classes g per row, and all combinations of semi-metrics, orders of derivation of the covariates, and numbers of nearest neighbors per column. Here, the single posterior probabilities $\hat{\pi}_{ig(l)}$ are estimated via leave-one-out cross-validation for each $x_i(t)$ from the learning sample (as otherwise the nearest neighbor of observation i would always be observation i itself). Alternatively, other procedures such as K-fold cross-validation could be used.
Minimizing Equation (3.6) with respect to the coefficients c_l by $\min_{\boldsymbol{c}}(Q(\boldsymbol{c}))$ yields a way of estimating the coefficients in terms of a quadratic programming problem. By employing the constraints (3.4) on the coefficients, the estimation procedure implicitly uses a (positive) Lasso-type penalty (see, e.g., Tibshirani, 1996), which typically sets some coefficients c_l to be exactly zero and thus enables feature selection. For solving the quadratic programming problem, we use the `lsei`-function of the `R` package `limSolve` (Soetaert et al., 2013; R Core Team, 2017).

3.2.4 The Functional Nearest Neighbor Ensemble Including Multiple Covariates

If V functional covariates $x_v(t)$, $v = 1,\dots V$, instead of a single functional covariate $x(t)$ are available, two problems have to be adressed. First, the covariates might be defined on different scales, or represent totally different situations, as for example measurements on a time and a spatial scale. Concerning the k-nearest-neighbor ensemble, this means that the semi-metrics possibly differ in their adequacy concerning different covariates. Second, the content of information of the covariates might differ, and with that their individual importance for the discrimination task. All this, however, is easily accounted for when using our nearest neighbor ensemble.
Let $x_{iv}(t)$ denote the ith observation of covariate $x_v(t)$, y_{vj} denotes the corresponding class membership, and $x_v^*(t)$ a new observation of that covariate. Further, let $d_v(\cdot,\cdot)$ denote semi-metrics that are used on covariate $x_v(t)$. Again, each tuple $\{d_v(\cdot,\cdot), a, k\}$ is represented by the index l. There are now V neighborhoods denoted by $\mathcal{N}_{v(l)}\left(x_v^{*(a)}(t)\right)$, and defined in the same way as before. The posterior probability estimate $\hat{\pi}_{gv(l)}$ that covariate $x_v^*(t)$ belongs to class g when considering $k \in \mathcal{K}$ nearest neighbors, and semi-metric $d_v(\cdot,\cdot)$ defined on the derivative of order a of the covariates, is given by

$$\hat{\pi}_{gv(l)} = \frac{1}{k} \sum_{x_{jv}^{(a)}(t) \in \mathcal{N}_{v(l)}\left(x_v^{*(a)}(t)\right)} I(y_{vj} = g).$$

Analogously to the univariate ensemble, the overall posterior probability estimate $\hat{\pi}_g$ that $y^* = g$ is set up as an ensemble

$$\hat{\pi}_g = \sum_{v=1}^{V} \sum_{l=1}^{p} c_{vl} \hat{\pi}_{gv(l)}$$
$$\text{with } c_{vl} \geq 0 \ \forall v, l, \sum_{v=1}^{V} \sum_{l=1}^{p} c_{vl} = 1. \tag{3.7}$$

With the assignment of a coefficient c_{vl} per covariate type $x_v(t)$, our functional k-nearest-neighbor ensemble permits not only for feature selection, but additionally allows for variable selection from the V covariates. Moreover, we may in- and exclude additional non-functional covariates: simply by defining appropriate distance measures on the corresponding predictors' space and including the resulting posterior probability estimates in the ensemble. This flexibility and general applicability is a huge advantage of our approach over existing methods for nonparametric functional discrimination. Sometimes, and depending on the data, however, the general model can be simplified, for example, by using the same set of semi-metrics for all V covariates.
The weight estimation can be performed analogously to the univariate case described in Section 3.2.3. For each covariate type $x_v(t)$, a matrix $\boldsymbol{P}_v = \left(\boldsymbol{P}_{1v}^T | \ldots | \boldsymbol{P}_{Nv}^T\right)^T$ is calculated, and these single matrices are merged to a final matrix $\boldsymbol{P} = (\boldsymbol{P}_1 | \ldots | \boldsymbol{P}_v)$, which is used for the coefficient estimation.

3.3 Simulation Studies

The performance of ensemble (3.3) and its value concerning the interpretability of the estimated coefficients is investigated in simulation studies. All results will be compared to alternative classification methods. Existing methods take either the whole curve or few of its characteristics into account (as in James, 2001; Rossi and Villa, 2006; Epifanio, 2008). Some are interpretable in terms of a common (functional or non-functional) statistical model, e.g., a functional logistic model, some rather act like "black boxes". The main advantage of our approach is that its interpretability is based on a wide range of, potentially very different, curve characteristics through the ensemble of semi-metrics. All classification methods used are listed in Table 3.2.
Since only a limited number of classification methods for functional data has been developed, and only a few come with an implementation, we also include multivariate models. For the multivariate models, the functional principal component (FPC) scores instead of the functional covariates will be used (as has also been done, for example, by Ramsay and Silverman, 2002). Those scores have been computed with the `fpca.sc`-function of the `R`-package `refund` (Di et al., 2009; Crainiceanu et al., 2013; Goldsmith et al., 2013; R Core Team, 2017). For more details on FPC scores, see Appendix B.1. The number of scores is

Method	Abbreviation	R function used (package name)
Functional k-nearest-neighbor ensemble	kNN Ensemble	see online supplement in Fuchs et al. (2015a)
Nonparametric functional classification (NPFC)	NPFC-deriv	funopadi.knn.lcv (http://www.math.univ-toulouse.fr/staph/npfda/)
NPFC	NPFC-Fourier	see above
NPFC	NPFC-mplsr	see above
NPFC	NPFC-pca	see above
Functional linear model	FLM-log	gam (mgcv)
Support vector classifiers	SVM-cov.	svm (e1071)
Support vector classifiers	SVM-FPCs	see above
Random forests	RF-cov.	randomForest (randomForest)
Random forests	RF-FPCs	see above
Linear discriminant analysis	LDA	lda (MASS)
Penalized discriminant analysis	PDA-cov.	fda (mda)
Multinomial model	mM	maxent (maxent)

Table 3.2: The classification methods used for comparison. The second column gives the abbreviations that are used when presenting the results, the third column gives details concerning the implementations.

chosen such that at least 95% of the learning samples' variability can be explained. The considerably large proportion of 95% ensures that all substantial features of the functions are covered, as the scores with largest variance (corresponding to the first one or two principal components only) are not necessarily those with largest discriminative power. For further notes on the relationship between the choice of the number of principal components and the prediction performance, see Appendix B.2.

3.3.1 Competing Methods

Nonparametric functional classification
A nonparametric functional classification (NPFC) approach was introduced in Ferraty and Vieu (2003). Analogously to our ensemble (3.3), posterior probabilities $\hat{\pi}_{g,h}(x^*(t))$ of the probability that a functional random covariate $x^*(t)$ is of class g are estimated. Estimation is based on one (pre-) chosen semi-metric, and done via a consistent kernel estimator

$$\hat{\pi}_{g,h}(x^*(t)) = \frac{\sum_{j=1}^{N} I(y_j=g) K\big(h^{-1} d(x^*(t), x_j(t))\big)}{\sum_{j=1}^{N} K(h^{-1} d(x^*(t), x_j(t)))},$$

with bandwidth h and $K(\cdot)$ being a fixed positive kernel function. The bandwidth h is determined by minimizing the criterion $M_{(x_j(t),y_j)}(h) = 1 - \sum_{j=1}^{N} I(\hat{y}_{j,h} = y_j)/N$. $x^*(t)$ is assigned to the class with the highest estimated probability. There are, in contrast to our approach, two weak points in this estimation method. First, it uses a single, unweighted semi-metric, in contrast to our approach, which uses a variety of semi-metrics and estimates their weights with respect to their discriminative power. The user has to choose both, the kernel function $K(\cdot)$ and the semi-metric $d(\cdot,\cdot)$. The second drawback is that the NPFC approach allows only for a single covariate. The extension to multiple covariates is not straightforward, in contrast to the simple extension of our ensemble.
For our comparison, we use four semi-metrics implemented for this method. The first semi-metric will be called *NPFC-deriv*. After approximating covariates $x(t)$ by a B-spline basis of $\mathcal{B}$ B-spline functions $B(t)$ and coefficients α such that

$$x(t) \approx \tilde{x}(t) = \sum_{b=1}^{\mathcal{B}} \alpha_b B_b(t),$$

the semi-metric is defined on the approximated covariates $\tilde{x}(t)$ similar to our semi-metric $d_a^{Eucl}(\cdot,\cdot)$ by

$$d_a\left(\tilde{x}^*(t), \tilde{x}_j(t)\right) = \sqrt{\int \left(\tilde{x}^{*(a)}(t) - \tilde{x}_j^{(a)}(t)\right)^2 dt}.$$

Parameters that have to be chosen by the user are the order of derivation a and the number of the interior knots of the B-spline basis. The second semi-metric, called *NPFC-Fourier*, builds the same semi-metric, but uses covariates approximated by a Fourier expansion. Parameter choices are the order of derivation a and the number of basis functions. The third semi-metric is denoted by *NPFC-mplsr*. It uses the decomposition of the covariates and response via multivariate partial least squares regression (MPLSR). Let $\boldsymbol{\nu}_D$ denote a vector calculated by MPLSR when D factors are retained, and let ω_q denote quadrature weights from the integral approximation $\int (x^*(t) - x_j(t))\, dt \approx \sum_{q=1}^{Q} \omega_q \left(x^*(t_q) - x_j(t_q)\right)$. The semi-metric *NPFC-mplsr* is defined as

$$d_D\left(\boldsymbol{x^*}, \boldsymbol{x}_j\right) = \sqrt{\left[\sum_{q=2}^{Q} \omega_q \left(x^*(t_q) - x_j(t_q)\right) \boldsymbol{\nu}_{|q}^D\right]^2},$$

with $\boldsymbol{x}_\bullet = (x_\bullet(t_1), \ldots x_\bullet(t_Q))^T$ denoting a vector of covariate values at the observation points $t_q \in \mathbb{D}$, $q = 1, \ldots Q$. The user has to choose the number of retained factors D. The last semi-metric is called *NPFC-pca* and is based on a FPCA decomposition , with $\boldsymbol{\nu}_d$ denoting the dth eigenvector and weights ω_q as above. The respective semi-metric is

$$d_D\left(\boldsymbol{x^*}, \boldsymbol{x}_j\right) = \sqrt{\sum_{d=1}^{D} \left[\sum_{q=1}^{Q} \omega_q \left(x^*(t_q) - x_j(t_q)\right) \boldsymbol{\nu}_{d_{|q}}\right]^2},$$

and the user again has to choose the number of retained factors D.
For more details on the semi-metrics and the NPFC method, see Ferraty and Vieu (2006).

For all four semi-metrics, the parameters that have to be specified are chosen via K-fold cross-validation (CV), minimizing the mean prediction error.

Functional linear model
In the case of a two-class problem, we use a parametric functional model. This means that the functional covariates $x_{iv}(t)$ are directly used as functional predictors. With a Bernoulli distributed response y_i based on the linear predictor η_i, with intercept β_0 and V smooth terms, the model takes the form

$$y_i \sim B\left(1, \frac{\exp(\eta_i)}{1+\exp(\eta_i)}\right) \text{ with } \eta_i = \beta_0 + \sum_{v=1}^{V} \int_{\mathbb{D}} x_{iv}(t)\xi_v(t)dt.$$

This model was examined for instance in Reiss and Ogden (2009), Wood (2011), and Gertheiss et al. (2013), and is implemented in the `gam`-function of the `R` package `mgcv` (Wood, 2014). It will be referred to by the abbreviation *FLM-log.*

Support vector machines
Support vector machines (SVM) try to find a not necessarily linear decision boundary by transforming the given feature space in such a way that a linear boundary between classes exists. They can deal with problems where the feature spaces of the single classes overlap (Hastie et al., 2011). In the case of $G > 2$ classes, the SVM are trained as binary classifiers following the 'one-against-one' approach. The posterior probabilities are then obtained via quadratic optimization after fitting a logistic distribution using maximum likelihood to the decision values of all binary classifiers (Meyer et al., 2014). We use the implementation of the `R` package `e1071` (Meyer et al., 2014), by using the function `svm`, with `probability=TRUE` and default settings else. The SVM is applied to both, the discretized data $x_{iv}(t_q)$ (called *SVM-cov.*) and the FPC scores (called *SVM-FPCs*).

Random forests
This technique builds an ensemble of (classification) trees by growing a predefined (large) number of trees, with each tree being trained on a bootstrapped sample from the learning data. Class membership is then determined by majority vote of the ensemble. Let p denote the number of unknown coefficients that have to be estimated. The terminal nodes of each tree are split only on a randomly drawn part $r < p$ of the variables. The draw is repeated until a certain minimum node size is reached. The method is implemented in the `R` package `randomForest` (Breiman et al., 2012). We used the `randomForest`-function with 500 trees to grow and $r = \lfloor\!\lfloor \sqrt{p} \rfloor\!\rfloor$ variables randomly sampled as candidates. Random forests (RF) are applied to both the discretized data $x_{iv}(t_q)$ (called *RF-cov.*) and the FPC scores (called *RF-FPCs*).

Linear discriminant analysis
We apply linear discriminant analysis (LDA) to the FPC scores, as done by Ramsay and

Silverman (2002). LDA assumes multivariate Gaussian class-conditional densities with common covariance matrix, and builds a linear discriminant function of the density parameters. If the densities are unknown, the class-specific relative frequencies and means as well as the covariance matrix are estimated from the data via maximizing the log-likelihood (Tutz, 2012). LDA is implemented in the `R` package `MASS` (Ripley et al., 2014), function `lda`. Results from the LDA are referred to by the abbreviation *LDA*.

Penalized discriminant analysis
Penalized discriminant analysis (PDA) was developed from LDA. PDA was especially designed for high-dimensional and highly correlated covariates (Hastie et al., 1995), such that it can be applied to the discretized data. The approach is implemented in the `R` package `mda` (Hastie et al., 2015b), function `fda`. Results from the PDA are referred to by the abbreviation *PDA-cov.*

Multinomial model
A multinomial logistic regression model is used on the FPC scores. This method is implemented in the `maxent`-function of the `R` package `maxent` (Jurka and Tsuruoka, 2013). The abbreviation used in the results is *mM*.

3.3.2 Simulation Study A

It is important to note that the single semi-metric parameter choices are based on background information concerning the gas sensor data examined in Section 3.5. The first generating process (GP) used to build functional covariates is based on the gas measurements' mean, such that the parameter choices reflect certain curve characteristics. In contrast, the second generating process builds curves of a very different shape, such that the parameters are essentially arbitrary. Nonetheless, our k-nearest-neighbor ensemble will be shown to yield a sensible feature selection and a good classification performance.

Setup

We will evaluate our ensemble (3.3) for two generating processes simulating two data situations, one two-class and one multi-class problem. For both, let $U(\tau_1, \tau_2)$ denote an uniform distribution with limits $[\tau_1, \tau_2]$, $N(\mu, \sigma^2)$ a normal distribution with mean μ and variance σ^2, and $f(t; \mu, \sigma^2)$ a normal density function with mean μ and variance σ^2.
In the first generating process, we will take the gas sensor data into account by using its overall mean

$$x_{g\bar{a}s}(t) = \frac{1}{N} \sum_{n=1}^{N} x_n(t),\ n = 1, \dots N,$$

across all N measurements as well as gas species, as a "starting point" for the covariate generation. The ith covariate,

$$x_i(t) = x_{g\bar{a}s}(t) + \alpha_i \max\left(x_{g\bar{a}s}(t)\right) \sum_{s=1}^{L_i} \sin(\gamma_s t),$$

is the sum of the gas measurements' mean and a sum of varying sine functions. For every covariate $x_i(t)$, the parameters are $\alpha_i \sim U(-1,1)$, $L_i = \llbracket \beta_i \rrbracket$, with $\beta_i \sim U(1,7)$, and $\gamma_s \sim U(-2.5, 2.5)$. The parameters α_i, β_i and γ_s are drawn from uniform distributions on the respective intervals. The corresponding classes are defined with regard to the curves' means,

$$y_i = \begin{cases} 1 & <=> \quad \int_{\mathbb{D}} x_i(t)dt < \int_{\mathbb{D}} x_{g\bar{a}s}(t)dt \\ 2 & <=> \quad \int_{\mathbb{D}} x_i(t)dt \geq \int_{\mathbb{D}} x_{g\bar{a}s}(t)dt. \end{cases}$$

The second generating process builds functional covariates

$$x_i(t) = \sum_{s=1}^{L_i} f_s(t; \mu_s, \sigma_s^2)$$

as a sum of L_i normal densities $f_s(t; \mu_s, \sigma_s^2)$, with means $\mu_s \sim U(-1,3)$, variances $\sigma_s^2 = |\nu_s|$, $\nu_s \sim N(0,1)$, and L_i being chosen at random from $\{1, \dots 11\}$. Since computation is only possible for a discretized covariate, let $\boldsymbol{x}_i = (x_i(t_1), \dots x_i(t_Q))$ denote the discretization of $x_i(t)$ at observation points $t_q \in \mathbb{D}$, $q = 1, \dots Q$. The classes y_i are defined with respect to the position of the maximum of the curves. To this end, we divide the domain of definition in five equal sized parts and assign class $y_i = g$ if the maximum of curve $x_i(t)$, $\max(x_i(t)) = x_i(t)_{|t=t_{\max(x_i(t))}}$, lies in the gth part of the domain, with $g \in \{1,2,3,4,5\}$, namely

$$y_i = g \qquad \text{if} \qquad \left(t_{(gQ-Q)/5} < t_{\max(x_i(t))} <= t_{gQ/5}\right).$$

An example of covariates generated by these processes can be found in Figure 3.2. For both generating processes, the number of observation points for the discretized covariates $\boldsymbol{x}_i$, $i = 1, \dots N$, is $Q = 100$, with $t_q \in \mathbb{D} = [0.1, 1]$, $q = 1, \dots Q$ equidistant points. To be able to use the semi-metrics of Table 3.1 on the discretized curves, the integrals are approximated by quadrature sums (analogously to, for example, Wood, 2011). The number of observations N is one out of the set $\{100, 300, 1000\}$.

The data generation, and with that the estimation of the coefficients c_l of Model (3.3), is repeated $W = 100$ times to draw conclusions concerning the stability of estimation. As numbers of nearest neighbors k, the set $k \in \mathcal{K} = \{1, 5, 11, 21\}$ is used. As orders of derivative a of the covariates, the set $a \in \{0, 1, 2\}$ is used. For semi-metric $d_{a,\mathbb{D}_{small}}^{shortEucl}(\cdot,\cdot)$, one of the intervals $[t_1, t_{17}]$, $[t_{18}, t_{36}]$, $[t_{37}, t_{56}]$, $[t_{57}, t_{76}]$, $[t_{77}, t_{100}]$ or $[t_{30}, t_{65}]$ is used for $\mathbb{D}_{small}$. For semi-metric $d_a^{relAreas}(\cdot,\cdot)$, $\mathbb{D}_1$ is one of the intervals $[t_1, t_{17}]$, $[t_{57}, t_{76}]$ or $[t_{30}, t_{65}]$ and $\mathbb{D}_2 = [t_{37}, t_{56}]$. For semi-metric $d_{no}^{Jump}(\cdot,\cdot)$, one of the sets $\{t_{15}, t_{19}\}$, $\{t_{34}, t_{40}\}$, $\{t_{54}, t_{58}\}$ or $\{t_{74}, t_{78}\}$ is used for $\{t_n, t_o\}$. For semi-metric $d_a^{Points}(\cdot,\cdot)$, an equidistant grid $t_q \in \{t_{mQ/10}\}$, $m = 1, \dots 10$, is used. For $d_{a\tau}^{Scan}(\cdot,\cdot)$, function $\phi_\tau(t) = \left(\frac{\max\left(\mathbb{X}^{(a)}(t)\right)}{\max(\phi_{1,\tau}(t))}\right) \phi_{1,\tau}(t)$ with $\max\left(\mathbb{X}^{(a)}(t)\right) := \max\left(\left\{\max\left(x_i^{(a)}(t)\right), i = 1, \dots N\right\}\right)$ and $\phi_{1,\tau}(t) = \frac{1}{\sqrt{2\pi}\sigma} \exp^{-\frac{1}{2}\left(\frac{t-\tau}{\sigma}\right)^2}$ is

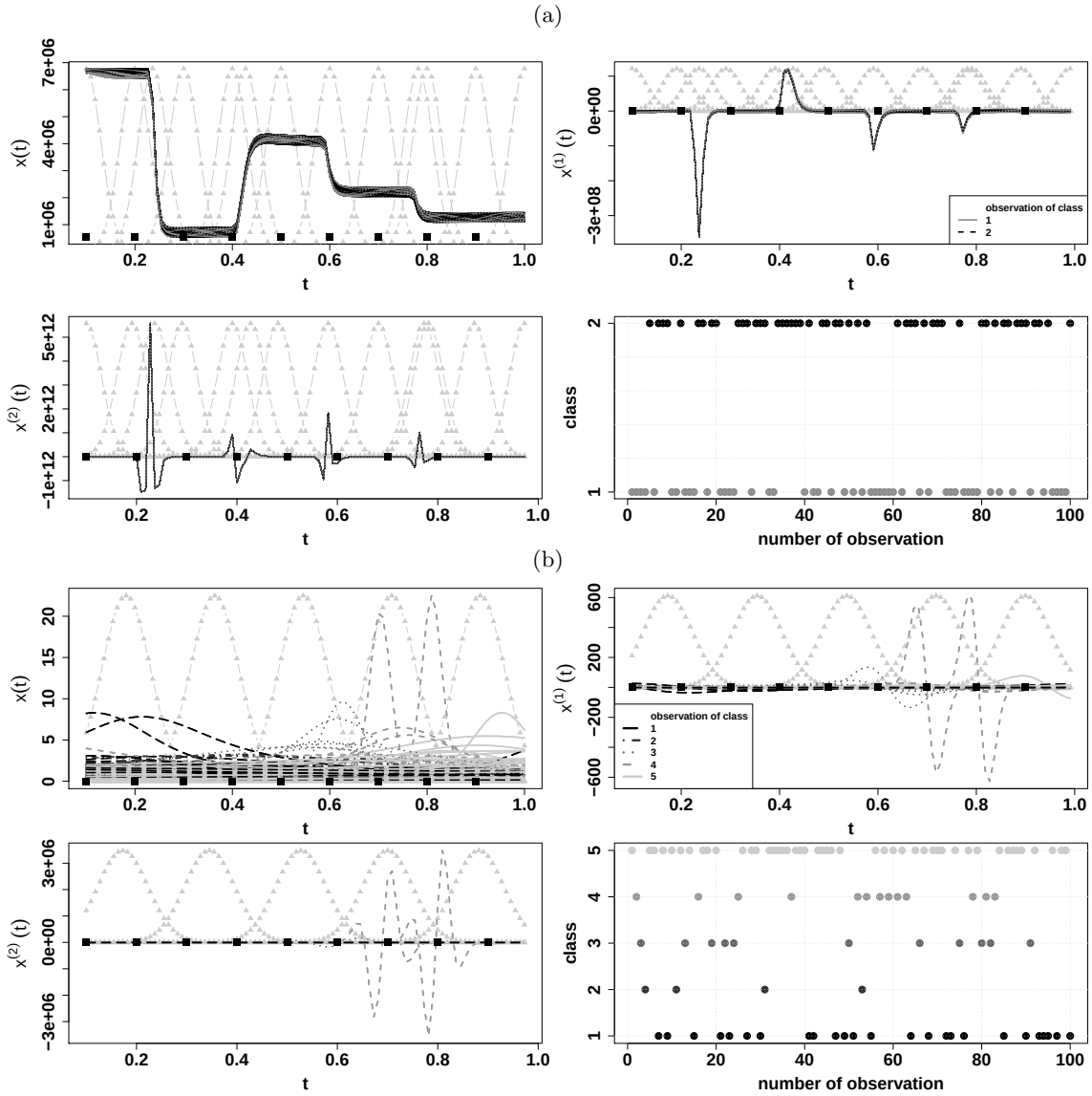


Figure 3.2: (a) Exemplary realizations of the first generating process concerning the two-class discrimination problem. The upper left panel shows $N = 100$ realizations of the covariates. The upper right panel shows their first, the lower left panel their second derivatives. Curve color and line type coding is with respect to the curves' class. The function $\phi_\tau(t)$ used in $d_{a\tau}^{Scan}(\cdot,\cdot)$ is depicted as light gray, dotted lines, the impact points t_q used in $d_a^{Points}(\cdot,\cdot)$ as black boxes. The class of each covariate can be found in the lower right panel. (b) The same for the multi-class generating process.

used. The parameters for the first generating process here are $\sigma = 0.03$ and $\tau \in \{0.10, 0.20, 0.24, 0.30, 0.40, 0.43, 0.50, 0.60, 0.70, 0.78, 0.80, 0.90, 1.0\}$, those for the second are $\sigma = 0.05$, $\tau \in \{0.18, 0.36, 0.55, 0.73, 0.91\}$. As mentioned before, all these choices are made having the application in Section 3.5 in mind, but are rather arbitrary with respect to the second generating process. This enables us to impartially test the performance and interpretability of the estimated coefficients. If special knowledge about the data at hand is available, one might optimize the above parameters, as has been done in the application Sections 3.4 and 3.5. All of the semi-metrics introduced in Section 3.2.1, except $d_{no}^{Jump}(\cdot,\cdot)$, are employed on the generated covariates as well as on their centered counterparts $\tilde{x}_i(t) = x_i(t) - \bar{x}_i(t)$. Thus, $p_{GP1} = 696$ coefficients c_l have to be estimated for the first generating process, and $p_{GP2} = 504$ for the second one.
The optimal parameters per semi-metric of the NPFC approach are chosen in such a way that they minimize the mean prediction error of a 10-fold CV.

Results

Figure 3.3 illustrates the selection results of the proposed ensemble method. The first two panels give the results of the first generating process, the third and fourth panels the results of the second generating process.
In the respective upper panels, the coefficients c_l that have been estimated to be of mean values above 0.001 are plotted as boxplots across $W = 100$ replications, with sample size $N = 100$. The lower panels show the respective mean (gray +) and median (black $\times$) values of these coefficients. As can be seen, only few of the $p_{GP1} = 696$ or $p_{GP2} = 504$ coefficients were selected. The estimation is similar across all three sample sizes, becoming more stable if more observations are used for estimation.
For the first generating process representing a two-class discrimination problem, one coefficient clearly dominates. This is coefficient number 8, representing a tuple of semi-metric $d^{Mean}(\cdot,\cdot)$, order of derivation $a = 0$ and number of nearest neighbors $k = 1$ (see also Table 3.3). The two coefficients with the second and third highest means are also representing tuples with $d^{Mean}(\cdot,\cdot)$. The absolute estimated value of the coefficients can not be interpreted, since the probabilities resulting from the tuple that coefficient 8 belongs to often mirrors the true response (see also Table B.1 in Appendix B.3, and Section 3.7 for a discussion of mirroring effects). Nonetheless, the choice of tuples with $d^{Mean}(\cdot,\cdot)$ is sensible, since the class assigned to a covariate depends solely on the covariates' mean. Thus, the ensemble does not only give a nicely sparse solution, but also yields sensible results in terms of interpretability.
Concerning the second generating process, recall that the classes y_i of the covariates $x_i(t)$ were assigned with respect to the position of the curves' maximum. But as seen from Table 3.3, the most important features for the discrimination of the curve classes are the curves' Euclidian distances. This can be understood from the curves' progression (see Figure 3.2). Since the $x_i(t)$ follow normal densities, they are very smooth, and show only

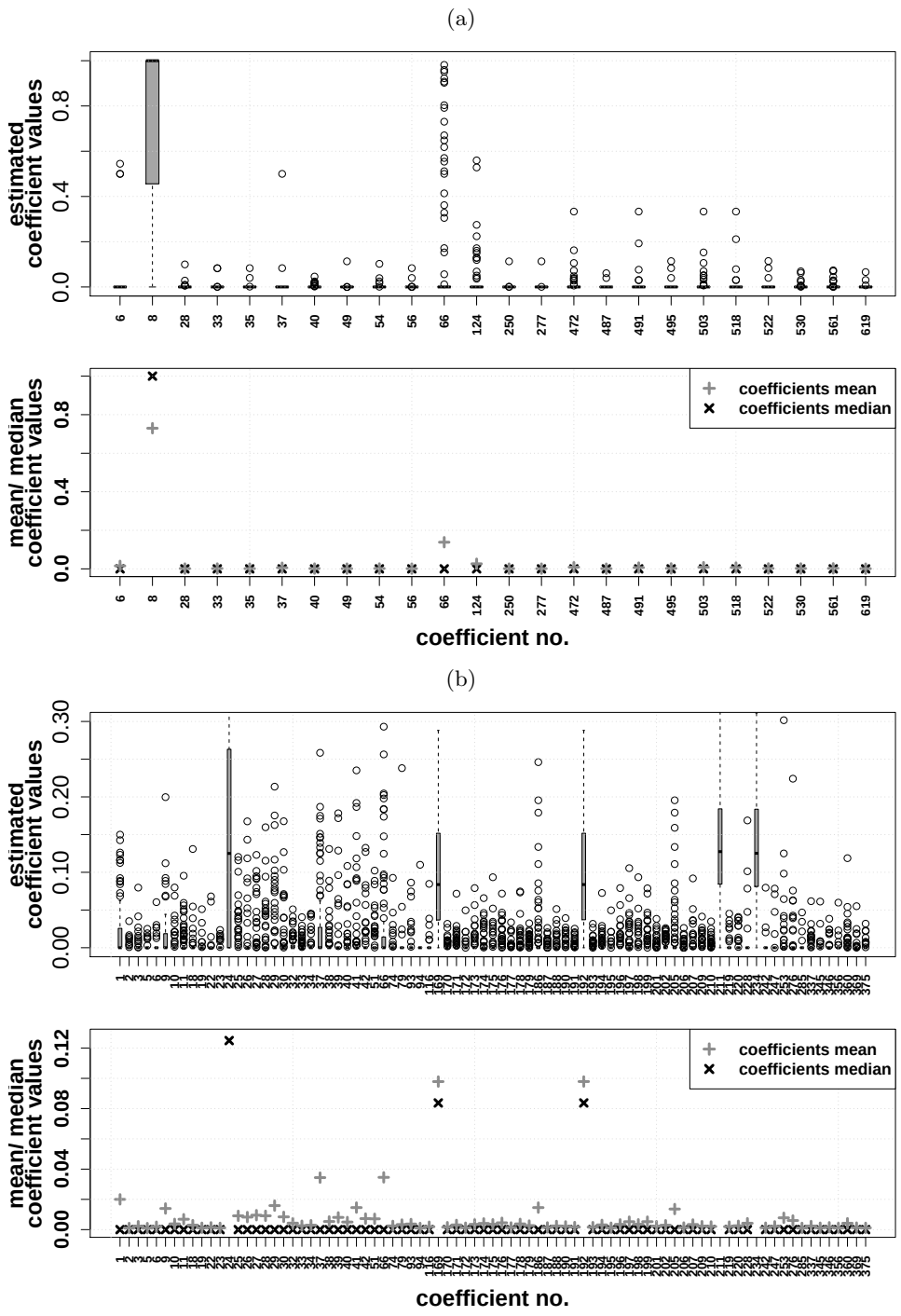


Figure 3.3: (a) Estimated coefficients for the two-class generating process yielding mean values above 0.001. The upper panel shows boxplots across 100 replications when $x_i(t) = x_{g\bar{a}s}(t) + \alpha_i \max(x_{g\bar{a}s}(t)) \sum_{s=1}^{L_i} \sin(\gamma_s t)$, $i = 1, \dots N$, with $N = 100$ observations. The second panel shows the mean and median values for these coefficients. (b) The same for the multi-class generating process.

	IDs (no.) of estimated coefficients			coefficient number	parameter tuple
GP 1	N=100	N=300	N=1000	6	$\{d^{shortEucl}$, $a = 0$, $k = 1\}$, $\mathbb{D}_{small} = [t_{77}, t_{100}]$
	8	8	8	8	$\{d^{Mean}$, $a = 0$, $k = 1\}$
	66	66	66	28	$\{d^{Scan}$, $a = 0$, $k = 1\}$, $\tau = 0.78$
	124	124	124	66	$\{d^{Mean}$, $a = 0$, $k = 5\}$
	6	6	28	124	$\{d^{Mean}$, $a = 0$, $k = 11\}$
	472	286	518	286	$\{d^{Scan}$ with $x_i(t)$ centered, $a = 1$, $k = 1\}$, $\tau = 0.7$
				472	$\{d^{Mean}$, $a = 2$, $k = 1\}$
				518	$\{d^{Scan}$ with $x_i(t)$ centered, $a = 2$, $k = 1\}$, $\tau = 0.7$
GP 2	N=100	N=300	N=1000	24	$\{d^{Eucl}$ with $x_i(t)$ centered, $a = 0$, $k = 1\}$
	24	211	234	66	$\{d^{Eucl}$ with $x_i(t)$ centered, $a = 0$, $k = 5\}$
	211	234	211	169	$\{d^{Eucl}$, $a = 1$, $k = 1\}$
	234	24	24	192	$\{d^{Eucl}$ with $x_i(t)$ centered, $a = 1$, $k = 1\}$
	192	169	66	211	$\{d^{Eucl}$, $a = 1$, $k = 5\}$
	169	192	192	234	$\{d^{Eucl}$ with $x_i(t)$ centered, $a = 1$, $k = 5\}$

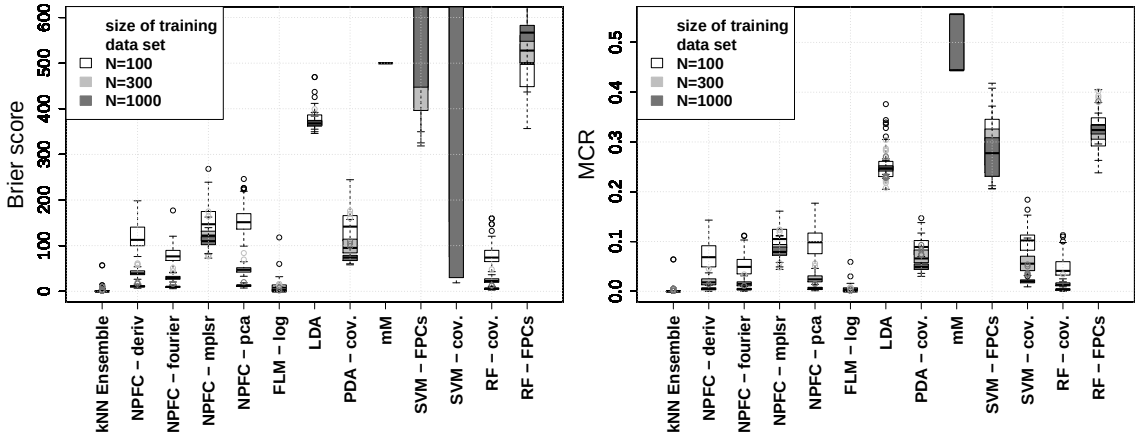
Table 3.3: Selection results for the two generating processes. Left three columns: IDs (no. according to Figure 3.3) of the five estimated coefficients that show the largest means (in decreasing order, for differing numbers of observations N). On the right, the chosen ensemble coefficients are decoded; the value of a indicates the order of derivation, k indicates the number of nearest neighbors used.

slight gradients to and from their maximum. Thus, the position of the maximum itself often does not offer more discriminative power than the whole curves' Euclidian distances. Here, the estimated weights c_l yielded additional insight in the generated data, revealing the Euclidian distance to contain most information concerning the classification task.

To validate our results, new data sets of $N_{val} = 1000$ observations are generated for each generating process, and the respective posterior probabilities $\hat{\pi}_{g(l)}$ are calculated. In addition to the Brier score, we give the misclassification rate, which is also a popular measure to judge classification performance. With y_i denoting the true class of observation $x_i(t)$ and $\hat{y}_i$ being the class assigned by the method considered, the misclassification rate is defined as $\text{MCR} = (1/N_{val}) \sum_i I(y_i = \hat{y}_i)$. However, it should be kept in mind that, in contrast to the Brier score, the misclassification rate is not a proper scoring rule concerning the estimated posterior probabilities. The global Brier scores and misclassification rates with respect to the validation data across 100 independent replicates (of training data) can be found in Figure 3.4. The first row of panels shows the results for the first generating process, the second row of panels the results for the second generating process. The results of the nearest neighbor ensemble are shown as the first boxplots. The other boxplots show the results for the competing methods from Section 3.3.1. Results for different sample sizes are plotted in different colors: (a) white boxes for $N = 100$, (b) light gray for $N = 300$, (c) dark gray for $N = 1000$. Table 3.4 shows the respective mean values of both performance measures.

As expected, the Brier score as well as the misclassification rate decrease with an increasing number of observations, except for the Brier scores of the SVM method. It seems that SVM can not adequately reflect the underlying true probability distribution. The predictive power of ensemble (3.3) for the two-class problem is very good with respect to both performance measures. Our ensemble outperforms nearly all other discrimination methods including the functional approach *NPFC* and random forests. Solely the binomial model *FLM-log* yields comparable good results when using very many observations ($N = 1000$). For the multi-class problem, our functional k-nearest-neighbor ensemble yields the lowest Brier scores and misclassification rates across all N. The most competitive method is the nonparametric functional classification approach. The choice of the semi-metric used here has a non-negligible influence on the classification performance in both data situations, with, for example, option *NPFC-deriv* using the first order of derivation $a = 1$ and 7 interior knots yielding best results in the multi-class simulation data. This is consistent with the results of our approach, since *NPFC-deriv* uses the Euclidian distance as semi-metric, and the same semi-metric is assigned with high weights by the kNN ensemble. A combination of different semi-metrics as done by our ensemble, however, is apparently the optimal choice here. In general, functional classification approaches perform better than most multivariate approaches applied to the functional principal component scores.

(a)



(b)

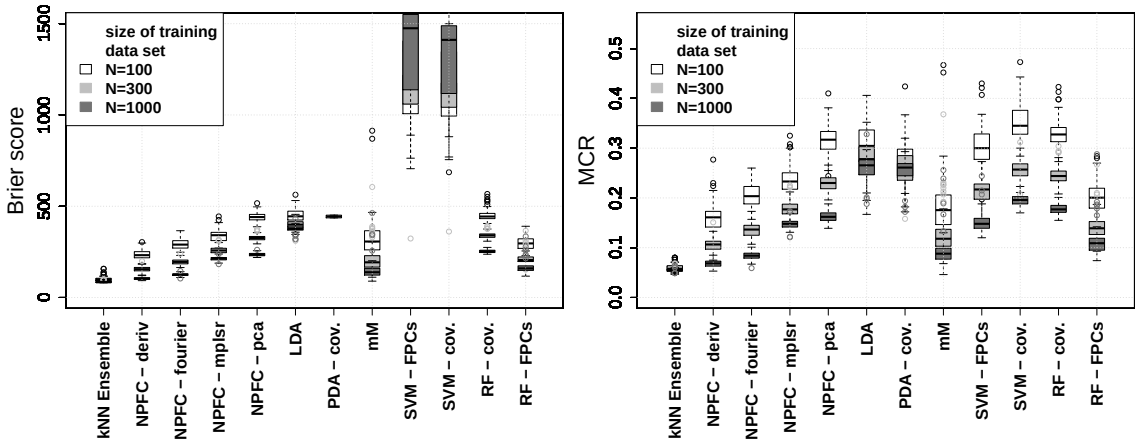


Figure 3.4: (a) Results for $N_{val} = 1000$ test observations for the two-class generating process. The models were estimated 100 times with sample sizes $N = 100$ (white boxes), $N = 300$ (light gray boxes) and $N = 1000$ (dark gray boxes). The left panel shows the Brier scores, the right panel the misclassification rates (MCR). (b) The same for the multi-class generating process.

	Method	mean Brier score			mean MCR		
		N=100	N=300	N=1000	N=100	N=300	N=1000
GP 1	kNN Ensemble	**1.61**	**0.78**	0.49	$\mathbf{9.4 \cdot 10^{-4}}$	$\mathbf{3.5 \cdot 10^{-4}}$	$1.7 \cdot 10^{-4}$
	NPFC-deriv	120.37	40.77	11.04	0.071	0.019	0.005
	NPFC-Fourier	79.16	30.01	9.85	0.052	0.015	0.005
	NPFC-mplsr	153.24	120.72	110.36	0.105	0.086	0.080
	NPFC-pca	155.66	47.87	12.63	0.099	0.025	0.006
	FLM-log.	10.2	2.2	**0.12**	0.005	0.001	$\mathbf{6.0 \cdot 10^{-5}}$
	LDA	377.66	369.14	368.5	0.252	0.247	0.248
	PDA	142.31	100.12	74.29	0.089	0.069	0.051
	mM	500	500	500	0.501	0.502	0.499
	SVM-FPCs	726.58	707.94	848.55	0.301	0.291	0.276
	SVM-cov.	903.77	911.76	1274.40	0.100	0.057	0.021
	RF-cov.	80.08	23.96	6.07	0.047	0.014	0.004
	RF-FPCs	499.93	529.27	564.31	0.319	0.324	0.324
		N=100	N=300	N=1000	N=100	N=300	N=1000
GP 2	kNN Ensemble	**100.32**	**86.68**	**83.76**	**0.059**	**0.056**	**0.054**
	NPFC-deriv	233.34	155.02	104.41	0.162	0.106	0.069
	NPFC-Fourier	292.57	195.66	125.27	0.206	0.136	0.084
	NPFC-mplsr	338.17	258.64	212.94	0.237	0.179	0.148
	NPFC-pca	441.69	326.62	235.96	0.317	0.229	0.163
	FLM-log.	-	-	-	-	-	-
	LDA	446.41	400.46	376.06	0.303	0.274	0.261
	PDA	335.23	344.33	348.63	0.272	0.259	0.256
	mM	324.8	210.162	147.74	0.182	0.127	0.092
	SVM-FPCs	1117.87	1255.98	1384.85	0.302	0.214	0.15
	SVM-cov.	1087.12	1215.11	1337.53	0.353	0.257	0.195
	RF-cov.	450.49	340.84	252.93	0.329	0.245	0.177
	RF-FPCs	296.59	210.68	166.99	0.208	0.142	0.113

Table 3.4: The mean values of both, Brier scores and misclassification rates (MCR), for differing numbers of observations N (simulation study A). The best results are highlighted by bold numbers.

3.3.3 Simulation Study B: Waveform Data

With regard to the results presented in Figure 3.4, another advantage of the k-nearest-neighbor ensemble is its robust prediction performance with regard to high in-class variability of the functional covariates, compared to the competing methods.
Functional covariates which exhibit similar characteristics in each class can be simulated by the well-studied waveform data (Ferraty and Vieu, 2003; Epifanio, 2008). Let $u_i \sim U(0,1)$ and $\varepsilon_i(t) \sim N(0,1)$ denote curve specific variables, and define three waveform functions

$$h_1(t) = \max(6 - |t - 11|, 0),$$
$$h_2(t) = h_1(t - 4), \text{ and}$$
$$h_3(t) = h_1(t + 4).$$

The functional covariates are generated by

$$y_i = 1,\ x_{1i}(t) = u_i h_1(t) + (1 - u_i)h_2(t) + \varepsilon_i(t),$$
$$y_i = 2,\ x_{2i}(t) = u_i h_1(t) + (1 - u_i)h_3(t) + \varepsilon_i(t), \text{ or}$$
$$y_i = 3,\ x_{3i}(t) = u_i h_2(t) + (1 - u_i)h_3(t) + \varepsilon_i(t).$$

The waveform functions as well as the covariates are observed on an equidistant grid of $Q = 100$ points, with $t_q \in \mathbb{D} = [1, 21]$, $q = 1, \dots Q$, see Figure 3.5 for exemplary curve realizations per class.

Analogously to previous studies (Ferraty and Vieu, 2003; Epifanio, 2008), we simulate 50 training samples containing 150 curves per class, and 50 validation samples containing 250 curves per class. All competing methods were applied to the same sample sets. The semi-metric parameters were chosen arbitrarily with respect to the functional covariates' eye-catching differences. The estimated k-nearest-neighbor ensemble coefficients yielding the highest means across the 50 estimations correspond to tuples that include the semi-metrics $d_{a=0}^{Eucl}(\cdot,\cdot)$ and $d_{a=0,\mathbb{D}_{small}}^{shortEucl}(\cdot,\cdot)$, $\mathbb{D}_{small} = [t_{30}, t_{65}] = [6.86, 13.93]$, with $k = 11$ or $k = 21$. Thus, the covariates' Euclidian distances seem to contain more discriminative power than the positions of the covariates' maxima. Figure 3.6 shows the classification results for the validation data, Table 3.5 the means of the performance measures. As can be seen, the PDA approach performs worst. The other methods perform comparable, with the NPFC and the mM approaches tending to the best results. The k-nearest-neighbor ensemble is hardly competitive in this example. Probably, different semi-metrics or parameter choices could improve the results.

3.4 Application to Real World Data – Cell Based Sensor Chips

This section deals with data of cell based silicon sensor chips. Cell based sensor technologies are promising tools concerning environmental quality monitoring (see e.g. Bohrn et al.,

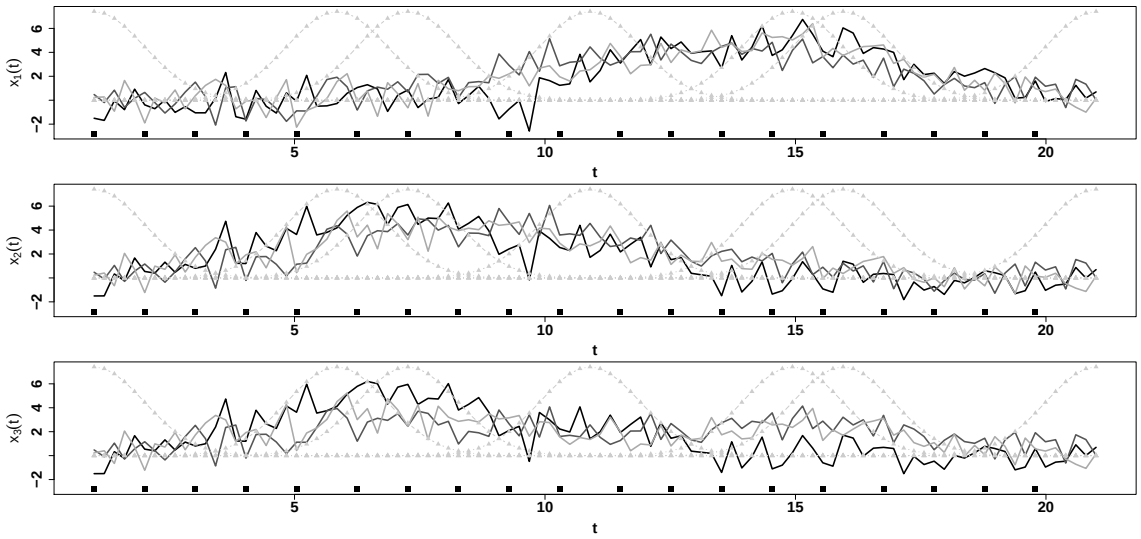


Figure 3.5: The panels show $N = 3$ realizations of waveform covariates for each class. The function $\phi_\tau(t)$ used in $d_{a\tau}^{Scan}(\cdot,\cdot)$ is depicted as light gray, dotted lines, the impact points t_q used in $d_a^{Points}(\cdot,\cdot)$ as black boxes.

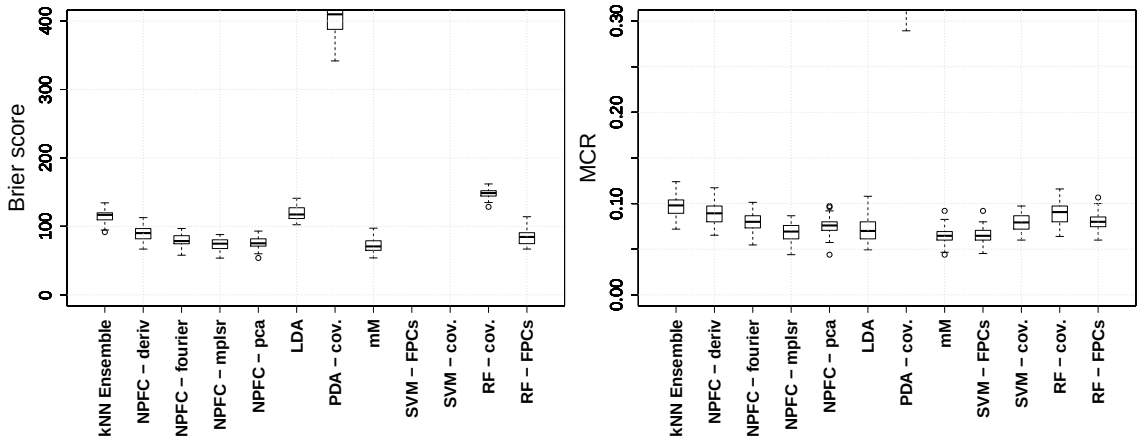


Figure 3.6: Results for $N_{val} = 250$ test observations per waveform function class. The models were estimated 50 times with sample sizes of $N = 150$ covariates per class. The left panel shows the Brier scores, where the boxes of the SVM-FPCs and the SVM-cov. methods (mean values 915.98 and 863.76) are not shown due to y-axis pruning. The right panel shows the misclassification rates (MCR).

Method	mean Brier score	mean MCR
kNN Ensemble	115.27	0.116
NPFC-deriv	89.83	0.089
NPFC-Fourier	79.74	0.079
NPFC-mplsr	73.78	0.069
NPFC-pca	76.32	0.076
LDA	119.19	0.072
PDA	418.58	0.339
mM	**71.44**	**0.065**
SVM-FPCs	915.98	**0.065**
SVM-cov.	863.76	0.079
RF-cov.	148.36	0.089
RF-FPCs	84.06	0.08

Table 3.5: The mean values of both, Brier scores and misclassification rates (MCR), for the simulated waveform data, comparing all competing classification methods. The best results are highlighted by bold numbers.

2012; Kubisch et al., 2012). The cell based chips are covered with a monolayer of a living cell population. There are three different kinds of sensors distributed across the chip surface, which record three different cell reactions. Five ion sensitive field effect transistors (ISFET), one interdigitated electrode structure (IDES) and two oxygen sensitive (CLARK) electrodes. For more details, please see Chapter 1 and Appendix D. We use the arithmetic mean of signals of the same type for our study.
Analogously to Chapter 2, we use chinese hamster lung fibroblast cells as a cell detection layer because of their stable and reliable growth (Bohrn et al., 2013). Our goal is to discriminate between measurements with nutrient medium only, and measurements where paracetamol (2.5mM) is added. Our data set includes $N = N_0 + N_1 = 120$ measurements per signal type of $Q = 89$ observation points, $N_0 = 63$ without and $N_1 = 57$ with AAP, depicted in Figure 3.7. Since, just before the test substance reaches the cells, one expects the cells to exhibit 100% viability, all signals were standardized in such a way that, at the respective data point (about 215 minutes), the signals have a value of 100.

3.4.1 Results

Since the cell chip data consists of the three very different signal types ISFET, IDES, and CLARK, the adequate approach here is to deal with them as a number of $V = 3$ covariate types. The character of the single curves, however, is similar, exhibiting all three measurement phases, the acclimatization, the testing and the devitalization phase, such that identical semi-metrics are used for each signal type. The members of ensemble (3.7) were calculated via leave-one-out. The parameters used for the semi-metrics are the numbers of nearest neighbors $k \in \mathcal{K} = \{1, 5, 11, 21\}$, and orders of derivation $a \in \{0, 1, 2\}$. The

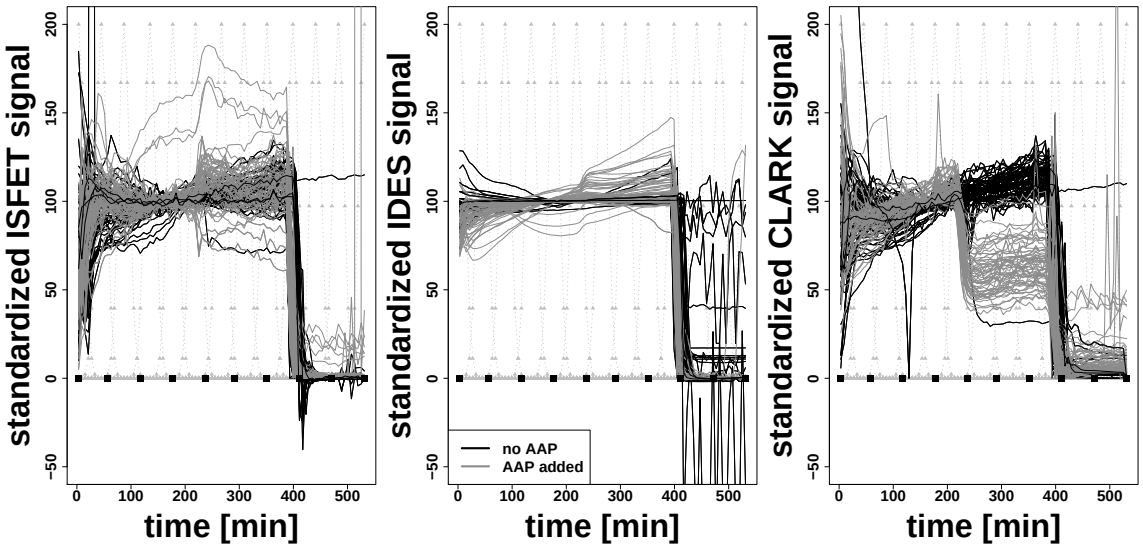


Figure 3.7: The $N = 120$ standardized signals for each of the three sensor types ISFET, IDES, and CLARK, measured at 89 time points on an equidistant grid. The gray shades represent the presence (gray lines) or absence (black lines) of AAP. The light gray, dotted lines depict the function $\phi_\tau(t)$ in $d_{a\tau}^{Scan}(\cdot,\cdot)$ used at certain observation points, the impact points t_q used in $d_a^{Points}(\cdot,\cdot)$ are depicted as black boxes.

choices of $\mathbb{D}_{small}$, $\mathbb{D}_1$, $\mathbb{D}_2$, t_q and τ reflect the signal ranges and points where the AAP reaches the cells in phase two, and the changeover of phase two and three. For semi-metric $d_{a,\mathbb{D}_{small}}^{shortEucl}(\cdot,\cdot)$, one of the intervals $[t_1, t_{35}]$, $[t_{36}, t_{40}]$, $[t_{41}, t_{64}]$, $[t_{65}, t_{69}]$ and $[t_{70}, t_{89}]$ is used for $\mathbb{D}_{small}$; for semi-metric $d_{no}^{Jump}(\cdot,\cdot)$, one of the sets $\{t_{36}, t_{39}\}$ or $\{t_{65}, t_{68}\}$ is used for $\{t_n, t_o\}$; for semi-metric $d_a^{relAreas}(\cdot,\cdot)$, $\mathbb{D}_1$ is one of the intervals $[t_1, t_{35}]$ or $[t_{41}, t_{64}]$ and $\mathbb{D}_2 = [t_{41}, t_{64}]$; for semi-metric $d_a^{Points}(\cdot,\cdot)$, an equidistant grid $t_q = t_{mQ/10}$, $m = 1, \ldots 10$, is used; and for semi-metric $d_{a\tau}^{Scan}(\cdot,\cdot)$, the function $\phi_\tau(t) = \left(\frac{\max\left(\mathbb{X}^{(a)}(t)\right)}{\max(\phi_{1,\tau}(t))}\right)\phi_{1,\tau}(t)$ with $\max\left(\mathbb{X}^{(a)}(t)\right) := \max\left(\left\{\max\left(x_i^{(a)}(t)\right), i = 1, \ldots N\right\}\right)$, $\phi_{1,\tau}(t) = \frac{1}{\sqrt{2\pi}\sigma}\exp^{-\frac{1}{2}\left(\frac{t-\tau}{\sigma}\right)^2}$, $\sigma = 10$ and $\tau \in \{3.120,\ 45.120, 87.120, 135.120,\ 177.120, 219.129, 267.129,\ 309.129, 351.129,\ 399.140,\ 441.140,\ 483.140,\ 531.140\}$ is used.

Only five of the $p = 1872$ coefficients (624 per signal type) are estimated to be of values unequal to zero. The respective coefficients are listed in Table 3.6. They correspond to the semi-metrics $d_{a\tau}^{Scan}(\cdot,\cdot)$ and $d_{a,\mathbb{D}_{small}}^{shortEucl}(\cdot,\cdot)$, which take the part of the signal where the AAP reaches the cells, around data point $t_{37} = 219.129$, into account. These results are sensible: In theory, the curve progression per signal type should be similar for two

coefficient IDs	parameter tuple	covariate	semi-metric parameters
1268	$\{d^{Scan}, a = 0, k = 1\}$	CLARK	$q = 37$, $\tau = 219.129$
1476	$\{d^{Scan}, a = 1, k = 1\}$	CLARK	$q = 37$, $\tau = 219.129$
1459	$\{d^{shortEucl}, a = 1, k = 1\}$	CLARK	$\mathbb{D}_{small} = [t_{36}, t_{40}]$
1486	$\{d^{shortEucl}$ with $x_i(t)$ centered, $a = 1, k = 1\}$	CLARK	$\mathbb{D}_{small} = [t_{36}, t_{40}]$
1501	$\{d^{Scan}$ with $x_i(t)$ centered, $a = 1, k = 1\}$	CLARK	$q = 37$, $\tau = 219.129$

Table 3.6: The five coefficients that were selected for the cell chip data. Left column: the coefficient numbers of the selected ensemble members. On the right, the selected members are decoded. The value of a indicates the order of derivation, k indicates the number of nearest neighbors used.

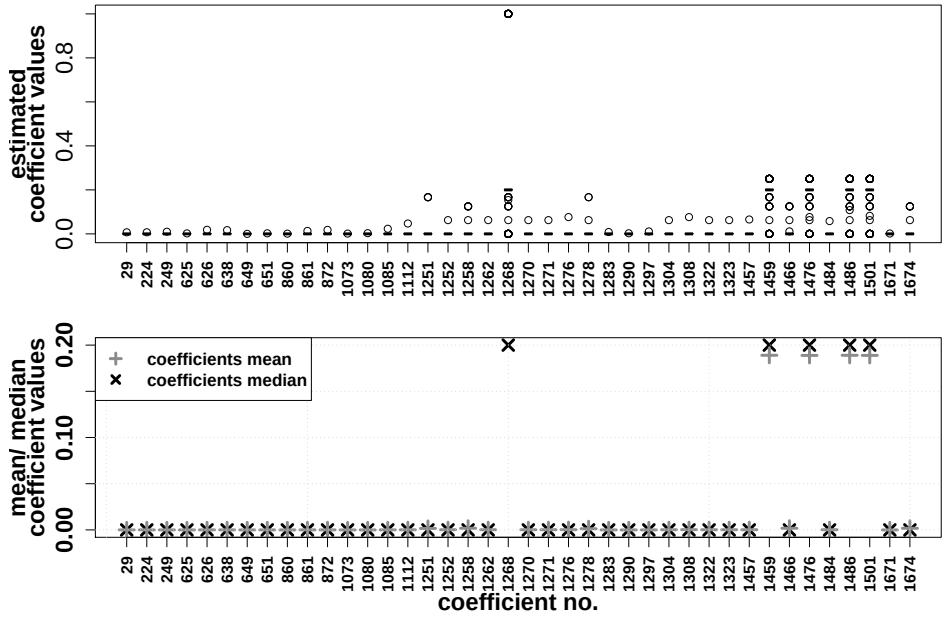


Figure 3.8: First panel: Estimated coefficients as boxplots across 25 replications of a 15-fold CV when using the cell chip data. All coefficients with a mean value unequal to zero are shown. Second panel: The mean and median values of these coefficients.

curves when the cells meet similar conditions. Furthermore, when AAP reaches the cells at about 220 minutes, this should stimulate a cell reaction, which is reflected by a jump in the curves of most measurements with AAP. In contrast, measurements without AAP should not notably alter their progression. This clustering of the curves representing non or 2.5mM AAP is especially obvious in the CLARK-signals, which are selected by the ensemble to be the most informative signal type for this classification task. Also, the ensemble gives zero weight to the first and third phase, which both exhibit the same test conditions for all classes and therefore should not yield discriminative information.

To test the performance of our model and compare its prediction accuracy to other approaches, the data was split randomly $W = 25$ times into 15 subsets of 8 observations each. With these, a 15-fold cross-validation is performed W times to estimate the coefficients of ensemble (3.7) and to validate the results.
Practically for all replications of estimation, the same five coefficients are selected that have already been selected when using the whole data set, see Figure 3.8. This shows that, for this two-class task, the ensemble coefficient estimation is stable under subsampling. Analogously to the first generating process in Simulation Study A, the probabilities of those tuples to which the coefficients in Table 3.6 belong often mirror the true response (see also Table B.2 in Appendix B.3, and Section 3.7). This leaves their contextual interpretation unaffected, but explains the nearly identical mean and median values.
Figure 3.9 summarizes the results for the validation data. The global Brier scores and misclassification rates are shown for all approaches, with the results of our functional k-nearest-neighbor ensemble being presented as the first boxes. For the NPFC approach, white boxes show results if only ISFET is used, light gray boxes if only IDES is used, and dark gray boxes if only CLARK is used. The optimal parameters per semi-metric and covariate type of the NPFC approach are chosen via minimization of the mean prediction error of a 10-fold CV. The mean Brier scores and misclassification rates can be found in Table 3.7.

For the validation data set, most multivariate approaches applied to the functional principal component scores are performing worse than the functional approaches in the Brier score. Our approach is competitive in terms of prediction performance. For the NPFC approach, the choices of the semi-metric as well as the covariate type are essential. In accordance to the results of our k-nearest-neighbor ensemble, the results of the NPFC method are best when using the CLARK-signals. When using the NPFC method, however, the input variable as well as the single semi-metric, i.e., one particular curve characteristic, have to be chosen by the user. Given those, NPFC rather acts like a "black box" but does not give interpretable results in terms of feature selection. The same is true for random forests. Thus, although its prediction performance is rather comparable than outstanding, our k-nearest-neighbor ensemble is a very attractive choice if automated, interpretable variable and feature selection is of main interest.

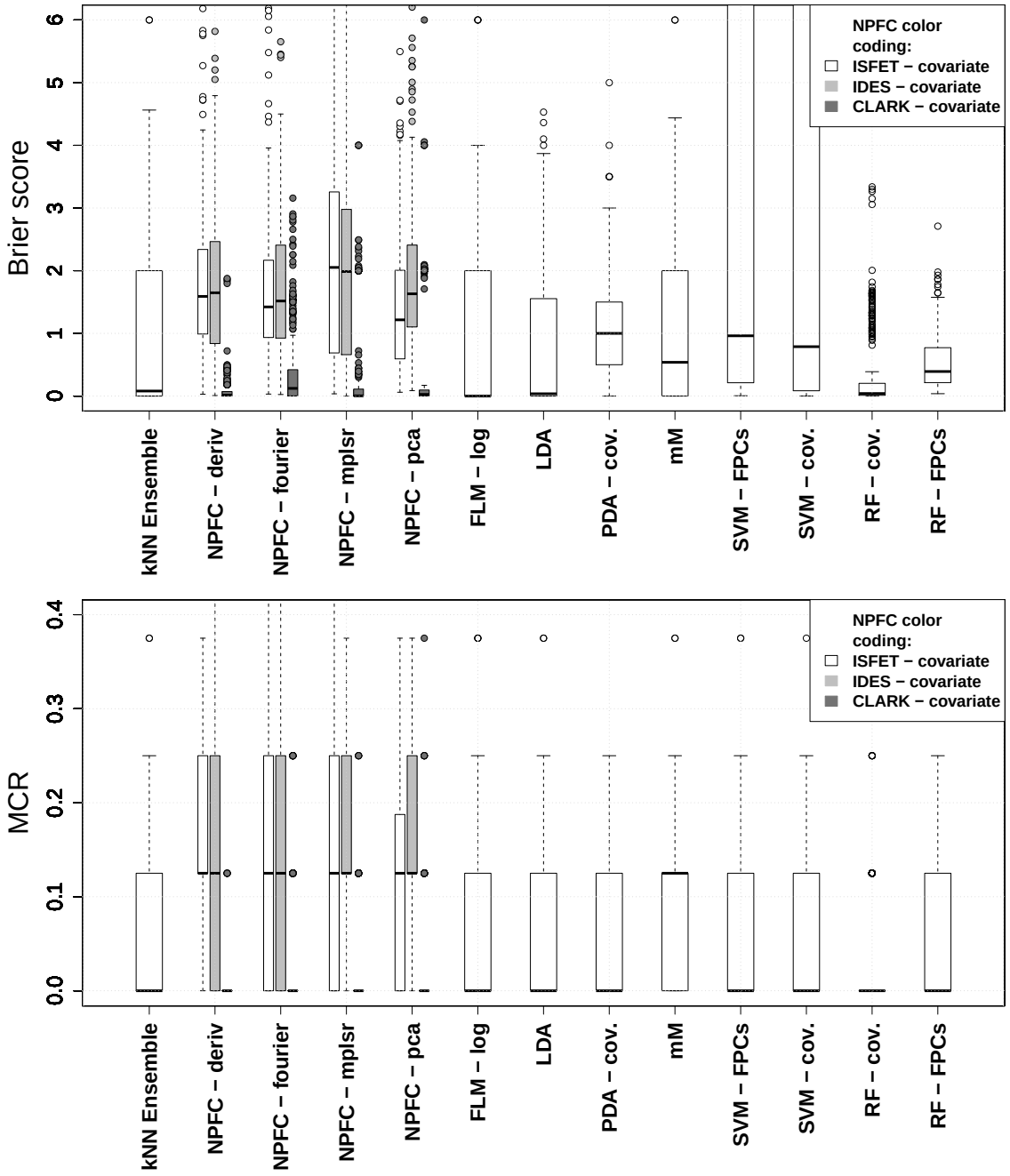


Figure 3.9: Validation results of the cell chip data for all classification approaches on basis of 25 replications of a 15-fold CV. The upper panel shows the Brier scores, the lower panel the misclassification rates (MCR).

Method	mean Brier score			mean MCR		
kNN Ensemble	0.709			0.183		
FLM-log	1.10			0.069		
LDA	0.7			0.054		
PDA	1.12			0.043		
mM	1.22			0.078		
SVM-FPCs	4.93			0.061		
SVM-cov.	5.01			0.041		
RF-cov.	0.33			0.025		
RF-FPCs	0.53			0.037		
	ISFET	IDES	CLARK	ISFET	IDES	CLARK
NPFC-deriv	1.78	1.82	**0.23**	0.169	0.147	**0.002**
NPFC-Fourier	1.66	1.73	0.52	0.134	0.147	0.025
NPFC-mplsr	2.26	2.06	0.46	0.157	0.148	0.017
NPFC-pca	1.51	1.87	0.57	0.117	0.175	0.025

Table 3.7: The mean values of both, Brier scores and misclassification rates (MCR), for the cell chip validation data, comparing all competing classification methods. The best results are highlighted by bold numbers.

3.5 Application to Real World Data – Gas Sensor Data

The challenge in the development of semiconductor gas sensors of most sensor types nowadays is to reduce cross-sensitivities and improve selectivity. Even with modern sensors, it is often difficult to differ chemically between similar gases with classical data analysis. In this section we will examine how well our functional ensemble works on this discrimination problem.

Our data is obtained from four (identically constructed) metal oxide gas sensors with a tin oxide based sensitive layer. Each sensor is operated at temperature cycling mode. The temperature changes can be identified as steps in the gas sensor signals (see also Figure 3.1, page 59, middle and lower panels). We expose the sensors successively to various gases of different concentrations, always superimposed and in turn with synthetic air (SA) (40% r.h.), see upper panel of Figure 3.1. In the middle panel, the means of the sensor signals per sensor, gas species and gas concentration are depicted. The measurements are recorded at a grid of 89 equidistant data points over time. For more details, please see Chapter 1 and Appendix D. The third panel again shows the mean signals of the fourth gas sensor, in addition with the function $\phi_\tau(t)$ in $d_{a\tau}^{Scan}(\cdot,\cdot)$ (light gray, dotted lines) and the impact points t_q used in $d_a^{Points}(\cdot,\cdot)$ (black boxes).

We use 43 measurements of each of the three highest concentrations of pentanal, ethanol, acetaldehyde, acetone, and NO_2, 43 measurements of the two concentrations of CO, and $n = 129$ measurements of SA, resulting in $N = 860$ observations per sensor.

3.5.1 Results

For the sake of computational time, we build the arithmetic mean per measurement of all four sensors. From this, the coefficients of ensemble (3.3) were estimated. The parameters used for the semi-metrics are the numbers of nearest neighbors $k \in \mathcal{K} = \{1, 5, 11, 21\}$ and orders of derivation $a \in \{0, 1, 2\}$. The choices of $\mathbb{D}_{small}$, $\mathbb{D}_1$, $\mathbb{D}_2$, t_q and τ reflect the plateaus of constant temperature and the changeovers between the heating steps. For semi-metric $d^{shortEucl}_{a,\mathbb{D}_{small}}(\cdot,\cdot)$, one of the intervals $[t_1, t_{10}]$, $[t_{11}, t_{15}]$, $[t_{16}, t_{28}]$, $[t_{29}, t_{35}]$, $[t_{36}, t_{48}]$, $[t_{49}, t_{54}]$, $[t_{55}, t_{68}]$, $[t_{69}, t_{74}]$, $[t_{75}, t_{89}]$ and $[t_{28}, t_{75}]$ is used for $\mathbb{D}_{small}$; for semi-metric $d^{Jump}_{no}(\cdot,\cdot)$, one of the sets $\{t_{11}, t_{14}\}$, $\{t_{29}, t_{35}\}$, $\{t_{49}, t_{53}\}$ or $\{t_{69}, t_{73}\}$ is used for $\{t_n, t_o\}$; for semi-metric $d^{relAreas}_{a}(\cdot,\cdot)$, $\mathbb{D}_1$ is one of the intervals $[t_{11}, t_{15}]$, $[t_{29}, t_{35}]$, $[t_{49}, t_{54}]$, $[t_{69}, t_{74}]$ and $\mathbb{D}_2 = [t_{16}, t_{28}]$; for semi-metric $d^{Points}_{a}(\cdot,\cdot)$, an equidistant grid $t_q = t_{mQ/10}$, $m = 1, \dots 10$, is used; and the function $\phi_\tau(t) = \left(\frac{\max\left(\mathbb{X}^{(a)}(t)\right)}{\max(\phi_{1,\tau}(t))}\right)\phi_{1,\tau}(t)$ with $\max\left(\mathbb{X}^{(a)}(t)\right) := \max\left(\left\{\max\left(x_i^{(a)}(t)\right), i = 1, \dots N\right\}\right)$, $\phi_{1,\tau}(t) = \frac{1}{\sqrt{2\pi}\sigma}\exp^{-\frac{1}{2}\left(\frac{t-\tau}{\sigma}\right)^2}$, $\sigma = 2.5$ and $\tau \in \{11, 23, 33, 43, 51, 63, 70, 83\}$ is used for semi-metric $d^{Scan}_{a\tau}(\cdot,\cdot)$.
The calculation is performed on both, the original data and its logarithm. The logarithm of the data is frequently analysed in the gas sensor community and is thus included in our study.

Again, only few of the 696 coefficients are estimated to have values above zero. As can be seen from Table 3.8, the coefficients with the six highest values correspond to the semi-metrics $d^{Eucl}_{a}(\cdot,\cdot)$, $d^{shortEucl}_{a,\mathbb{D}_{small}}(\cdot,\cdot)$ and $d^{Scan}_{a\tau}(\cdot,\cdot)$, and especially to the first, second and last step as well as the second and third plateau in the signals around the data points t_{11} - t_{15}, t_{70}, t_{28} - t_{75} and t_{29} - t_{35}. These results indicate that these curve parts yield the most information for the discrimination task. The fact that the whole signal part instead of the jump height itself (represented by the semi-metric $d^{Jump}_{no}(\cdot,\cdot)$) is determined to be important is consistent with the underlying physical mechanisms in the sensitive layer during changes in temperature. From a physical point of view, a change in temperature goes inline with a quantity of de- and adsorption reactions in the sensitive layer, until the energy potentials stabilize. Since these reactions depend, among other things, on the combination of temperature, sensitive layer properties and the gas species applied, they should also be specific for the latter. Thus, signal regions containing temperature changes, i.e., de- and adsorption reactions, are expected to yield discriminative power.
There is only a small overlap in the sets of coefficients c_l estimated for the original and logarithmic data. However, the signal ranges assigned to these c_l are consistent. This confirms the intepretability of the ensemble coefficients.

To assess the performance of our model and compare its prediction accuracy to other approaches, the data was split randomly $W = 15$ times into 14 parts of the size 57 and one of size 62 to perform 15-fold CV. Figure 3.10 shows the coefficient selection perfor-

	IDs of estimated coefficients	parameter tuple
original data	34	$\{d^{shortEucl}$ with $x_i(t)$ centered, $a = 0,\ k = 1\}$, $\mathbb{D}_{small} = [t_{11}, t_{15}]$
	92	$\{d^{shortEucl}$ with $x_i(t)$ centered, $a = 0,\ k = 5\}$, $\mathbb{D}_{small} = [t_{11}, t_{15}]$
	51	$\{d^{Scan}$ with $x_i(t)$ centered, $a = 0,\ k = 1\}$, $\tau = 11$
	32	$\{d^{Eucl}$ with $x_i(t)$ centered, $a = 0,\ k = 1\}$
	57	$\{d^{Scan}$ with $x_i(t)$ centered, $a = 0,\ k = 1\}$, $\tau = 70$
	109	$\{d^{Scan}$ with $x_i(t)$ centered, $a = 0,\ k = 5\}$, $\tau = 11$
logarithmic data	51	$\{d^{Scan}$ with $x_i(t)$ centered, $a = 0,\ k = 1\}$, $\tau = 11$
	32	$\{d^{Eucl}$ with $x_i(t)$ centered, $a = 0,\ k = 1\}$
	42	$\{d^{shortEucl}$ with $x_i(t)$ centered, $a = 0,\ k = 1\}$, $\mathbb{D}_{small} = [t_{28}, t_{75}]$
	235	$\{d^{shortEucl}$, $a = 1,\ k = 1\}$, $\mathbb{D}_{small} = [t_{11}, t_{15}]$
	266	$\{d^{shortEucl}$ with $x_i(t)$ centered, $a = 1,\ k = 1\}$, $\mathbb{D}_{small} = [t_{11}, t_{15}]$
	94	$\{d^{shortEucl}$ with $x_i(t)$ centered, $a = 0,\ k = 5\}$, $\mathbb{D}_{small} = [t_{29}, t_{35}]$

Table 3.8: Selection results for the two preprocessing variants (left column). Middle column: the numbers of the six estimated coefficients that show the largest values (in decreasing order). On the right, the chosen ensemble coefficients are decoded; the value of a indicates the order of derivation, k indicates the number of nearest neighbors used.

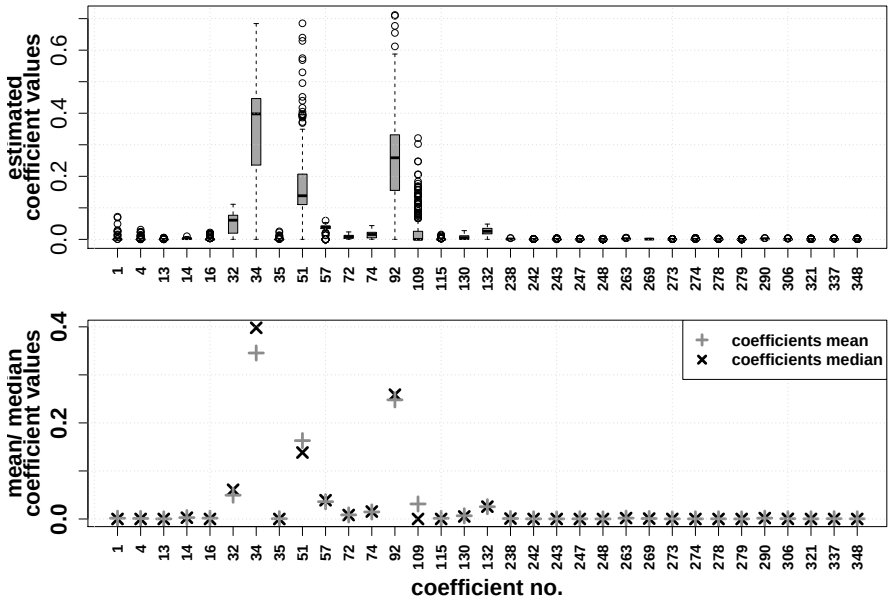
Method	mean Brier score		mean MCR	
	orig.	log.	orig.	log.
kNN Ensemble	15.67	12.09	0.174	0.116
NPFC-deriv	2.02	1.52	0.018	0.016
NPFC-Fourier	2.44	**0.39**	0.027	**0.004**
NPFC-mplsr	6.43	3.35	0.087	0.032
NPFC-pca	**0.88**	0.95	**0.007**	0.01
LDA	42.68	44.15	0.549	0.735
PDA	10.55	24.56	0.112	0.284
mM	49.14	49.14	0.851	0.859
SVM-FPCs	62.75	66.57	0.345	0.328
SVM-cov.	49.25	68.51	0.914	0.301
RF-cov.	4.66	4.64	0.053	0.052
RF-FPCs	28.41	26.28	0.336	0.313

Table 3.9: The mean values of both, Brier scores and misclassification rates, for the original ('orig.') and logarithmic ('log.') gas data application, comparing all competing classification methods. The best results are highlighted by bold numbers.

mance for the data. In the two upper panels, estimation results for the original data are shown: the first panel gives the estimated coefficients c_l that have been estimated to be of mean values above $1 \cdot 10^{-4}$, plotted as boxplots across 15 replications of a 15-fold CV. The second panel shows the mean (gray pluses) and median (black crosses) values of these coefficients. In the third and fourth panel, the same is shown for the logarithmic data. For both preprocessings, the coefficient estimation is quite stable under subsampling.

The global Brier scores and misclassification rates of the validation data can be found in Figure 3.11, where the results of our functional k-nearest-neighbor ensemble are presented as the first boxes, while the other boxes show the validation results of the other classification approaches. Table 3.9 gives the mean values for both performance measures for all approaches. The optimal parameters per semi-metric of the NPFC approach are chosen via minimization of the mean prediction error of a 10-fold CV.
As before, most multivariate approaches applied to the functional principal component scores are not performing as well as functional classification approaches. Our ensemble works similarly well on the validation data sets of the original and logarithmic data. Both classification performance measures are small, but random forests applied to the discretized covariates and the NPFC approach yield better results. However, the advantage of our k-nearest-neighbor ensemble remains, yielding interpretability achieved by the coefficient estimation.

(a)



(b)

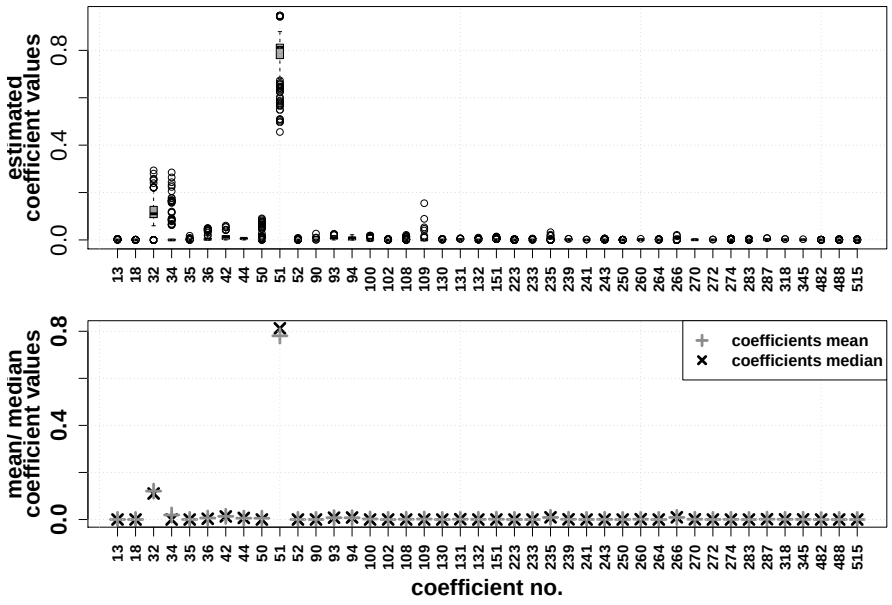


Figure 3.10: (a) First panel: Estimated coefficients as boxplots across 15 replications of a 15-fold CV when using the original gas sensor data. For clarity, only those coefficients with a mean value above $1 \cdot 10^{-4}$ are shown. Second panel: The mean and median values of these coefficients. (b) The same, but estimation is based on the logarithm of the gas sensor data as covariates.

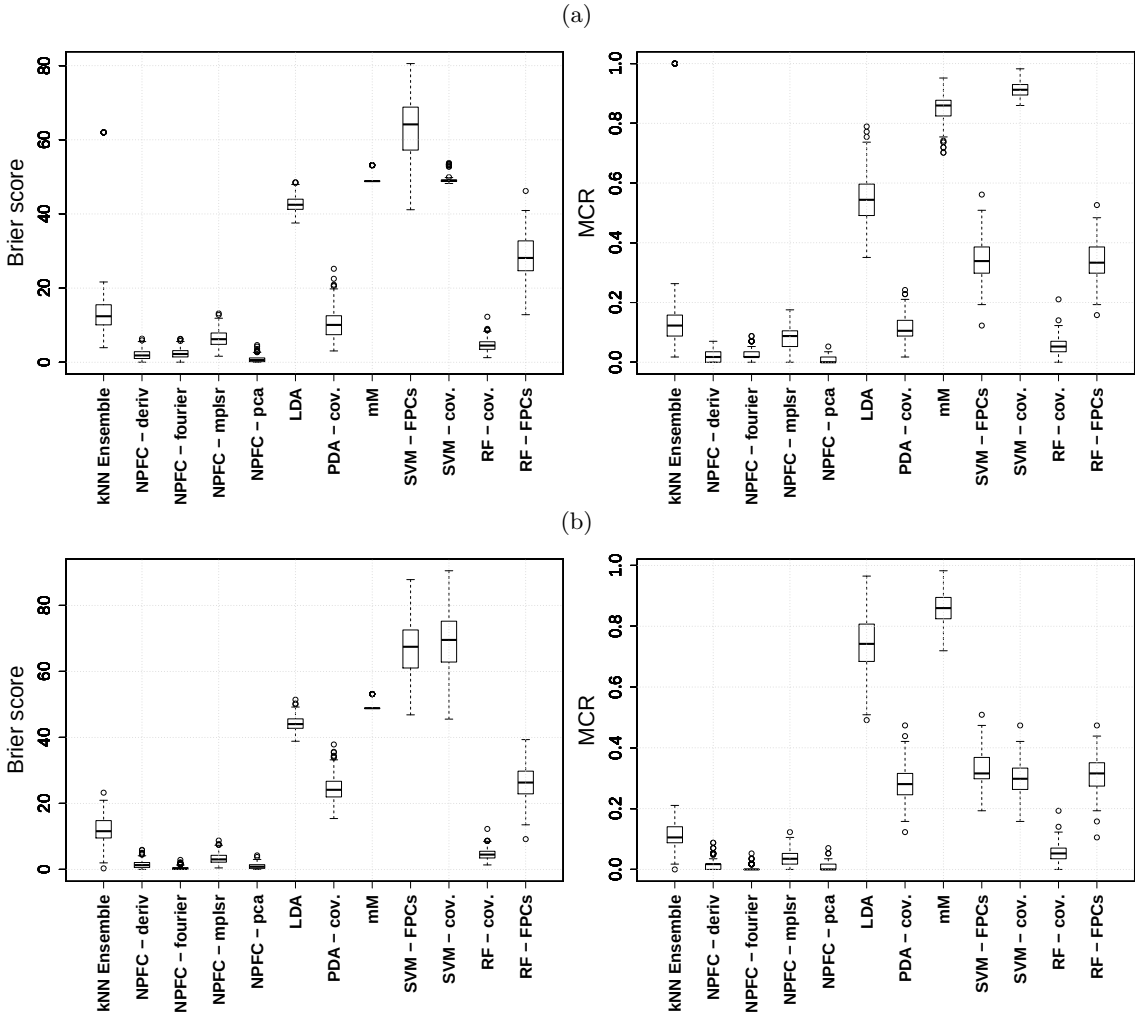


Figure 3.11: (a) Validation results for the original gas sensor data, for all classification approaches on basis of 15 replications of a 15-fold CV. The left panel shows the Brier scores, the right panel the misclassification rates. (b) The same for the logarithmic data.

3.6 Application to Real World Data – Phoneme Data

A popular classification problem in the functional data context is the classification of phoneme data, which was studied for example in Hastie et al. (1995), Ferraty and Vieu (2003), Epifanio (2008), or Li and Yu (2008). The main aim is to discriminate between the log-periodograms of five phonemes, namely "aa", "ao", "dcl", "iy", and "sh". For more information, please see Hastie et al. (1995). Three exemplary log-periodograms of each phoneme can be found in Figure 3.12.
From the 4509 available* log-periodograms, 50 samples are drawn randomly. The samples are divided into training samples containing 150 curves per class and test samples with 250 curves per class. All competing methods were applied to the same sample sets. All semi-metrics listed in Table 3.1 were used. The respective parameters were chosen arbitrarily, since no detailed background knowledge is available. With respect to the resulting 816 k-nearest-neighbor ensemble coefficients, those yielding the highest estimated values correspond to tuples that include the semi-metrics $d_{a=0}^{Eucl}(\cdot,\cdot)$ and $d_{a=0,\mathbb{D}_{small}}^{shortEucl}(\cdot,\cdot)$, $\mathbb{D}_{small} = [30, 65]$, with $k \in \{5, 11, 21\}$. Thus, the log-periodograms' Euclidian distances seem to contain most discriminative power. Figure 3.13 shows the classification results for the validation data, Table 3.10 the mean Brier scores and MCRs. As can be seen, the performance of the NPFC approach depends considerably on the choice of the semi-metric used. Apart from that, the various methods perform comparable, except for the Brier score values of the SVM method, which are not shown in Figure 3.13 due to y-axis pruning. The k-nearest-neighbor ensemble, *NPFC-mplsr* and random forests yield slightly superior results.

*The data was taken from the `R` package `ElemStatLearn`, Hastie et al. (2015a)

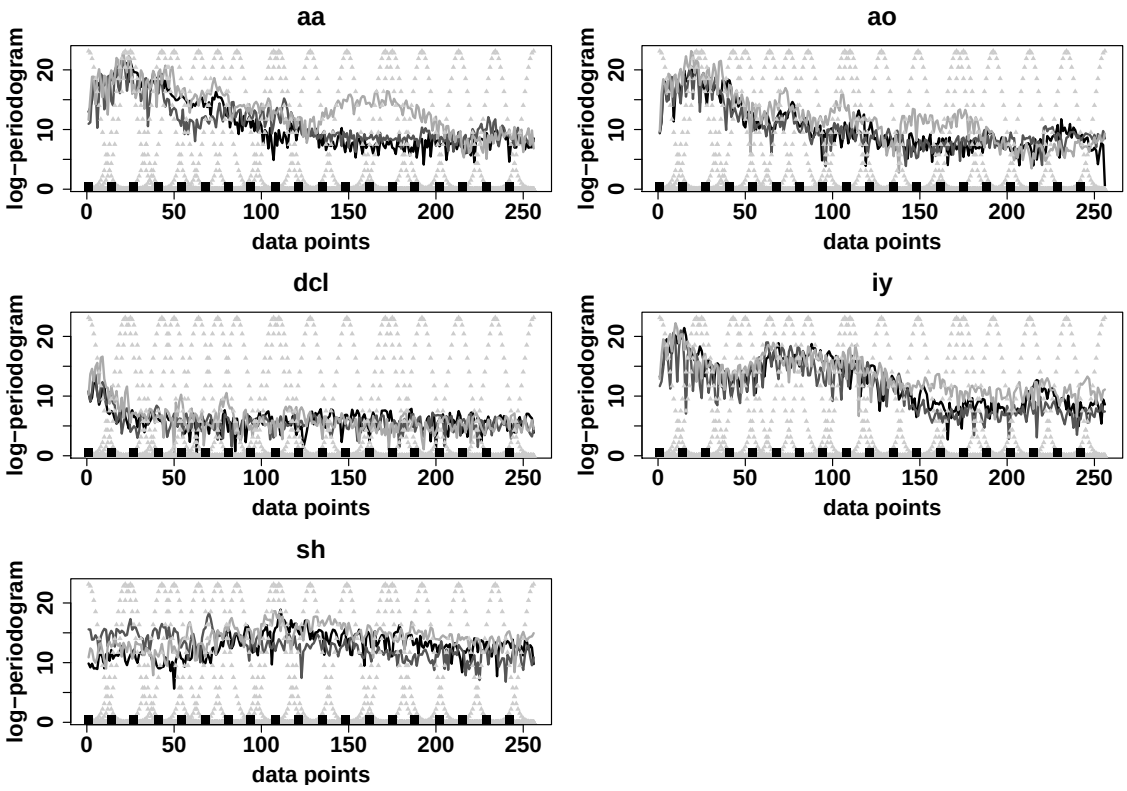


Figure 3.12: The panels show $N = 3$ log-periodograms of each phoneme. The function $\phi_\tau(t)$ used in $d_{a\tau}^{Scan}(\cdot,\cdot)$ is depicted as light gray, dotted lines, the impact points t_q used in $d_a^{Points}(\cdot,\cdot)$ as black boxes.

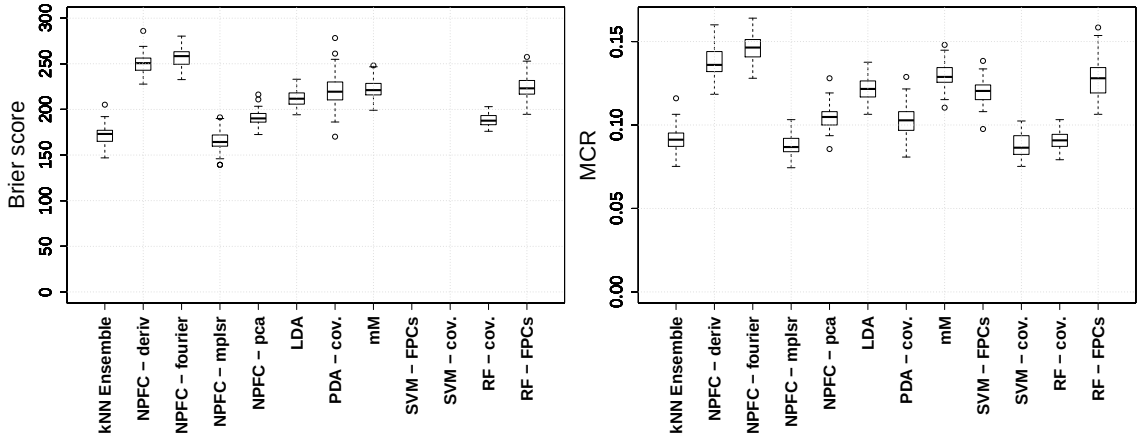


Figure 3.13: Results for $N_{val} = 250$ test observations per class. The models were estimated 50 times with sample sizes of $N = 150$ per phoneme. The left panel shows the Brier scores, where the boxes of the SVM-FPCs and the SVM-cov. methods (mean values 1769.47 and 1817.73) are not shown due to y-axis pruning. The right panel shows the misclassification rates (MCR).

Method	mean Brier score	mean MCR
kNN Ensemble	172.58	0.091
NPFC-deriv	250.87	0.138
NPFC-Fourier	257.46	0.146
NPFC-mplsr	**165.29**	**0.087**
NPFC-pca	190.98	0.105
LDA	212.07	0.122
PDA	221.62	0.103
mM	223.29	0.129
SVM-FPCs	1769.47	0.119
SVM-cov.	1817.73	0.088
RF-cov.	187.79	0.091
RF-FPCs	224.32	0.128

Table 3.10: The mean values of both, Brier scores and misclassification rates (MCR), for the phoneme data, comparing all competing classification methods. The best results are highlighted by bold numbers.

3.7 Conclusion and Outlook

We introduced a functional k-nearest-neighbor ensemble that allows for automatic feature and, depending on the data at hand, variable selection. For that purpose, a set of semi-metrics was defined. Here, each semi-metric focused on a specific feature of the functional covariates. Additionally, sets of numbers of nearest neighbors and of orders of derivation of the covariates were defined. A particular combination of a semi-metric, a number of nearest neighbors and an order of derivation made up a parameter tuple. The ensemble members were then calculated by a k-nearest-neighbor approach and the leave-one-out technique, using a specific tuple. Each ensemble member was weighted by an unknown coefficient. These coefficients were estimated such that the global Brier score was minimized. A constraint put on the coefficients yielded an implicit (positive) Lasso-type penalty, such that some coefficients were estimated to be exactly zero. Zero-valued coefficients mean that the respective ensemble member, i.e., a certain tuple, has a weight of zero. Thus, an automatic feature selection was performed during the estimation process. In the case of multiple functional (and non-functional) covariates, the parameter tuple can also include the covariate type, such that the ensemble allows for additional variable selection.
Our ensemble presents a flexible and powerful tool for the classification of all sorts of functional (in addition with non-functional) data. While the predictive classification performance heavily depends on the data at hand and ranges, compared to alternative classification approaches, from very good to hardly competitive, the automatic and interpretable feature selection is an important advantage compared to other discrimination methods. In simulation studies, it was shown that even a set of essentially arbitrarily chosen semi-metrics yields excellent predictions and sensible results in terms of interpretability. In the cell chip data application, the prediction performance was competitive to the alternative methods. The feature and variable selection here was outstanding since the estimated coefficients ideally agreed with the biological background knowledge. The coefficient selection of the gas sensor application also was consistent with the background knowledge of the data. Although the prediction accuracy of the k-nearest-neighbor ensemble was good, here, the method was outperformed by the NPFC and random forest approaches. The results of the phoneme data indicate that especially the curves' Euclidian distances are relevant for the classification.
We can conclude that the results of the cell chip data as a binomial and the gas sensor and phoneme data representing multi-class discrimination problems confirm that, in real world data, a) the functional k-nearest-neighbor ensemble is good concerning the prediction performance itself, and b) gives additional insight in the data, by weighting parts of the data that really yield discriminative power.
Despite these results, it should be mentioned that care has to be taken when interpreting the coefficients. Sometimes, columns of the probability matrix $\boldsymbol{P}$ (analogously, $\boldsymbol{P}_v$) might be identical to the coded true response vector $\boldsymbol{z}$, which we call "mirroring effect". This means that, for one or a few tuples $\{d(\cdot,\cdot), a, k\}$ ($\{d_v(\cdot,\cdot), a, k\}$) with probabilities $\hat{\pi}_{ig(l)}$

$(\hat{\pi}_{igv(l)})$, one has

$$z_{ig} \equiv \hat{\pi}_{ig(l)} \qquad \forall i, g.$$

Consider an example with $N = 2$ observations of $G = 2$ classes, such that $\boldsymbol{z} = (1, 0, 0, 1)^T$. Let us further consider three tuples yielding the probability matrix

$$\boldsymbol{P} = \begin{pmatrix} \hat{\pi}_{11(1)} & \hat{\pi}_{11(2)} & \hat{\pi}_{11(3)} \\ \hat{\pi}_{12(1)} & \hat{\pi}_{12(2)} & \hat{\pi}_{12(3)} \\ \hat{\pi}_{21(1)} & \hat{\pi}_{21(2)} & \hat{\pi}_{21(3)} \\ \hat{\pi}_{22(1)} & \hat{\pi}_{22(2)} & \hat{\pi}_{22(3)} \end{pmatrix} = \begin{pmatrix} 1 & \hat{\pi}_{11(2)} & \hat{\pi}_{11(3)} \\ 0 & \hat{\pi}_{12(2)} & \hat{\pi}_{12(3)} \\ 0 & \hat{\pi}_{21(2)} & \hat{\pi}_{21(3)} \\ 1 & \hat{\pi}_{22(2)} & \hat{\pi}_{22(3)} \end{pmatrix}.$$

Additionally, in the two-class case, it is always $\hat{\pi}_{ig_1(l)} = 1 - \hat{\pi}_{ig_2(l)} \; \forall \; i, g_1, g_2, l$, with $g_1, g_2 \in \{1, 2\}$, $g_1 \neq g_2$. Recall that constraint (3.4),

$$c_l \geq 0 \; \forall l, \; \textstyle\sum_{l=1}^{p} c_l = 1,$$

should hold when minizing the global Brier score (3.6),

$$Q(\boldsymbol{c}) = \left(\underset{NG\times 1}{\boldsymbol{z}} - \underset{NG\times pp\times 1}{\boldsymbol{P}}\,\boldsymbol{c} \right)^T \left(\underset{NG\times 1}{\boldsymbol{z}} - \underset{NG\times pp\times 1}{\boldsymbol{P}}\,\boldsymbol{c} \right).$$

This obviously implies the solution

$$c_1 = 1 \text{ and } c_2 = c_3 = 0.$$

If, instead of the above probabilities, we consider

$$\boldsymbol{P} = \begin{pmatrix} 1 & \hat{\pi}_{11(2)} & 1 \\ 0 & \hat{\pi}_{12(2)} & 0 \\ 0 & \hat{\pi}_{21(2)} & 0 \\ 1 & \hat{\pi}_{22(2)} & 1 \end{pmatrix}$$

the estimation becomes to some extend arbitrary, since all solutions

$$c_1 \in [0, 1], \; c_2 = 0, \; c_3 = 1 - c_1$$

minimize the global Brier score while at the same time fulfilling constraint (3.4). This means that, while the selection of coefficients unequal to zero itself still reflects tuples $\{d(\cdot, \cdot), a, k\}$ ($\{d_v(\cdot, \cdot), a, k\}$) with discriminative power, no information can any longer be derived from the coefficients' magnitudes.
In our experience, such effects occur especially for binary classification problems, and for coefficients being assigned to tuples with $k = 1$, see also Tables B.1 and B.2 in Appendix B.3 for respective results of the examined data sets.

Thanks to the flexibility of the functional k-nearest-neighbor ensemble, there is even ample room for extension. Of course, the ensemble members are not limited to members that are

based on distance measures. Other members could be included, for example the results of basic models such as a functional linear model. Also, the ensemble could be adapted to regression problems by a suitable modification of the underlying k-nearest-neighbor method and the optimization criterion. One could think about further developments in the direction of time series analysis.
The implicit (positive) Lasso-type penalty imposed on the ensemble coefficients c_l by the constraints $c_l \geq 0 \ \forall l$ and $\sum_{l=1}^{p} c_l = 1$ could be relaxed by altering the second condition to $\sum_{l=1}^{p} c_l = \rho$. The estimation, thus, would be obtained by a Lasso-type estimator with tuning parameter ρ which controls the sparseness of the ensemble.
In the analyses provided, a certain degree of smoothness of the functional data is implicitly assumed in the semi-metrics used. If this is not the case for the data at hand, some preprocessing is advisable. Options are for example to approximate the data by decomposition and to build a semi-metric based on the decomposition, or use smoothed signals in a semi-metric.
Although our approach provides built-in feature selection, the question which set of semi-metrics and respective parameters should be used must still be answered by the user. If background knowledge on the data is provided, the semi-metrics can be chosen adequately, as has been done with the cell chip and gas sensor data. If no such knowledge is available, one can, for example, use a very large set of semi-metrics and let the data decide using the proposed ensemble with built-in feature selection. If the set of potential semi-metrics becomes too large, similar to a random forest approach, the semi-metrics and parameters actually used in the ensemble could be subsets randomly drawn from the predefined sets. This procedure could be repeated until a model choice criterium is fulfilled.

All calculations were performed on a machine with 256 GB RAM and four AMD Opteron CPUs of 12 cores and 2.2 GHz, with software R version 3.1.0 (R Core Team, 2017) and the add-on packages mentioned above.

Chapter 4

Classification of Functional Data with k-Nearest-Neighbor Ensembles by Fitting Constrained Multinomial Logit Models

4.1 The Multinomial Model and the Lasso

This chapter focuses on an alternative estimation approach to calculate the coefficients of the functional k-nearest-neighbor ensemble introduced in Chapter 3. The functional k-nearest-neighbor ensemble is set up analogously to the previous chapter. The respective single posterior probabilities are then preprocessed and used as inputs in a penalized and constrained multinomial logit model (cMLM). Thus, the previous constraint put on the ensemble coefficients can be relaxed, as explained later in this chapter.
Examples for functional predictors are given by the data motivating our approach, see also Chapter 1 and Appendix D. Here, we use the phoneme data introduced by Hastie et al. (1995) and the cell chip data. The findings concerning the mirroring effects discussed in the conclusion of Chapter 3 show that some of the estimated posterior probabilities calculated from the CLARK data are strongly related to the binary response "paracetamol" or "no paracetamol". That is why, in this chapter, the CLARK-signals are excluded from the cell chip measurements. The remaining data is shown in Figure 4.1.
The classification task in this data set again results from disturbing the cells' habitual environment: While they are usually kept in nutrient medium, one can add test substances, as, for example, paracetamol (short: AAP), to this medium and monitor the cells' reactions. A functional two-class discrimination task is then given by comparing ion sensitive field effect transistor (ISFET) and interdigitated electrode structure (IDES) curves of measurements with (gray curves in Figure 4.1) and without (black curves) adding AAP to the nutrient

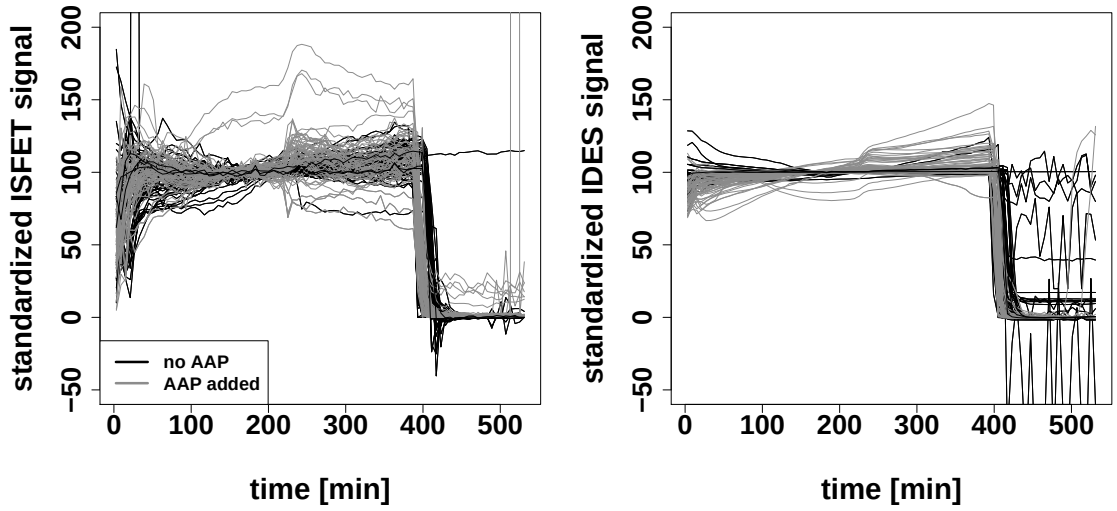


Figure 4.1: A total of $n = 120$ standardized ISFET- and IDES-signals, recorded over time. Gray scales refer to the two classes of the discrimination task, with gray curves representing measurements with, and black curves measurements without the test substance AAP.

medium.

As mentioned, we introduce a novel interpretable feature selection method for classifying functional data, where the estimation of the functional $\underline{k}$-$\underline{n}$earest-$\underline{n}$eighbor $\underline{e}$nsemble (kNNE) is carried out by a penalized and constrained multinomial logit model. Analogously to before, the kNNE is set up by a large number of posterior probabilities for class membership that represent the ensemble members. Each posterior probability depends on the k neighbors relative to the observation that is to be classified, as well as on a chosen semi-metric. As was shown by Ferraty and Vieu (2006), semi-metrics are a suitable mathematical formulation to capture certain characteristics of a curve or function. Thus, by using adequate semi-metrics, a large variety of curve features, for example the curves' maxima or their mean values, can be included in the kNNE. The ensemble combines its members in a linear way, and each posterior probability, i.e. ensemble member, is multiplied by an unknown coefficient which has to be determined. This yields the possibility to let members have differing importances. A multinomial logit link is then applied to these ensemble members to obtain a classification model.

The ensemble coefficients are estimated by a cMLM, combined with a Lasso-type penalty. MLM are special GLM, employing a logit link on a multinomial distributed response. The covariates can, in principle, belong to any scale of measure. Let $Y \in \{1, \dots G\}$ denote

a nominal, individual-level response, and $\boldsymbol{x} = (1, x_1, \dots x_k)^T$ and $\boldsymbol{\nu}_g = (\nu_{g1}, \dots \nu_{gp})^T$, $g \in \{1, \dots G\}$, vectors of scalar covariates. Then, an ordinary MLM has the form

$$P\left(Y = g|\boldsymbol{x}, \{\boldsymbol{\nu}_g\}\right) =: \pi_g = \frac{\exp(\eta_g)}{\sum_{s=1}^{G} \exp(\eta_s)}, \tag{4.1}$$

with the linear predictor

$$\eta_g = \boldsymbol{x}^T \boldsymbol{\beta}_g + \boldsymbol{\nu}_g^T \boldsymbol{c}$$

and unknown model parameters $\boldsymbol{\beta}_g = (\beta_{g0}, \beta_{g1}, \dots \beta_{gk})^T$ and $\boldsymbol{c} = (c_1, \dots c_p)^T$. Here, the first term comprises class-specific effects, while the second term includes class-specific covariates.

We will use the single posterior probability estimates from the k-nearest-neighbor ensemble as inputs in the MLM. Since these probabilities are class-specific, the ordinary MLM (4.1) is reduced to

$$\pi_g = \frac{\exp(\tilde{\eta}_g)}{\sum_{g=1}^{G} \exp(\tilde{\eta}_g)} \text{ , with } \tilde{\eta}_g = \boldsymbol{\nu}_g^T \boldsymbol{c}. \tag{4.2}$$

Thus, we consider a special case of MLM comprising solely class-specific covariates.

Naturally, for probabilities $\sum_{g=1}^{G} \pi_g = 1$ has to hold. This implies that there is redundancy in Model (4.2). The model is not identifiable, i.e. the parameter $\boldsymbol{c}$ can not be estimated uniquely. To resolve this problem, an identifiability constraint can be put on $\boldsymbol{c}$. In the following, we will define a constraint by choosing G to be the so-called reference class. Then, Model (4.2) can be formulated as

$$\pi_g = \begin{cases} \frac{1}{1+\sum_{s=1}^{G-1} \exp(\tilde{\eta}_s)} = 1 - \sum_{s=1}^{G-1} \pi_s & \text{if } g = G \\ \\ \frac{\exp(\tilde{\eta}_g)}{1+\sum_{s=1}^{G-1} \exp(\tilde{\eta}_s)} & \text{else.} \end{cases} \tag{4.3}$$

Since the kNNE yields a multitude of single posterior probability estimates, i.e. $p >> n$ in most data situations, with n denoting the number of observations, Model (4.3) is penalized to abet stable estimation (see also Section 4.2.2).

Lasso penalization allows for sparse results, estimating most coefficients equal to zero, as shown in Tibshirani (1996). Various Lasso-type penalties, suited for MLM, have been developed (see Argyriou et al., 2007; Simon et al., 2013; Chen and Li, 2013; Vincent and Hansen, 2014, among others). Apart from the standard global Lasso penalty (Tibshirani, 1996), we also employ the category-specific Lasso and the categorically structured (CATS) Lasso penalty (Tutz et al., 2015) for multi-class applications. The main advantage of using a Lasso-type penalty is the sparse estimation result enabling feature selection. Additionally we extend an ordinary penalized MLM such that a non-negativity constraint is put on the ensemble coefficients. This constraint ensures a proportional interpretability between the

selected features with regard to the data background. As an example, consider two potentially interesting features in the above cell chip data, the step most curves exhibit around 220 minutes, and the distances between measurement curves in the region from about 220 to 400 minutes. By including corresponding semi-metrics in the kNNE and estimating the assigned coefficients by the penalized cMLM, one can decide whether the step or rather the curve distances yield more discriminative power.

As mentioned, several functional classification approaches have been developed in recent years, as lately in Zhu et al. (2010), Delaigle and Hall (2012), or Nguyen et al. (2016). Some of them also use ensemble methods that combine scores or features of some kind in a linear combination, as was discussed in the previous chapter. Semi-metrics in the context of functional data classification were used for example in Ferraty and Vieu (2003) and Alonso et al. (2012). All these approaches use a pre-defined and limited number of features that are given as input to a classification algorithm, and usually use the observed curve across its whole domain. This also holds for Matsui (2014), who use a functional logistic regression model to differ between functional observations of different classes, focusing on variable selection. In contrast, our approach can handle a very large number of semi-metrics, i.e. features, in the functional kNNE. This includes taking only specific parts of the curves into account, which might be more significant in certain data situations than using the whole curve, as the cell chip data suggests. Additionally, for most functional classification methods, the extension to multiple functional covariates is not straight forward, and the estimation results are not interpretable with regard to the data background. Interpretability also is the most weak point in the method by Möller et al. (2016), who built random forests out of summary quantities calculated from (randomly chosen) curve segments. The approach presented here offers both, the potential inclusion of multiple functional (and non-functional) covariates as well as interpretability. These two advantages are also present in the method of Fuchs et al. (2015a), see Chapter 3. But the estimation of the ensemble coefficients by means of Brier score minimization implies a more restrictive constraint on the coefficients than is necessary with the estimation via a penalized cMLM. Further advantages that arise from the use of a MLM are that MLM can readily be adapted to ordinal data, and estimation is fast and stable. The penalty might be chosen with respect to probable interrelations between the coefficients, such as the CATS Lasso penalty.

The remainder of this chapter is organized as follows: In Section 4.2, our approach is introduced in detail. The numerical experiments from Chapter 3, Section 3.3.2, are re-estimated in Section 4.3 with the new estimation approach. The respective results of both approaches are compared. In Sections 4.4 and 4.5, the performance of the presented approach is analysed with respect to the cell chip and phoneme data sets, and is compared to other (functional) classification approaches. Also, the differences and similarities relative to the results from the estimation via minimizing the Brier score are evaluated. The

manuscript closes with a discussion of the results and further possible extensions concerning our method in Section 4.6.
We provide exemplarily code as well as the respective data in the online supplement of Fuchs et al. (2016) to make the application results fully reproducible.

4.2 Method

In this section, the proposed approach is given in detail. First, the functional kNN ensemble setup is described, with a focus on the basic design and the semi-metrics that capture specific curve features. The following Section 4.2.2 introduces penalized cMLM and various Lasso-type penalties. Section 4.2.3 outlines details of the estimation of our penalized cMLM model. In Section 4.2.4, a short description of alternative classification techniques is given, and the Brier score and misclassification rate are defined as performance measures to explore the predictive capability of the single methods. An additional feature importance measure is introduced, allowing for the proportional interpretability of our method's selection results.

4.2.1 Functional Nearest Neighbor Ensembles

Let $x_i(t)$, $i = 1, \dots n$, be a set of curves, i.e. functional predictors, with t from a domain $\mathbb{D} \subseteq \mathbb{R}$. Assume that each curve belongs to one of G different classes, denoted by $y_i \in \{1, \dots G\}$. Further, let $x_i^{(a)}(t)$ denote the ath derivative of the functional covariate $x_i(t)$, and $d\left(x_i^{(a)}(t), x_j^{(a)}(t)\right)$ a semi-metric of (the derivatives of) two functional covariates $x_i^{(a)}(t)$, $x_j^{(a)}(t)$, $i \neq j$.

Translating Curve Features into Probabilities via Semi-Metrics

A simple functional k-nearest-neighbor ensemble can be specified through a set of various semi-metrics $d_l(\cdot, \cdot)$, where $l = 1, \dots p$ is the index for the different semi-metrics. Each semi-metric is chosen such that it extracts a particular feature from the curves. Apart from the semi-metrics that were already mentioned in the Introduction, representing the step or distances in the cell chip data, further examples are the covariates' maxima or curvatures. Since some curve characteristics might be amplified after derivation, we use the derivatives of the functional predictors as well as the original curves. Thus, the nearest neighbor ensemble to be considered uses semi-metrics $d_l\big(x_i^{(a)}(t), x_j^{(a)}(t)\big)$, applied to the quantitites $x_i^{(a)}(t)$, $x_j^{(a)}(t)$, $i \neq j$. The whole set of semi-metrics we use is given in Table 4.1*.

*It might enhance results if semi-metrics are chosen with respect to the data at hand. Since we will examine the same data sets, the semi-metrics of Table 4.1 correspond to those in Chapter 3, Table 3.1.

Semi-metric	Formula	The semi-metric focuses on...
$d^{Eucl}(x_i^{(a)}(t), x_j^{(a)}(t))$	$\sqrt{\int_{\mathbb{D}} \left(x_i^{(a)}(t) - x_j^{(a)}(t)\right)^2 dt}$	... the absolute distance of two curves (or their derivatives).
$d_\tau^{Scan}(x_i^{(a)}(t), x_j^{(a)}(t))$	$\sqrt{\int_{\mathbb{D}} \left(\phi_\tau(t) \left(x_i^{(a)}(t) - x_j^{(a)}(t)\right)\right)^2 dt}$, with $\tau \in \mathbb{D}$	... the absolute distances of weighted profiles of the original curves (or their derivatives), centered around τ.
$d_{\mathbb{D}_{small}}^{shortEucl}(x_i^{(a)}(t), x_j^{(a)}(t))$	$\sqrt{\int_{\mathbb{D}_{small}} \left(x_i^{(a)}(t) - x_j^{(a)}(t)\right)^2 dt}$	... the absolute distance on a limited part of the domain of definition $\mathbb{D}_{small} \subset \mathbb{D}$ of two curves (or their derivatives).
$d^{Mean}(x_i^{(a)}(t), x_j^{(a)}(t))$	$\left\lvert \int_{\mathbb{D}} x_i^{(a)}(t)dt - \int_{\mathbb{D}} x_j^{(a)}(t)dt \right\rvert$	... the similarity of mean values of the whole curves (or their derivatives).
$d^{relAreas}(x_i^{(a)}(t), x_j^{(a)}(t))$	$\left\lvert \left\lvert \frac{\int_{\mathbb{D}_1} x_i^{(a)}(t)dt}{\int_{\mathbb{D}_2} x_i^{(a)}(t)dt} \right\rvert - \left\lvert \frac{\int_{\mathbb{D}_1} x_j^{(a)}(t)dt}{\int_{\mathbb{D}_2} x_j^{(a)}(t)dt} \right\rvert \right\rvert$	... the similarity of the relation of areas on parts of the domain of definition, $\mathbb{D}_1$, $\mathbb{D}_2 \subset \mathbb{D}$.
$d_{bo}^{Jump}(x_i(t), x_j(t))$	$\left\lvert (x_i(t_b) - x_i(t_o)) - (x_j(t_b) - x_j(t_o)) \right\rvert$	... the similarity of jump heights at points t_b, $t_o \in \mathbb{D}$.
$d^{Max}(x_i^{(a)}(t), x_j^{(a)}(t))$	$\left\lvert \max\left(x_i^{(a)}(t)\right) - \max\left(x_j^{(a)}(t)\right) \right\rvert$	... the difference of the curves' (or their derivatives') global maxima.
$d^{Min}(x_i^{(a)}(t), x_j^{(a)}(t))$	$\left\lvert \min\left(x_i^{(a)}(t)\right) - \min\left(x_j^{(a)}(t)\right) \right\rvert$	... the difference of the curves' (or their derivatives') global minima.
$d^{Points}(x_i^{(a)}(t), x_j^{(a)}(t))$	$\frac{1}{E} \sum_{e=1}^{E} \left\lvert x_i^{(a)}(t_e) - x_j^{(a)}(t_e) \right\rvert$	... the differences at certain observation points t_e (also called "points of impact").

Table 4.1: Semi-metrics used to set up the k-nearest-neighbor ensemble.

After having defined adequate semi-metrics, possibly with respect to expert knowledge about the data at hand, the corresponding semi-metric dependent neighborhoods can be defined. With respect to a generic or a new observation $(y^*, x^*(t))$ and a specific semi-metric $d_l(\cdot,\cdot)$, the learning sample $(y_i, x_i^{(a)}(t))$, $i = 1, \dots n$ is ordered such that

$$d_l\big(x^{*(a)}(t), x_{(1)}^{(a)}(t)\big) \leq \ldots \leq d_l\big(x^{*(a)}(t), x_{(k)}^{(a)}(t)\big) \leq \ldots \leq d_l\big(x^{*(a)}(t), x_{(n)}^{(a)}(t)\big).$$

Hence, the $x_{(i)}^{(a)}(t)$ are the curves (or their derivatives) from the learning sample, ordered by their distance to $x^{*(a)}(t)$ as measured by the semi-metric $d_l(\cdot,\cdot)$, so that one can define a neighborhood of the k nearest neighbors of $x^{*(a)}(t)$ by

$$\mathcal{N}_l^k\big(x^{*(a)}(t)\big) = \big\{x_j^{(a)}(t) :\ d_l(x^{*(a)}(t), x_j^{(a)}(t)) \leq d_l(x^{*(a)}(t), x_{(k)}^{(a)}(t))\big\}.$$

Let $1(\cdot)$ denote the indicator function. Then a single ensemble member is given by the estimated posterior probability w_{gl} that $x^{*(a)}(t)$ is from class g,

$$w_{gl} = \frac{1}{k} \sum_{\left\{j:\ x_j^{(a)}(t) \in \mathcal{N}_l^k\left(x^{*(a)}(t)\right)\right\}} 1(y_j = g).$$

It is determined by the number of nearest neighbors k, the semi-metric $d_l(\cdot,\cdot)$, and the order of the derivative a. For a specific observation $(y_i, x_i^{(a)}(t))$ from the learning sample, the posterior probabilities are obtained by a leave-one-out procedure, which is formally given by

$$w_{igl} = \frac{1}{k} \sum_{\left\{j\neq i:\ x_j^{(a)}(t) \in \mathcal{N}_l^k\left(x_i^{(a)}(t)\right)\right\}} 1(y_j = g), \qquad i = 1, \dots, n.$$

Simple Functional Nearest Neighbor Ensembles

The observed classes y_i from the previous section are considered as realizations of random variables Y_i that take on values in $\{1, \dots, G\}$. Since the methodology uses p semi-metrics, it comprises p posterior probabilities w_{igl} for each observation i and class g. These single posterior probabilities can be combined in a linear combination to obtain a simple ensemble model for the overall posterior probability that observation $x_i(t)$ is from class g, given by

$$\pi_{ig} = \sum_{l=1}^{p} w_{igl} c_l, \tag{4.4}$$

where c_l are weights that have to be estimated and must satisfy

$$\sum_{l=1}^{p} c_l = 1 \quad \text{and} \quad c_l \geq 0\ \forall l. \tag{4.5}$$

Similar to the k-nearest-neighbor classifier for multivariate data, each observation is assigned to the class of highest posterior probability, i.e. $\hat{y}_i = \max_g (\pi_{ig})$.

In addition to the various semi-metrics used in Ensemble (4.4), we also use different sizes for the number of nearest neighbors $k \in \mathcal{K}_{nN} = \{k_1, \dots k_M\}$ as well as varying orders of derivation $a \in \{a_1, \dots a_O\}$. The definitions of the above neighborhood $\mathcal{N}_l^k \left(x^{*(a)}(t)\right)$ and single posterior probability estimates w_{igl} are adapted accordingly. This means that from now on, the index l does not refer to a single semi-metric, but represents an ensemble member determined by an unique tuple $\{d(\cdot,\cdot), a, k\}$ of a specific semi-metric $d(\cdot,\cdot)$, a number of nearest neighbors k and an order of derivation a.
If multiple functional covariates are observed, it might be necessary to include covariate type-specific semi-metrics into the ensemble. For instance this might arise if the covariate types originate from different domains, as for example time and wavelengths. Another situation is given in our cell chip application in Section 4.4, where two different sensors measure different physiological parameters. With R functional predictor types, one has observations $(y_i, x_{i1}(t), \dots, x_{ir}(t))$, $r = 1, \dots R$, such that each neighborhood, and with that each ensemble member and ensemble coefficient, additionally depends on the covariate type.
Let q, R, M, and O denote the numbers of semi-metrics, covariate types, nearest neighbors, and orders of derivation used. Then, the ensemble comprises a total number of $p = q \cdot R \cdot M \cdot O$ members.

One way to estimate Ensemble (4.4) with respect to Constraint (4.5) is to optimize some loss function like the Brier score, as done in Chapter 3. Such an estimation approach does not allow for category-specific ensemble coefficients, and the extension to ordinal classes is not self-evident. In the next Section, we propose an alternative estimation technique.

4.2.2 The Penalized and Constrained Multinomial Logit Model

Alternatively to loss functions, the estimation of Ensemble (4.4) can be performed via a multinomial logit model, yielding sparse and interpretable results if being penalized and constrained adequately. To illustrate this, for $g = 1, \dots G-1$, let $v_{igl} = (w_{igl} - w_{iGl})$ denote the differences in posterior probability between classes. For a more compact notation, let $\boldsymbol{v}_{ig} = (v_{ig1}, \dots, v_{igp})^T$ and $\boldsymbol{v}_i = (\boldsymbol{v}_{i1}^T, \dots, \boldsymbol{v}_{i,G-1}^T)^T$. We consider, for $g = 1, \dots G-1$, the

following constrained multinomial logit model:

$$\begin{aligned} P(Y_i = g|\boldsymbol{v}_i) &= \pi_{ig} \\ &= \frac{\exp(\boldsymbol{v}_{ig}^T \boldsymbol{c})}{1 + \sum_{s=1}^{G-1} \exp(\boldsymbol{v}_{is}^T \boldsymbol{c})} \\ &= \frac{\exp\left(\sum_{l=1}^{p} v_{igl} c_l\right)}{1 + \sum_{s=1}^{G-1} \exp\left(\sum_{l=1}^{p} v_{isl} c_l\right)} \qquad \text{s.t. } c_l \geq 0 \ \forall l. \end{aligned} \tag{4.6}$$

The probability for the "reference class" G is trivially given by

$$\pi_{iG} = 1 - \sum_{s=1}^{G-1} \pi_{is} = \frac{1}{1 + \sum_{s=1}^{G-1} \exp(\boldsymbol{v}_{is}^T \boldsymbol{c})}.$$

The novelty in Model (4.6) relative to an ordinary MLM is the constraint on the model coefficients. The restriction to values equal or above zero allows a proportional interpretation of the coefficients' values with respect to the data background, and thus is a crucial improvement.

Model (4.6) can also be rewritten in terms of log odds with regard to the reference class G as follows:

$$\begin{aligned} \log\left(\frac{P(Y_i = g|\boldsymbol{v}_i)}{P(Y_i = G|\boldsymbol{v}_i)}\right) &= \boldsymbol{v}_{ig}^T \boldsymbol{c} = \sum_{l=1}^{p} v_{igl} c_l \\ &= \sum_{l=1}^{p} (w_{igl} - w_{iGl}) c_l \qquad \text{s.t. } c_l \geq 0 \ \forall l \end{aligned}$$

for $g = 1, \ldots G-1$. If $w_{igl} > w_{iGl}$ with $c_l \neq 0$, then the logit will change in favor of class g over G accordingly, and vice versa for $w_{igl} < w_{iGl}$. Hence, when classes show differences in their posterior probabilities which are based on a particular curve feature l, this information gets automatically translated into a change of the overall class probability. Given the difference building on the posterior probabilities, which in our model have the role of covariates, the nonnegativity constraint that is imposed on the parameters c_l allows to interpret them as weights that reflect the importance of the different curve features for the classification. From a more technical point of view, the w_{igl} are class-specific covariates, and using their difference to a reference class (here G) is necessary to make the MLM identifiable. The use of these differences also arise naturally when the MLM is motivated via latent utility maximization as shown in McFadden (1973).
Since the linear predictors $\boldsymbol{v}_{ig}^T \boldsymbol{c}$ are transformed to class probabilities via the multinomial logit link, feasible estimates $\hat{\pi}_{ig} \in [0, 1]$ are guaranteed for any parameter estimate $\hat{\boldsymbol{c}}$. This

holds for both the observed data as well as in prediction, so that Model (4.6) can easily be extended to more flexible, class-specific weights c_{gl}:

$$\begin{aligned} P(Y_i = g|\boldsymbol{v}_i) &= \frac{\exp(\boldsymbol{v}_{ig}^T \boldsymbol{c}_g)}{1 + \sum_{s=1}^{G-1} \exp(\boldsymbol{v}_{is}^T \boldsymbol{c}_s)} \\ &= \frac{\exp\left(\sum_{l=1}^{p} v_{igl} c_{gl}\right)}{1 + \sum_{s=1}^{G-1} \exp\left(\sum_{l=1}^{p} v_{isl} c_{sl}\right)} \qquad \text{s.t. } c_{gl} \geq 0 \ \forall g, l. \end{aligned} \tag{4.7}$$

It was shown in Gertheiss and Tutz (2009) that $\hat{\pi}_{ig} \in [0, 1]$ cannot be guaranteed in prediction if one uses class-specific weights c_{gl} in the simple linear ensemble approach (4.4). Hence, the modeling option (4.7) is a major advantage of our cMLM approach since it allows a more flexible model for classification tasks in problems with more than two classes. The analysis of the phoneme data in Section 4.5 illustrates this advantage on real data.

Penalization for the Constrained MLM

Depending on the data at hand and the choices of the sets of predictors, semi-metrics, the number of different neighborhood sizes and the number of different orders of covariate derivatives used, the number p of coefficients that has to be estimated can easily reach orders of 10^2 to 10^5. To regularize the estimates and to find an interpretable set of curve features that explains the curves' class memberships, we propose to use penalized estimation with variable selection penalties.
In the following, let $\log(L(\boldsymbol{c}))$ denote the log-likelihood of Models (4.6) or (4.7), let $J(\boldsymbol{c})$ denote a penalty term and let $\lambda \geq 0$ denote a tuning parameter that controls the strength of the penalization. For the model from (4.6) that uses global weights on all curve features, a suitable penalty is given by the Lasso of Tibshirani (1996), yielding

$$\hat{\boldsymbol{c}} = \underset{\boldsymbol{c}}{\operatorname{argmax}} \left(\log(L(\boldsymbol{c})) \ - \ \lambda J(\boldsymbol{c}) \ = \ \log(L(\boldsymbol{c})) \ - \ \lambda \sum_{l=1}^{p} |c_l| \qquad \text{s.t. } c_l \geq 0 \ \forall l \right). \tag{4.8}$$

Due to its mathematical properties, the Lasso penalty $J(\boldsymbol{c}) = \sum_{l=1}^{p} |c_l|$ induces sparse estimates for suitably chosen λ, that is, one obtains $\hat{c}_l = 0$ for many l, yielding selection of curve features.
For the model from (4.7) with class-specific weights, the Lasso estimates are given by

$$\hat{\boldsymbol{c}} = \underset{\boldsymbol{c}}{\operatorname{argmax}} \left(\log(L(\boldsymbol{c})) \ - \ \lambda \sum_{l=1}^{p} \sum_{g=1}^{G-1} |c_{gl}| \qquad \text{s.t. } c_{gl} \geq 0 \ \forall g, l \right). \tag{4.9}$$

In this case, the Lasso approach induces solutions that are sparse on the parameter level, i.e. it is possible to obtain, for example, $\hat{c}_{1l} > 0$ and $\hat{c}_{2l} = 0$. Since the denominator in

(4.7) is influenced by all parameters, the respective probability $P(Y_i = 2|\boldsymbol{v}_i)$ would still be influenced by curve feature l.
To obtain a proper selection of curve features, one can adopt the Categorically Structured Lasso approach of Tutz et al. (2015), which computes estimates according to

$$\hat{\boldsymbol{c}} = \underset{\boldsymbol{c}}{\operatorname{argmax}} \left(\log(L(\boldsymbol{c})) \; - \; \lambda\sqrt{G-1} \sum_{l=1}^{p} \sqrt{\sum_{g=1}^{G-1} c_{gl}^2} \qquad \text{s.t. } c_{gl} \geq 0 \; \forall g, l \right). \tag{4.10}$$

The CATS approach treats all parameters that belong to the same curve feature l as one parameter group that is removed from the model jointly. However, here the CATS approach can produce solutions with so-called 'within-group-sparsity', which is in stark contrast to the behavior of CATS in the context of Tutz et al. (2015). This phenomenom has technical reasons, see also Equation (4.11) in the following section.

Among other sparsity inducing penalties, the non-zero estimates being obtained when using the above penalties tend to be biased towards zero for specific data situations. Thus, we apply the ordinary penalty versions (4.8) - (4.10) as well as their adaptive counterparts. The adaptive penalties are obtained from the above penalties by an extension adding data-driven weights, see also Zou (2006), Wang and Leng (2008) and Tutz et al. (2015).

4.2.3 Computation of Estimates

From a technical point of view, our constrained and penalized MLM is simply an ordinary penalized MLM with class-specific covariates and a constraint. Therefore, our model is covered by the framework of Tutz et al. (2015) except for the nonnegativity constraint on the parameters. Hence, the FISTA algorithm for the estimation of a penalized MLM proposed in Tutz et al. (2015) must be adapted to incorporate this constraint. The core problem to solve is the so-called proximal operator with nonnegativity constraint. For the Lasso penalty on global parameters as in (4.8), and with $\boldsymbol{u} \in \mathbb{R}^p$ denoting an arbitrary and generic input vector, this problem has the following form:

$$\boldsymbol{Prox}_{\text{lasso}}(\boldsymbol{u}|\lambda) \; = \; \underset{\boldsymbol{c} \in \mathbb{R}^p_{\geq 0}}{\operatorname{argmin}} \left(\tfrac{1}{2}||\boldsymbol{c} - \boldsymbol{u}||_2^2 + \lambda \sum_{l=1}^{p} |c_l| \right).$$

As proven in Jenatton et al. (2011), this problem is solved by simply replacing all negative entries of the input vector with zero, which reduces the problem to the well-studied unconstrained proximal operator. With $[\boldsymbol{u}]_+ = \max(\boldsymbol{u}, 0)$ (to be understood entry-wise),

one obtains

$$\begin{aligned}\underset{\boldsymbol{c}\in\mathbb{R}^p_{\geq 0}}{\operatorname{argmin}} \left(\tfrac{1}{2}||\boldsymbol{c}-\boldsymbol{u}||_2^2 + \lambda \sum_{l=1}^{p} |c_l| \right) &= \underset{\boldsymbol{c}\in\mathbb{R}^p}{\operatorname{argmin}} \left(\tfrac{1}{2}||\boldsymbol{c}-[\boldsymbol{u}]_+||_2^2 + \lambda \sum_{l=1}^{p} |c_l| \right) \\ &= \left(\Big[[u_l]_+ - \lambda \Big]_+ \right)_{l=1,\dots p} = \Big(\max\Big(\max(u_l, 0) - \lambda,\ 0 \Big) \Big)_{l=1,\dots p}.\end{aligned}$$

An equivalent statement holds for the Lasso penalty on class-specific weights from (4.9). For the CATS penalty from (4.10), one obtains for $l = 1 \dots p$

$$\underset{\boldsymbol{c}_l\in\mathbb{R}^{G-1}_{\geq 0}}{\operatorname{argmin}} \left(\tfrac{1}{2}||\boldsymbol{c}_l-\boldsymbol{u}_l||_2^2 + \lambda\sqrt{G-1} \sqrt{\sum_{g=1}^{G-1} c_{gl}^2} \right) = \left[1 - \frac{\lambda\sqrt{G-1}}{\big|\big|[\boldsymbol{u}_l]_+\big|\big|_2} \right]_+ [\boldsymbol{u}_l]_+. \tag{4.11}$$

The corresponding estimation algorithm is implemented in the publicly available `R`-package `MRSP` (Pößnecker, 2015). The nonnegativity constraint on the parameters can be activated by using the argument "`nonneg = TRUE`".

4.2.4 Competing Methods and Prediction Performance Measures

To be able to evaluate the prediction performance of the kNNE estimated via penalized cMLM, we compare all results to alternative approaches. Since only few functional classification methods introduced so far come with an implementation, we will also use some multivariate discrimination techniques that are known for their good performance. To this end, we either use the discretized covariates if appropriate, or otherwise compute a functional principal component (FPC) analysis and use the respective scores as inputs. To the best of our knowledge, the choice of the number of components in FPC analysis is still an open problem. An approach used by some researchers (e.g. Hall et al., 2001) is to use those scores that explain a specified percentage of the sample variability. We do the same, choosing the number of scores such that at least 95% variability is explained. For more details, see Chapter 3, Section 3.2.2, and Appendix B.1. The FPC computation is carried out with the `fpca.sc`-function of the `R`-package `refund` (Di et al., 2009; Crainiceanu et al., 2013; Goldsmith et al., 2013).
In the following, all methods included in the comparison are shortly presented. For more details on the single methods, see Chapter 3, Section 3.3.1, and respective literature. The calculations for all methods, including the penalized cMLM, are carried out using the software environment `R` (R Core Team, 2017) and respective add-on packages. Details on the latter as well as on the choices of parameters, where appropriate, are given in the following. If not stated otherwise, the default parameters are used.

Penalized constrained multinomial logit model (*cMLM*) The method introduced in Sections 4.2.1 - 4.2.3. Before modeling, both the learning as well as the test data set

are normalized to a standard deviation of one. The penalty parameter λ is chosen from a predefined grid of values. For the other penalty parameters, default settings are used. The weights in the adaptive penalty versions base on respective maximum likelihood estimates. The model choice bases on the minimization of the mean Akaike information criterion (AIC). The abbreviation *cMLM* is used whenever a global Lasso penalty is employed. The abbreviation *cs cMLM* implies that a category-specific Lasso penalty is used, *csCATS cMLM* denotes the usage of a categorically structured Lasso penalty. For computation, the add-on package `MRSP` (Pößnecker, 2015) is used.

***k*-nearest-neighbor ensemble** (*kNN Ensemble*, Fuchs et al., 2015a) The ensemble is, apart from the standard deviation's normalization, identical to the one set up for the cMLM, i.e. Model (4.4) has to satisfy Constraints (4.5). The ensemble coefficients are estimated via the Brier score minimization used in the original research article. Software from the respective supplement is used.

Nonparametric functional classification (*NPFC*, Ferraty and Vieu, 2003) We use four semi-metrics the approach offers, namely the Euclidian distance of the ath derivative of the functional covariates (abbreviation *NPFC-deriv*), the Euclidian distance of the Fourier expansions of the functional covariates (abbreviation *NPFC-Fourier*), the multivariate partial least squares regression semi-metric (abbreviation *NPFC-mplsr*), and the functional principal component semi-metric (abbreviation *NPFC-pca*). For details on the single semi-metrics, we refer to Ferraty and Vieu (2006). All semi-metrics require the choice of at least one parameter. These are chosen by minimizing the mean misclassification error of a 10-fold CV. The modeling software can be found at *http://www.math.univ-toulouse.fr/ ferraty/SOFTWARES/NPFDA/index.html*.

Functional linear model (*FLM-log*, Ramsay and Silverman, 2005) In the case of a two-class problem, we use a parametric functional model, which is implemented in the `gam`-function of the `R` package `mgcv` (Wood, 2014). The number of basis functions used for each smooth term has to be chosen. We test a grid $\{3, \dots 10\}$ and choose the number with minimal mean misclassification rate in the test data.

Functional random forests (*fRF*, Möller et al., 2016) The software has kindly been provided by the authors of Möller et al. (2016). The classification is carried out by the function `FuncRandomForest`. Differing from the default call, the variables `importance` and `overlap` are set to `TRUE` to have access to the variable importance measure of the method and achieve a more flexible interval choice. The model parameters λ and c were each chosen from predefined grids via minimizing the mean misclassification rate of a 5-fold CV.

Linear discriminant analysis (*LDA*, Fisher, 1936; Rao, 1973) We apply *LDA* to the FPC scores, as done by Ramsay and Silverman (2002). *LDA* is implemented in the `R` package `MASS` (Ripley et al., 2014), function `lda`.

Penalized discriminant analysis (*PDA-cov.,* Hastie et al., 1995) *PDA* was especially designed for high-dimensional and highly correlated covariates (Hastie et al., 1995), such that it can be applied to the discretized data. The approach is implemented in the `R` package `mda` (Hastie et al., 2015b), function `fda`.

Multinomial model (*mM*, see e.g. Tutz, 2012) A multinomial logistic regression model is used on the FPC scores. This method is implemented in the `maxent`-function of the `R` package `maxent` (Jurka and Tsuruoka, 2013).

Support vector machines (*SVM*, Vapnik, 1996) We use the implementation of the `R` package `e1071` (Meyer et al., 2014) by using the function `svm`, with `probability=TRUE` and default settings else. The *SVM* is applied to both, the discretized data, referred to by *SVM-cov.*, and the FPC scores (*SVM-FPCs*).

Random forests (*RF*, Breiman, 2001) The method is implemented in the `R` package `randomForest` (Breiman et al., 2012). We used the `randomForest`-function. *RF* are applied to both, the discretized data (*RF-cov.*) and the FPC scores (*RF-FPCs*).

Regularized discriminant analysis (*RDA*, Guo et al., 2007) *RDA* penalizes a *LDA* and was designed for high-dimensional data, thus being applied to the discretized covariates. It is available from the `R` package `rda` (Guo et al., 2012), function `rda`.

Sparse discriminant analysis (*SDA*, Clemmensen et al., 2011) Another modification of *LDA* is *SDA*, which we apply to the discretized covariates. An implementation can be found in the `R` package `sda` (Ahdesmaki et al., 2015), function `sda`.

To be able to compare the prediction performances of the competing methods described above, we will use two performance measures. The first is the normalized Brier score operating on the coded response $z_{ig} = 1$ if $y_i = g$ and $z_{ig} = 0$ otherwise, and the posterior probabilities per class,

$$Q \quad = \quad \frac{1}{n_{test}} \frac{1}{G} \sum_{i=1}^{n_{test}} \sum_{g=1}^{G} \left(z_{ig} - \pi_{ig} \right)^2 ,$$

introduced by Brier (1950) (cf. Chapter 3, Section 3.2.3). n_{test} denotes the sample size of the test data. The second performance measure we use is the classical misclassification rate (MCR)

$$MCR = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} 1 \left(y_i = \hat{y}_i \right) .$$

Here, $1(\cdot)$ again denotes the indicator function, y_i denotes the true class of observation $x_i(t)$, and $\hat{y}_i$ the class assigned by the method considered.

In addition to these performance measures we introduce a score that indicates how important the coefficients, estimated by the penalized cMLM, are relative to each other, i.e. which of the features corresponding to the coefficients yields most discriminative power. The score is called the relative feature importance (RFI) measure and yields the percental importance per estimated coefficient $\hat{c}_l$. It is defined by

$$RFI_l = 100 \cdot \frac{\sum_{g=1}^{G} \hat{c}_{gl}}{\sum_{g=1}^{G} \sum_{s=1}^{p} \hat{c}_{gs}}.$$

In the case of global coefficients, the RFI measure simplifies to $RFI_l = 100 \cdot \frac{\hat{c}_l}{\sum_{s=1}^{p} \hat{c}_s}$.

4.3 Simulation Study

The penalized cMLM approach introduced above constitutes one possibility to estimate the coefficients of the k-nearest-neighbor ensemble. Another approach is the Brier score minimization introduced in Chapter 3. To compare the selection results as well as the prediction performance of these two approaches for a known setup, the simulation study from Chapter 3, Section 3.3.2 is re-estimated.

4.3.1 Simulation Study Setup

The data sets generated and examined in Chapter 3 are used again in this section to reduce computational costs. This implies an identical semi-metric set as well as respective parameter choices. A short summary of the setup, i.e. of the underlying generating processes (GP) of the data, is given in the following.

Let $U(\tau_1, \tau_2)$ denote an uniform distribution with limits $[\tau_1, \tau_2]$, $N(\mu, \sigma^2)$ a normal distribution with mean μ and variance σ^2, and $f(t; \mu, \sigma^2)$ a normal density function with mean μ and variance σ^2.
The first GP represents a two-class discrimination task that bases on the gas measurements' mean $x_{g\bar{a}s}(t)$ (see Chapter 1 and Appendix D). The ith functional covariate is built by the sum of the gas measurements' mean and a sum of varying sine functions,

$$x_i(t) = x_{g\bar{a}s}(t) + \alpha_i \max\left(x_{g\bar{a}s}(t)\right) \sum_{m=1}^{L_i} \sin(\gamma_m t).$$

Here, the parameters are $\alpha_i \sim U(-1, 1)$, $L_i = \llbracket \beta_i \rrbracket$, with $\beta_i \sim U(1, 7)$, and $\gamma_m \sim U(-2.5, 2.5)$, with parameters α_i, β_i and γ_m being drawn from uniform distributions on the respective intervals. The ith class is defined with regard to the curves' mean,

$$y_i = \begin{cases} 1 & <=> \quad \int_{\mathbb{D}} x_i(t)dt < \int_{\mathbb{D}} x_{g\bar{a}s}(t)dt \\ 2 & <=> \quad \int_{\mathbb{D}} x_i(t)dt \geq \int_{\mathbb{D}} x_{g\bar{a}s}(t)dt. \end{cases}$$

The second GP simulates a multi-class classification problem. The functional covariates are constituted by

$$x_i(t) = \sum_{m=1}^{L_i} f_m(t; \mu_m, \sigma_m^2),$$

i.e. a sum of L_i normal densities $f_m(t; \mu_m, \sigma_m^2)$, with means $\mu_m \sim U(-1, 3)$, variances $\sigma_m^2 = |\nu_m|$, $\nu_m \sim N(0, 1)$, and L_i being chosen at random from $\{1, \dots 11\}$. The classes y_i are defined with respect to the position of the maximum of the curves. To this end, we divide the domain of definition in five equal sized parts and assign class $y_i = g$ if the maximum of curve $x_i(t)$, $\max(x_i(t)) = x_i(t)_{|t=t_{\max(x_i(t))}}$, lies in the gth part of the domain,

$$y_i = g \qquad \text{if} \qquad \left(t_{(gV-V)/5} < t_{\max(x_i(t))} <= t_{gV/5}\right),$$

with $g \in \{1, 2, 3, 4, 5\}$ and V being the number of observation points, i.e. $t \in \{t_1, \dots t_V\}$. Examples of covariates generated by these two generating processes can be found in Chapter 3, Figure 3.2.
The number of observation points for the discretized covariates $\boldsymbol{x}_i$, $i = 1, \dots n$, is $V = 100$ for both GP, with $t_v \in \mathbb{D} = [0.1, 1]$, $v = 1, \dots V$ equidistant points. The number of observations n is one out of the set $\{100, 300, 1000\}$. The data generation was repeated $W = 100$ times, such that there are 100 estimated ensemble coefficient sets per n and estimation approach. For each, the prediction performance is evaluated on a seperately generated test sample of size $n_{test} = 1000$.
In contrast to the preceding Chapter 3, the standard deviation $\text{sd}(v_{igl})$ is calculated. If $\text{sd}(v_{igl}) \equiv 0 \; \forall i$ for a certain tuple l, the respective tuple is removed from the data set prior to estimation, since it does not contain any information concerning the class. For the data sets used, 8 tuples of the first GP show no variation across classes and are excluded from further analysis. For the second generating process, GP 2, no tuple had to be removed.

4.3.2 Simulation Study Results

Two aspects of the estimation approaches can be compared, namely the prediction performance and the selection results together with their adequacy concerning interpretability.

Figure 4.2 shows the selection results of the penalized cMLM approach for the two-class generating process as boxplots across the 100 replications. For clarity, only those coefficient IDs which were estimated to have mean RFI values higher than 0.1 are shown. The white boxes show respective results for the ordinary, the gray boxes for the adaptive Lasso penalty. Here, the training data set size is $n = 300$. For $n = 100$/ $n = 1000$, the selected coefficient IDs are similar, although less/ more pronounced.
Figure 4.3 shows the same for the multi-class generating process. For the penalties allowing class-specific coefficients, color coding is with respect to the class. The results basing on the ordinary penalty versions are given by the wider boxes, those of the adaptive penalty

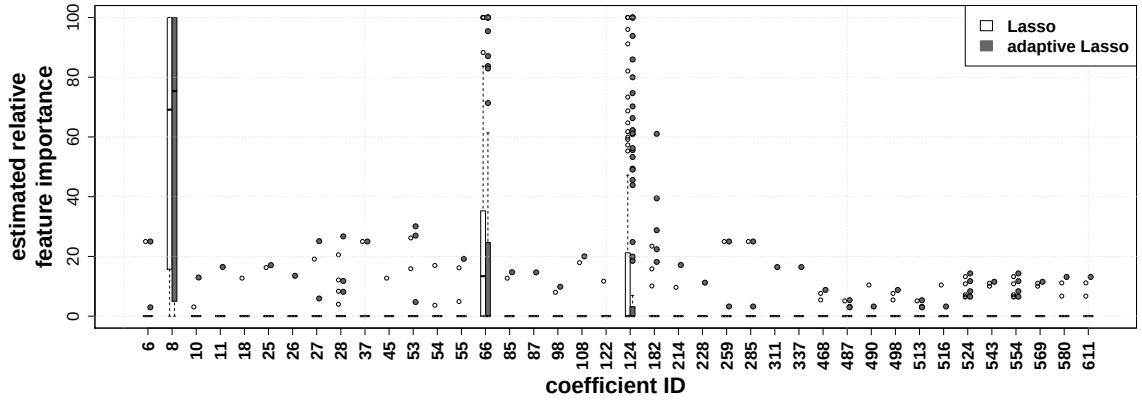


Figure 4.2: Selection results of the penalized cMLM approach for the simulated two-class data. The boxplots show the estimated RFI values across the 100 generated data sets with training data set size $n = 300$. For clarity, only coefficients with an estimated mean RFI value ≥ 0.1 are shown.

	IDs of estimated coefficients (adaptive results)			coefficient ID	parameter tuple
	n=100	n=300	n=1000		
GP 1	*8*	*8*	*8*	6	$\{d^{shortEucl}, a = 0, k = 1\}$, $\mathbb{D}_{small} = [t_{77}, t_{100}]$
	66	*66*	*66*		
	124	*124*	*124*	8	$\{d^{Mean}, a = 0, k = 1\}$
	6	*6*	*28*	28	$\{d^{Scan}, a = 0, k = 1\}$, $\tau = 0.78$
	8 (8)	8 (8)	8 (8)	66	$\{d^{Mean}, a = 0, k = 5\}$
	66 (66)	66 (66)	66 (66)	124	$\{d^{Mean}, a = 0, k = 11\}$
	124 (124)	124 (124)	124 (124)	182	$\{d^{Mean}, a = 0, k = 21\}$
	6 (6)	182 (182)	182 (182)		

Table 4.2: Selection results for the two-class generating process and both estimation approaches. Italic letters refer to the results from the Brier score minimization, ordinary letters to the results from the penalized cMLM approach. Left three columns: IDs of the four estimated coefficients that show the largest means (across replications, in decreasing order, for differing numbers of observations n). On the right, the chosen ensemble coefficients are decoded; the value of a indicates the order of derivation, k indicates the number of nearest neighbors used.

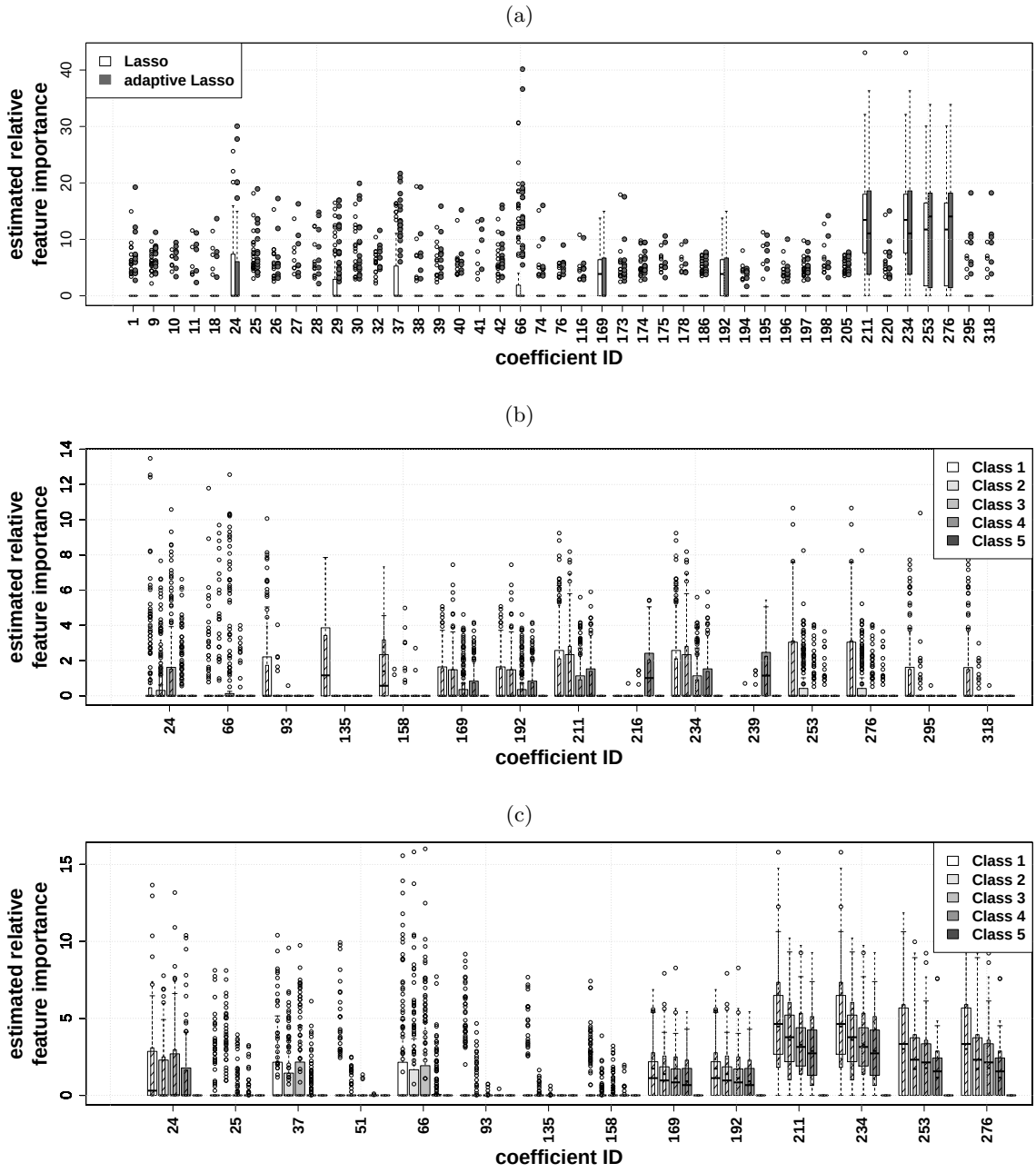


Figure 4.3: Selection results of the penalized cMLM approach for the generated multi-class data. The boxplots show the estimated RFI values across the 100 generated data sets with training data set size $n = 300$. For clarity, only coefficients with (a) an estimated RFI mean value ≥ 0.35 for the global Lasso penalties, (b) an estimated mean RFI value ≥ 0.95 for the cs-Lasso penalties, and (c) an estimated mean RFI value ≥ 0.65 for the csCATS-Lasso penalties, across classes and replications, are shown. For (b) the cs-Lasso penalties and (c) the csCATS-Lasso penalties, color coding refers to the class. Results of the ordinary penalty versions are given by the wider boxes, those of the adaptive penalty versions by the superimposed, narrow shaded boxes.

versions are given by the superimposed, narrow shaded boxes. Only those IDs which were estimated to have mean RFI values higher than 0.35 (Lasso)/ 0.95 (cs-Lasso)/ 0.65 (csCATS-Lasso) are shown. For each penalty, or penalty version, a smaller/ higher number of training data set observations again yields similar selection results.
For both generating processes, more coefficient IDs that were selected, i.e. that show estimated mean RFI values smaller than the above thresholds, can be found in the figures of the Appendices C.1 and C.2, respectively.

In Tables 4.2 and 4.3, one can find the decoding of the coefficient IDs that yield the four highest estimated mean RFI values per generating process and penalty (where appropriate). The ordinary letters refer to the results from the penalized cMLM approach, the italic letters to those from the Brier score minimization.
Concerning the two-class discrimination problem, the selection results of the two estimation approaches are identical for the three coefficient IDs with highest estimated mean RFI values. Only at the fourth position, first differences occur, with the Brier score minimization approach selecting IDs 6 and 28, and the penalized cMLM choosing IDs 6 and 182. Nonetheless, this discrepancy is negligible, since the estimated mean RFI values of these coefficient IDs are marginally larger than those of the following selected tuples (per estimation approach). For example, the penalized cMLM also chooses ID 28, but at the fifth or sixth position, depending on n. It can be concluded that the estimation approaches give very consistent selection results in this two-class task. Recall that, for this GP, the class assignment bases on the respective curves' means. Thus, the choice of tuples ascribed to semi-metric $d^{Mean}(\cdot, \cdot)$ is sensible.
For the second GP, the choice of coefficient IDs of the penalized cMLM and the Brier score minimization approach are mostly similar for small and medium training sample sizes $n < 300$, although their estimated mean RFI values differ. For higher numbers of training observations, the penalized cMLM tends to select higher coefficient numbers, implying a higher number of nearest neighbors in the respective tuples. However, both estimation approaches give the highest weights to tuples employing the Euclidian distance. The only exception is ID 37, being selected once for the small training data size. As was discussed in Chapter 3, these results are reasonable. Although the classes are assigned with respect to the position of the curves' maxima, the whole curves' Euclidian distances often offer more discriminative power, since the curves exhibit only slight gradients to and from their maxima.

As a second aspect of the estimation approach comparison, the prediction performance is examined. Figure 4.4 gives the Brier scores and misclassification rates as boxplots across the 100 replications, evaluated on the test data set. The upper panels present the results for the two-class, the lower panels those for the multi-class generating process. On the one hand, the prediction results of the penalized cMLM approach are shown. For each penalty used, the left boxes show the results of the ordinary, the right shaded boxes the results of

		IDs of estimated coefficients (adaptive results)			coefficient ID	parameter tuple
		n=100	n=300	n=1000		
GP 2		*24*	*211*	*234*	24	$\{d^{Eucl}$ with $x_i(t)$ centered, $a = 0,\ k = 1\}$
		211	*234*	*211*	37	$\{d^{Points}$ with $x_i(t)$ centered, $a = 0,\ k = 1\}$
		234	*24*	*24*	66	$\{d^{Eucl}$ with $x_i(t)$ centered, $a = 0,\ k = 5\}$
		192	*169*	*66*	169	$\{d^{Eucl}, a = 1,\ k = 1\}$
	Lasso	211 (211)	211 (253)	318 (318)	192	$\{d^{Eucl}$ with $x_i(t)$ centered, $a = 1,\ k = 1\}$
		234 (234)	234 (276)	295 (295)	211	$\{d^{Eucl}, a = 1,\ k = 5\}$
		24 (24)	253 (211)	253 (253)	234	$\{d^{Eucl}$ with $x_i(t)$ centered, $a = 1,\ k = 5\}$
		169 (169)	276 (234)	276 (276)	253	$\{d^{Eucl}, a = 1,\ k = 11\}$
	cs-Lasso	24 (24)	211 (211)	234 (234)	276	$\{d^{Eucl}$ with $x_i(t)$ centered, $a = 1,\ k = 11\}$
		37 (169)	234 (234)	211 (211)	295	$\{d^{Eucl}, a = 1,\ k = 21\}$
		169 (192)	24 (24)	276 (276)	318	$\{d^{Eucl}$ with $x_i(t)$ centered, $a = 1,\ k = 21\}$
		192 (37)	169 (169)	253 (253)		
	csCATS-Lasso	24 (24)	211 (211)	234 (318)		
		211 (169)	234 (234)	253 (295)		
		234 (192)	253 (253)	276 (234)		
		169 (211)	276 (276)	211 (211)		

Table 4.3: Selection results for the multi-class generating process and both estimation approaches. Italic letters refer to the results from the Brier score minimization, ordinary letters to the results from the penalized cMLM approach. Left three columns: IDs of the four estimated coefficients that show the largest means (across replications and classes, in decreasing order, for differing numbers of observations n). On the right, the chosen ensemble coefficients are decoded; the value of a indicates the order of derivation, k indicates the number of nearest neighbors used.

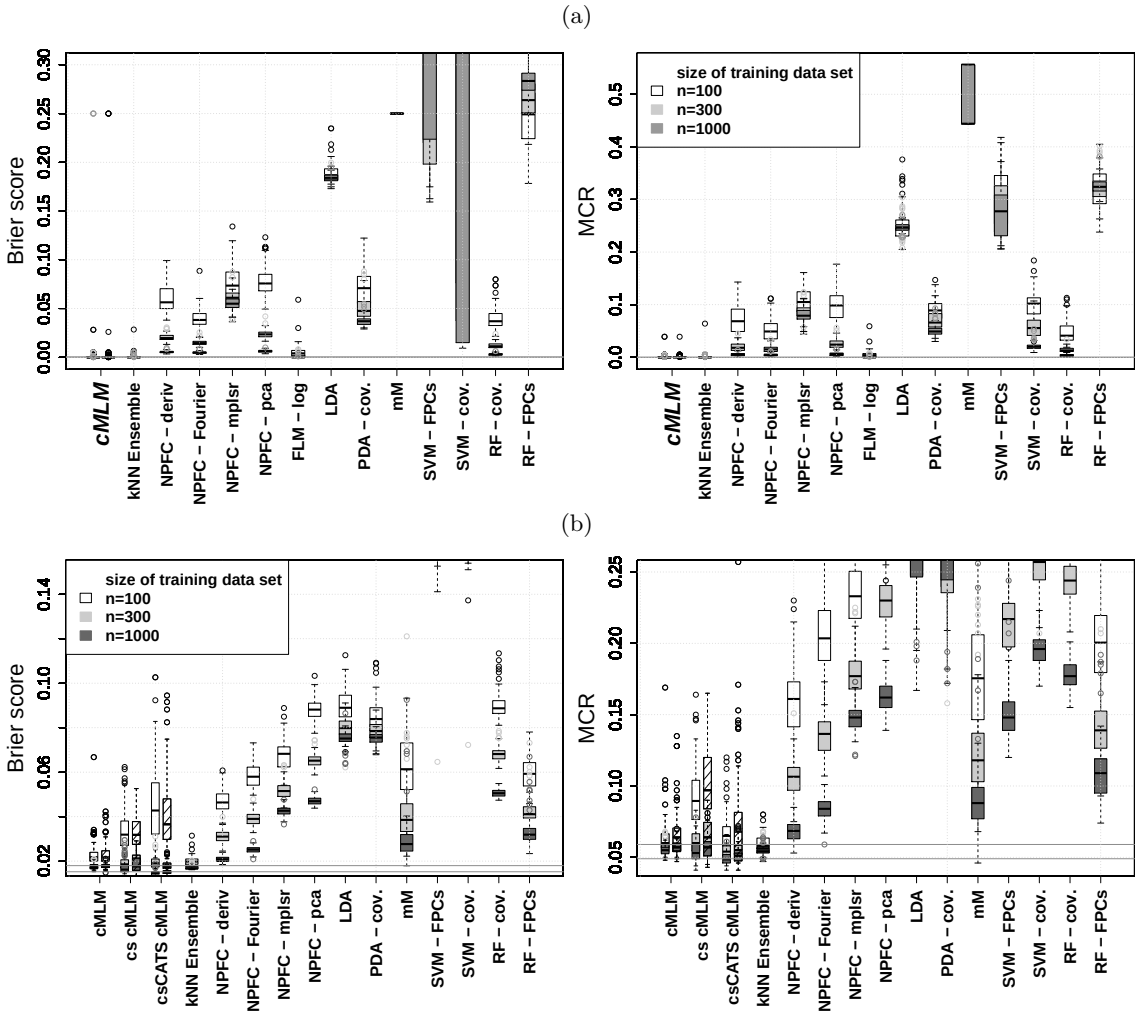


Figure 4.4: (a) Results for $n_{val} = 1000$ test observations for the two-class generating process. The models were estimated 100 times with sample sizes $n = 100$ (white boxes), $n = 300$ (light gray boxes) and $n = 1000$ (dark gray boxes). The left panel shows the Brier scores, the right panel the misclassification rates (MCR). For the penalized cMLM approach, the left boxes show results when using the ordinary, the right, shaded boxes results when using the adaptive penalty version. The horizontal lines indicate the values of the first and third quartiles of the cMLM box with $n = 1000$ and the cs-Lasso penalty. (b) The same for the multi-class generating process.

the adaptive penalty versions. On the other hand, the figure includes the prediction results for all methods that were compared in Chapter 3 (see also Section 4.2.4 for the meaning of the abbreviations). Since we use identical data sets, the results for these classification approaches are not re-estimated and thus are identical to those in Chapter 3, Figure 3.4 and Table 3.4. The only exception here are the results for the two-class discrimination task when estimated by the Brier score minimization (abbreviation *kNN Ensemble*). This is due to the 8 tuples without variation across classes, which were excluded from estimation. Nonetheless, the mean values of the prediction performance measures remain the same as in the previous chapter, with (mean Brier score = 1.61/ mean MCR = $9.4 \cdot 10^{-4}$) for $n = 100$ to (mean Brier score = 0.49/ mean MCR = $1.7 \cdot 10^{-4}$) for $n = 1000$.
Obviously, the penalized cMLM approach and the estimation via minimizing the Brier score outperform all other methods in both classification tasks. Comparing the penalties used in GP 2, the global Lasso penalty yields the smallest Brier score and MCR values for small sample sizes. With increasing training sample sizes, the ordinary csCATS-Lasso penalty shows the best performance. Furthermore, for a small number of training observations $n \leq 100$, the Brier score minimization performs worse than the penalized cMLM for both performance measures in GP 1, and better in GP 2. For higher training data sizes, this relation reverses.

In conclusion, both estimation approaches give similar and interpretable selection results. Concerning the prediction performance, the simulation study indicates that the Brier score minimization gives better results for data with a small sample size. The penalized cMLM approach should be preferred for more complex data including multiple classes.

4.4 Application to Real World Data – Cell Based Sensor Chips

Cell based sensor technologies attract much interest in biomechanical engineering, especially in application fields concerned with environmental quality monitoring, as was discussed in the previous chapters.
In this study we use cell based chips. There are different kinds of sensors distributed across the chip surface, which record different cell reactions. Here, we restrict the data to the five ion sensitive field effect transistors (ISFET) and the interdigitated electrode structure (IDES). For a detection layer, we use chinese hamster lung fibroblast cells. For more details, please see Chapter 1 and Appendix D. To evaluate the performance of our approach in a binary classification problem, the cell chip data is further restricted to measurements with nutrient medium only, and measurements where 2.5mM paracetamol (short: AAP) is added. The data set includes $n = n_0 + n_1 = 120$ measurements per signal type of $V = 89$ equidistant observation points, $n_0 = 63$ without and $n_1 = 57$ with AAP, depicted in Figure 4.5. After the cells have adapted to their environment during the acclimatisation phase,

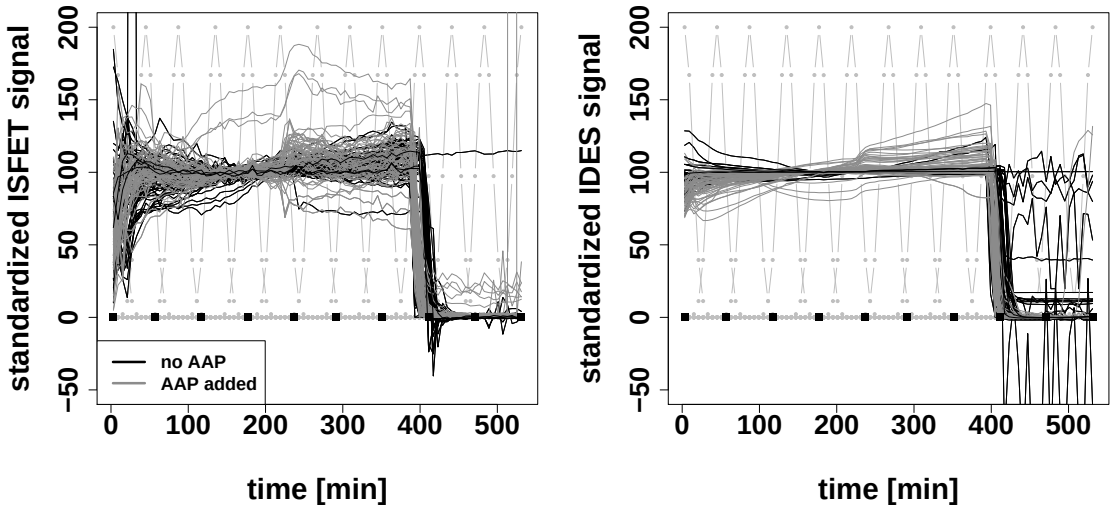


Figure 4.5: The same $n = 120$ standardized ISFET- and IDES-signals as before. The light gray, dotted lines depict the function $\phi_\tau(t)$ in $d_\tau^{Scan}(\cdot,\cdot)$ used at certain observation points τ, the impact points t_e used in $d^{Points}(\cdot,\cdot)$ are depicted as black boxes.

shortly before the test substance is applied, one expects the cells to exhibit 100% viability. At the respective time point, all signals were standardized to a value of 100.
Since the ISFET- and the IDES-signals originate from different biochemical processes and measurement principles, it is adequate to treat them as two functional covariate types, such that the modeling implies multiple covariates. The single curves, however, exhibit the three measurement phases described afore. Thus, identical semi-metrics with the same k-nearest-neighbor parameter tuples can be used for both signals.
We use all the semi-metrics listed in Table 4.1. To allow for comparison of the results from the two estimation approaches, the semi-metric parameters are equal to those in Chapter 3. They inlcude the numbers of nearest neighbors $k \in \mathcal{K}_{nN} = \{1, 5, 11, 21\}$ and orders of derivation $a \in \{0, 1, 2\}$. The choices of $\mathbb{D}_{small}$, $\mathbb{D}_1$, $\mathbb{D}_2$, t_e and τ mainly reflect curve regions where the AAP reaches the cells in phase two and the changeover of phase two and three. For the semi-metric $d^{shortEucl}_{\mathbb{D}_{small}}(\cdot,\cdot)$, one of the intervals $[t_1, t_{35}]$, $[t_{36}, t_{40}]$, $[t_{41}, t_{64}]$, $[t_{65}, t_{69}]$, or $[t_{70}, t_{89}]$ is used for $\mathbb{D}_{small}$; for semi-metric $d^{Jump}_{bo}(\cdot,\cdot)$, one of the sets $\{t_{36}, t_{39}\}$ or $\{t_{65}, t_{68}\}$ is used for $\{t_b, t_o\}$; for semi-metric $d^{relAreas}(\cdot,\cdot)$, $\mathbb{D}_1$ is one of the intervals $[t_1, t_{35}]$ or $[t_{41}, t_{64}]$, and $\mathbb{D}_2 = [t_{41}, t_{64}]$; for semi-metric $d^{Points}(\cdot,\cdot)$, an equidistant grid $t_e = t_{mV/10}$, $m = 1, \ldots 10$, is used; and the function $\phi_\tau(t) = \frac{300}{\max(\phi_{1,\tau}(t))}\phi_{1,\tau}(t)$ with $\phi_{1,\tau}(t) = \frac{1}{\sqrt{2\pi}\sigma}\exp^{-\frac{1}{2}\left(\frac{t-\tau}{\sigma}\right)^2}$, $\sigma = 10$ and $\tau \in \{3.120,\ 45.120,\ 87.120,\ 135.120,\ 177.120,\ 219.129,\ 267.129,\ 309.129,\ 351.129,\ 399.140,\ 441.140,\ 483.140,\ 531.140\}$ is used in semi-

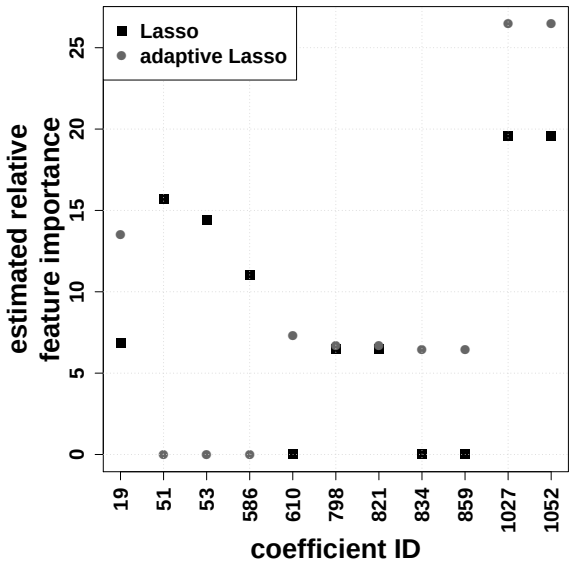


Figure 4.6: Relative feature importance of the coefficients which have been estimated unequal to zero for at least one of the penalties, as estimated from the whole cell chip data.

metric $d_\tau^{Scan}(\cdot,\cdot)$. In Figure 4.5, the weight functions $\phi_\tau(t)$ per τ, used in $d_\tau^{Scan}(\cdot,\cdot)$, are depicted as light gray, dotted lines; the impact points t_e used in $d^{Points}(\cdot,\cdot)$ are marked by black boxes.

4.4.1 Results

With the above parameter choices, the ensemble comprises $p = 1248$ ensemble members, i.e. 624 coefficients per signal type that have to be estimated. The respective single posterior probability estimates w_{igl} per curve $x_i(t)$ are calculated as described in Section 4.2.1. Afterwards, analogously to Section 4.3.1, the standard deviation $\mathrm{sd}(v_{igl})$ is calculated, and all tuples yielding $\mathrm{sd}(v_{igl}) \equiv 0\ \forall i$ are removed from the data set. In that way, 80 tuples are removed.
Finally, the penalized cMLM is applied to the whole cell chip data as described in Section 4.2.2, using a global (adaptive) Lasso penalty and a final number of 1168 coefficients that have to be estimated. The models yielding minimal mean AIC employ penalty parameters of $\lambda_{Lasso}/\ \lambda_{ada.Lasso} = 1.9/\ 0.549$. The RFI of the coefficient estimates resulting from the whole cell chip data set are depicted in Figure 4.6. For clarity, only the 11 coefficients that have been estimated with values unequal to zero for at least one of the penalties used, i.e. the global Lasso (black boxes) or global adaptive Lasso (gray dots), are shown. As can be seen, the estimates differ considerably for both penalties, with only coefficients 798 and 821 exhibiting similar and coefficients 1027 and 1052 showing the highest estimated

coefficient ID (estimated RFI values)	parameter tuple
Lasso	
1027 (19.55)	$\{d^{shortEucl}$, $a = 2$, $k = 5\}$, $\mathbb{D}_{small} = [t_{36}, t_{40}]$, covariate IDES
1052 (19.55)	$\{d^{shortEucl}$, with $x_i(t)$ centered, $a = 2$, $k = 5\}$, $\mathbb{D}_{small} = [t_{36}, t_{40}]$, covariate IDES
51 (15.68)	$\{d^{Eucl}$, $a = 0$, $k = 5\}$, covariate ISFET
53 (14.41)	$\{d^{shortEucl}$, $a = 0$, $k = 5\}$, $\mathbb{D}_{small} = [t_{36}, t_{40}]$, covariate ISFET
586 (11.03)	$\{d^{shortEucl}$, $a = 0$, $k = 1\}$, $\mathbb{D}_{small} = [t_1, t_{35}]$, covariate IDES
adaptive Lasso	
1027 (26.47)	$\{d^{shortEucl}$, $a = 2$, $k = 5\}$, $\mathbb{D}_{small} = [t_{36}, t_{40}]$, covariate IDES
1052 (26.47)	$\{d^{shortEucl}$, with $x_i(t)$ centered, $a = 2$, $k = 5\}$, $\mathbb{D}_{small} = [t_{36}, t_{40}]$, covariate IDES
19 (13.52)	$\{d^{Scan}$, $a = 0$, $k = 1\}$, $\tau = 219.13$, covariate ISFET
610 (7.28)	$\{d^{Scan}$, $a = 0$, $k = 1\}$, $\tau = 531.14$, covariate IDES
798 (6.71)	$\{d^{Scan}$, $a = 1$, $k = 1\}$, $\tau = 3.12$, covariate IDES

Table 4.4: First column: the IDs of the five estimated coefficients that show the largest relative feature importance (RFI) values (in brackets, in decreasing order), estimated from the whole cell chip data. Second column: The chosen ensemble coefficients are decoded, with value a indicating the order of derivation and k indicating the number of nearest neighbors used.

RFI values. The tuples corresponding to the five coefficients estimated with the highest RFI values are decoded in Table 4.4.

To evaluate the prediction performance of our method and to be able to compare it to the other classification methods, we divide the data set randomly 100 times into learning sets comprising 90 curves and test sets of size 30. All competing methods are applied to identical sample sets. Respective abbreviations are given in Table 4.5, or else are identical to those in Chapter 3, Table 3.2. Please note that, to ensure comparability, the 80 tuples yielding $\text{sd}(v_{igl}) \equiv 0 \ \forall i$ are also removed prior to estimation of the ensemble coefficients via Brier score minimization.

As with the whole data set, the estimation results of our approach per draw are sparse. Overall, only 84 (Lasso)/ 94 (adaptive Lasso) coefficient IDs across all replications were estimated to have values above zero, and most were selected very seldom, see also the figure in Appendix C.3. The coefficient IDs with the five highest estimated mean RFI values are 1027, 1052, 586, 597, and 1043 for the Lasso penalty (in decreasing mean value order), and 1027, 1052, 1043, 1066, and 586 for the adaptive Lasso penalty, partly overlapping with the

Method	Abbreviation	R function used (package name)
Constrained multinomial logit model (global Lasso penalty)	cMLM	see Pößnecker (2015) and the online supplement in Fuchs et al. (2016)
Constrained multinomial logit model (category-specific Lasso penalty)	cs cMLM	see above
Constrained multinomial logit model (category-specific CATS penalty)	csCATS cMLM	see above
Functional random forests	fRF	FuncRandomForest (provided by the authors of Möller et al., 2016)
Regularized discriminant analysis	RDA	rda (rda)
Shrinkage discriminant analysis	SDA	sda (sda)

Table 4.5: List of methods, additional to those in Chpater 3, included in the comparison.

coefficients that have been estimated from the whole data listed in Table 4.4. The tuples corresponding to the unlisted IDs are 597 $\hat{=}$ $\{d^{Points}, a = 0, k = 1\}$ with covariate IDES, 1043 $\hat{=}$ $\{d^{Scan}, a = 2, k = 5\}$ with $\tau = 219.13$ and covariate IDES, and 1066 $\hat{=}$ $\{d^{Scan}, a = 2, k = 5\}$ with centered covariates, $\tau = 219.13$ and covariate IDES. From the selected coefficients, one can conclude that both signal types, ISFET- as well as IDES-signals, imply discriminative information. Thus, the classification task is fulfilled without variable selection. Concerning feature selection, most of the selected coefficient IDs, i.e. parameter tuples, include the curve region around 220 minutes. This is reasonable since, at this time, the AAP reaches the cells. Coefficients 586 and 798, representing tuples including the first measurement phase of the IDES-signals, seems also a sensible choice, since many of the IDES curves show a class-dependent slope. Solely coefficient ID 610 putting weight on the very last part of the IDES-signals is not in accordance with the biological background. Remember that the cells are devitalized in the last measurement phase, such that the signals carry information about the chip state, but not about the classification task anymore. However, this coefficients' estimated RFI is small compared to that of the other tuples. In conclusion, the feature selection of our penalized cMLM agrees very well with background knowledge of the cell chip data.
Since, in the present chapter, the CLARK-signals were excluded from the evaluation, selection results for the kNNE are different in this chapter and Chapter 3 (cf. Table 3.6). The coefficient IDs yielding the five highest mean values across splits, selected from the reduced cell chip data by the loss optimization estimation approach, are 586, 597, 1052,

1027, and 1 (in decreasing mean value order), with $1 \hat{=} \{d^{Eucl}, a = 0, k = 1\}$ with covariate ISFET. Thus, except for the last ID, the selection results are the same for both estimation approaches. The estimated RFI values naturally differ, since the constraints put on the ensemble coefficients differ.

Figure 4.7 gives the test results for all classification approaches on basis of the 100 modeling replications. The upper panel shows boxplots of the Brier score across all draws (for RDA, the estimated probabilities were not accessible), the lower panel shows boxplots of the misclassification rates. The prediction performance results of the two penalties used in the penalized cMLM are comparable, with the Lasso penalty yielding somewhat lower MCR values. Obviously, the prediction performance of the kNNE when estimated via minimization of the Brier score deteriorates when excluding the CLARK-signals and the 80 tuples yielding $\text{sd}(v_{igl}) \equiv 0 \; \forall i$ from the data (compare results presented in Figure 3.9). The penalized cMLM outperforms all other methods in terms of both performance measures. The estimation via Brier score minimization is only competitive if retaining the 80 tuples, with a prediction performance comparable to that of the penalized cMLM in both performance measures (not shown here). Thus, the penalized cMLM is a very attractive choice for this discrimination task.

4.5 Application to Real World Data – Phoneme Data

Another data example that became quite popular in functional data classification is the phoneme data introduced in Hastie et al. (1995), available through the `R`-package `ElemStatLearn` (Hastie et al., 2015a). The data consists of 4509 log-periodograms, taken as covariates $x_i(t)$, that each are ascribed to one of the five phonemes "aa", "ao", "dcl", "iy", and "sh", recorded at 256 frequencies, i.e. observation points. Figure 4.8 shows five exemplarily curves per phoneme. The goal of this discrimination task is to differ between the log-periodograms, a task arising in the field of speech recognition.

The parameter settings for the k-nearest-neighbor ensemble members were chosen arbitrarily due to the absence of relevant background knowledge. As in Chapter 3, Section 3.6, we use $k \in \mathcal{K}_{nN} = \{1, 5, 11, 21\}$ nearest neighbors and orders of derivation $a \in \{0, 1, 2\}$. One of the intervals $[t_1, t_{17}]$, $[t_{18}, t_{36}]$, $[t_{37}, t_{56}]$, $[t_{57}, t_{76}]$, $[t_{77}, t_{100}]$, or $[t_{30}, t_{65}]$ is used for $\mathbb{D}_{small}$ in semi-metric $d^{shortEucl}_{\mathbb{D}_{small}}(\cdot, \cdot)$; for semi-metric $d^{Jump}_{bo}(\cdot, \cdot)$, one of the sets $\{t_{15}, t_{19}\}$, $\{t_{34}, t_{40}\}$, $\{t_{54}, t_{58}\}$, or $\{t_{74}, t_{78}\}$ is used for $\{t_b, t_o\}$; for semi-metric $d^{relAreas}(\cdot, \cdot)$, $\mathbb{D}_1$ is one of the intervals $[t_1, t_{17}]$, $[t_{57}, t_{76}]$, or $[t_{30}, t_{65}]$, and $\mathbb{D}_2 = [t_{37}, t_{56}]$; for semi-metric $d^{Points}(\cdot, \cdot)$, a grid $t_e \in \{1, 14.42, 27.84, 41.26, 54.68, 68.11, 81.53, 94.95, 108.37, 121.79, 135.21, 148.63, 162.05, 175.47, 188.89, 202.32, 215.74, 229.16, 242.58, 256\}$ is used; and the function $\phi_\tau(t) = \frac{300}{\max(\phi_{1,\tau}(t))} \; \phi_{1,\tau}(t)$ with $\phi_{1,\tau}(t) = \frac{1}{\sqrt{2\pi}\sigma} \exp^{-\frac{1}{2}\left(\frac{t-\tau}{\sigma}\right)^2}$, $\sigma = 6$ and $\tau \in \{1, 22, 25, 43, 50, 64, 75, 86, 107, 110, 128, 149, 171, 175, 192, 213, 234, 256\}$ is used for semi-metric $d^{Scan}_{\tau}(\cdot, \cdot)$.

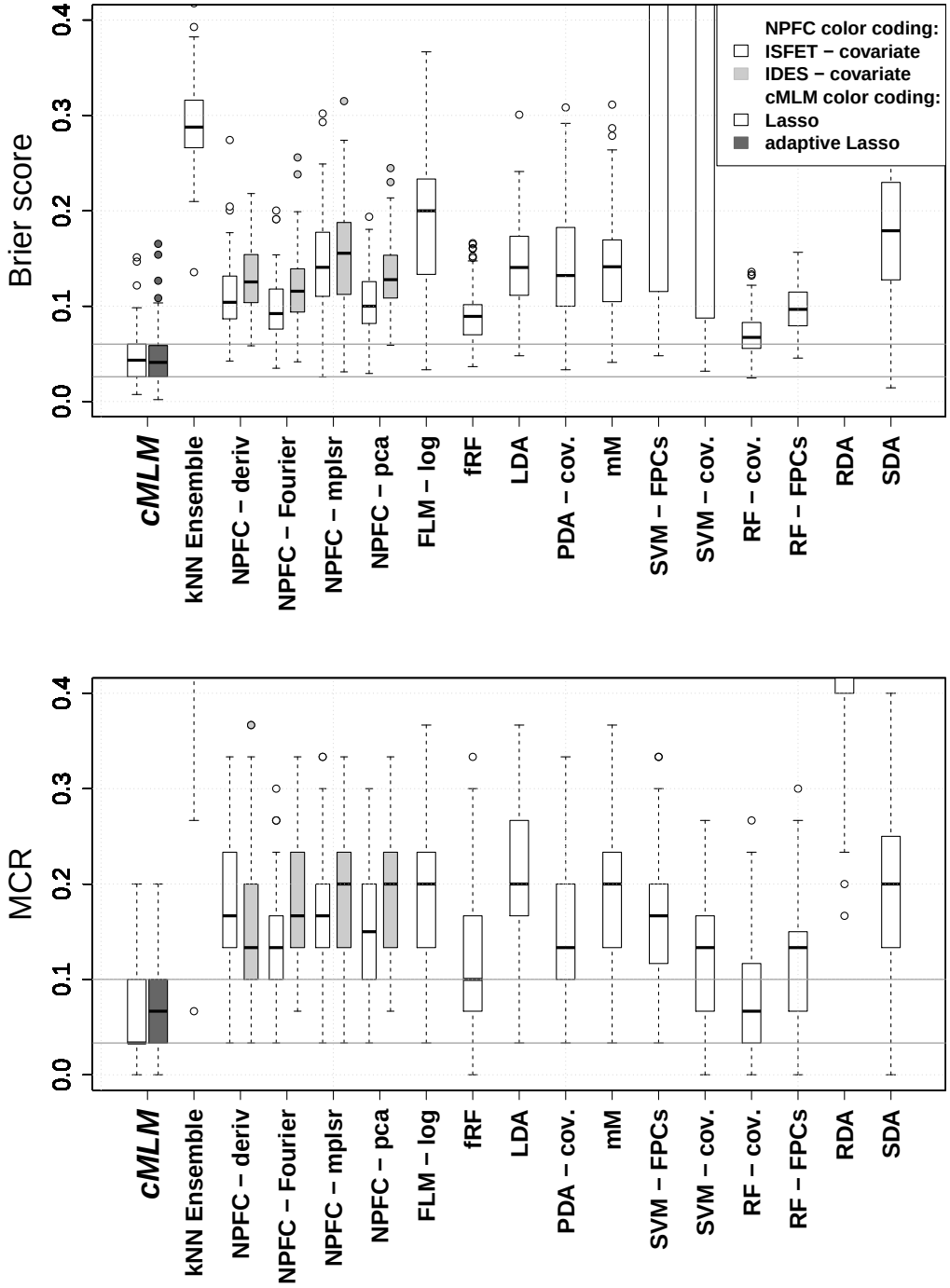


Figure 4.7: Test results of the cell data for all classification approaches on basis of 100 replications. The upper panel shows the Brier scores, the lower panel the misclassification rates (MCR). The horizontal lines indicate the values of the first and third quartiles of the cMLM box with Lasso penalty.

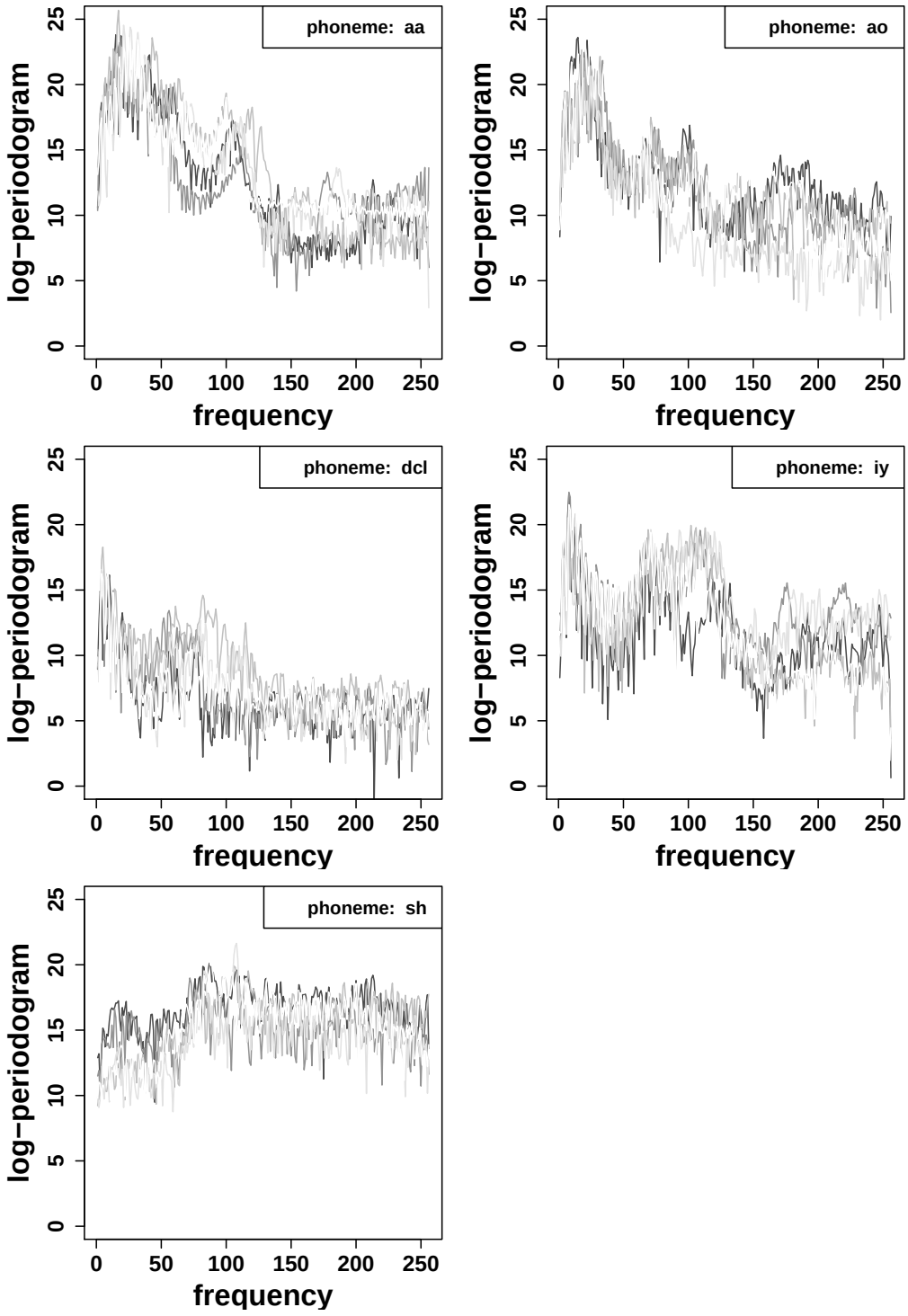


Figure 4.8: Five exemplarily log-periodograms per phoneme.

4.5.1 Results

The upper parameter setting results in 816 ensemble coefficients. Using the whole phoneme data, one finds 16 tuples yielding standard deviations $\text{sd}(v_{igl}) \equiv 0 \; \forall i$. These tuples are removed such that the ensemble contains $p = 800$ coefficients that have to be estimated. Since the data at hand is multi-class data, category-specific penalties might be useful. Thus, we test the performance of the penalized cMLM using the global, the category-specific (cs) and the category-specific CATS (csCATS) penalties introduced in Section 4.2.2, each with and without adaptive weights. The models yielding minimal AIC employ penalty parameters $(\lambda_{Lasso}/\ \lambda_{cs-Lasso}/\ \lambda_{csCATS-Lasso}) = (3.34/\ 6.57/\ 1.7)$, and for the adaptive penalties $(\lambda_{ada.Lasso}/\ \lambda_{ada.cs-Lasso}/\ \lambda_{ada.csCATS-Lasso}) = (1.21/\ 7.78/\ 2.38)$, respectively.

In Figures 4.9 and 4.10, the black symbols present the model coefficients' RFI (per class where appropriate), as estimated from the whole phoneme data, using the respective penalties. The gray symbols show the same for the adaptive penalty versions. For clarity, solely the (Lasso/ cs-Lasso/ csCATS-Lasso) $\hat{=}$ (42/ 101/ 58) coefficients for the ordinary Lasso and the (Lasso/ cs-Lasso/ csCATS-Lasso) $\hat{=}$ (34/ 88/ 58) coefficients for the adaptive Lasso penalties that are estimated to be of values unequal to zero (for at least one category in the case of cs- or csCATS-Lasso) are shown.
For the global penalty, the estimated coefficient RFI values are mostly similar if chosen by both, the ordinary and the adaptive version. Solely the RFI values of coefficients with the IDs 138 and 175 tend to be noticeably higher for the adaptive penalty. For both penalty versions, the coefficients 138, 205, 241, and 252 exhibit the highest estimated RFI values. The corresponding tuples can be found in Table 4.6. For the two category-specific penalties, most estimated RFI values vary strongly between the single classes, and differ from their adaptive counterparts. This indicates that a category-specific penalty is adequate for this data. The decoding of the four coefficients with the highest estimated RFI values can be found in Table 4.6.

To again evaluate the prediction performance of our method and compare it to the other classification methods, 150 curves per class are drawn randomly from the complete data set, being used as a learning data set. The test sample contains another 250 randomly drawn curves per class. The draws, modeling and test steps were repeated 100 times, with all competing methods being applied to identical sample sets. The previously mentioned 16 tuples with $\text{sd}(v_{igl}) \equiv 0 \; \forall i$ are excluded from the ensemble prior to estimation of the kNNE via minimizing the Brier score.
Again, the estimation results of our approach per draw and penalty type are sparse. Across the 100 replications, the penalized cMLM estimates overall 541 (Lasso)/ 726 (cs-Lasso)/ 363 (csCATS-Lasso) coefficient IDs to have values above zero across both respective penalty versions. However, most coefficients are chosen seldomly, see also the examples in the Appendices C.4 - C.6. The coefficient IDs with the four highest estimated mean RFI values are

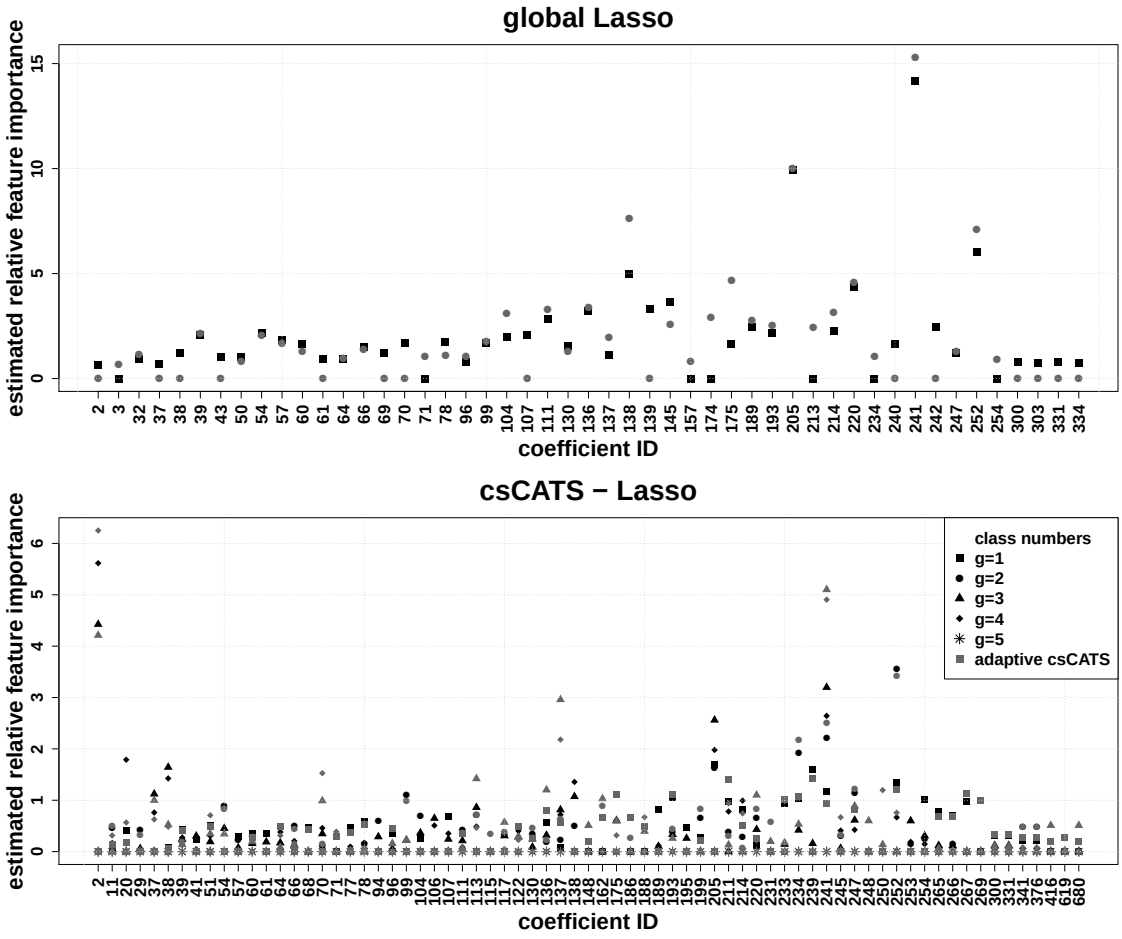


Figure 4.9: Relative feature importance of the coefficients which have been estimated unequal to zero, as estimated from the whole phoneme data. Black symbols denote the results for the ordinary, gray symbols those for the adaptive penalty versions. The upper panel shows the results for the (global) Lasso penalty, the lower panel those for the csCATS penalty.

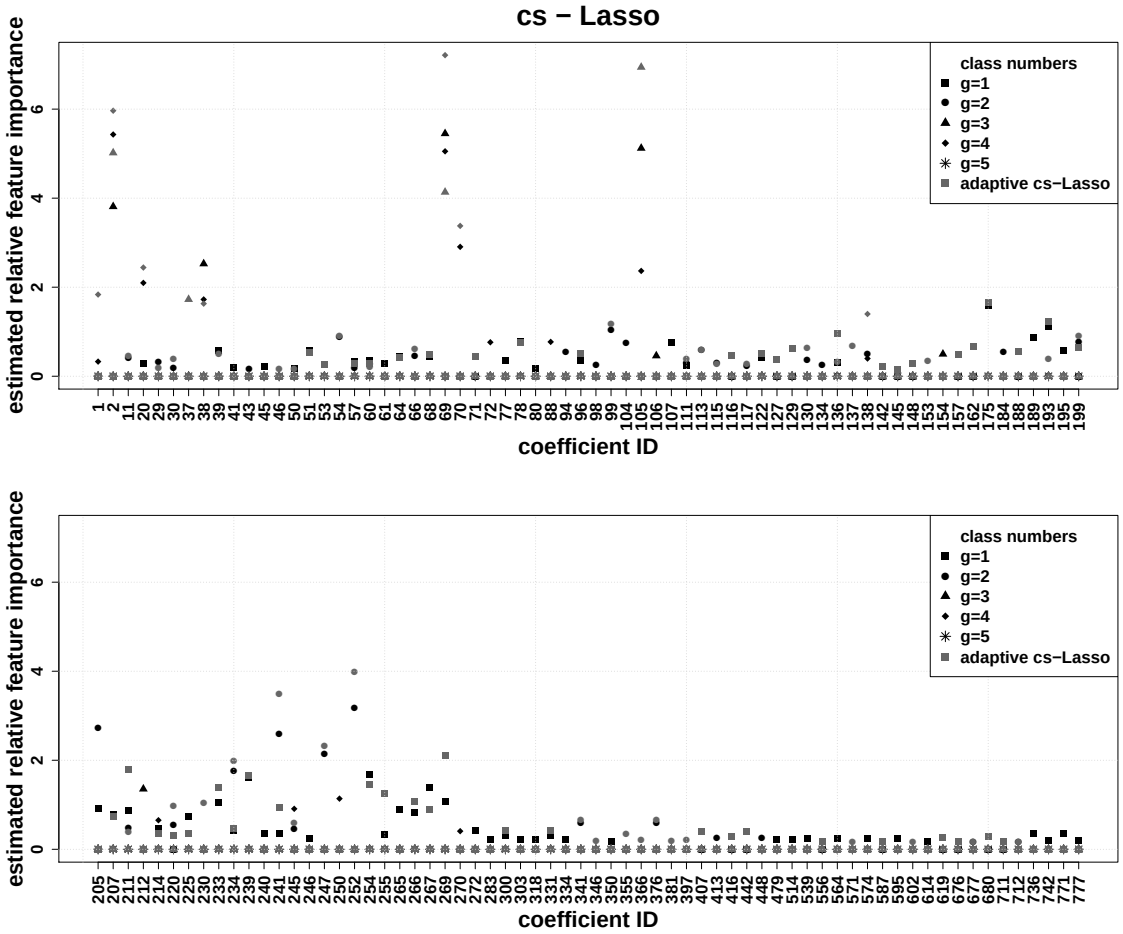


Figure 4.10: Relative feature importance of the coefficients which have been estimated unequal to zero, as estimated from the whole phoneme data, using the cs-Lasso penalty. Black symbols denote the results for the ordinary, gray symbols those for the adaptive penalty version.

coefficient ID (estimated RFI values)	parameter tuple
Lasso / adaptive Lasso	
241 (14.16 / 15.29)	$\{d^{Eucl}$ with $x_i(t)$ centered, $a = 0$, $k = 21\}$
205 (9.92 / 10.01)	$\{d^{Eucl}$, $a = 0$, $k = 21\}$
252 (6.04 / 7.08)	$\{d^{Max}$ with $x_i(t)$ centered, $a = 0$, $k = 21\}$
138 (4.97 / 7.64)	$\{d^{shortEucl}$, $a = 0$, $k = 11\}$, $\mathbb{D}_{small} = [t_1, t_{17}]$

coefficient ID (estimated RFI values)	parameter tuple
cs-Lasso	
69 (10.5)	$\{d^{Eucl}$, $a = 0$, $k = 5\}$
2 (9.24)	$\{d^{shortEucl}$, $a = 0$, $k = 1\}$, $\mathbb{D}_{small} = [t_1, t_{17}]$
105 (7.49)	$\{d^{Eucl}$ with $x_i(t)$ centered, $a = 0$, $k = 5\}$
38 (4.25)	$\{d^{shortEucl}$ with $x_i(t)$ centered, $a = 0$, $k = 1\}$, $\mathbb{D}_{small} = [t_1, t_{17}]$
adaptive cs-Lasso	
69 (11.35)	$\{d^{Eucl}$, $a = 0$, $k = 5\}$
2 (10.98)	$\{d^{shortEucl}$, $a = 0$, $k = 1\}$, $\mathbb{D}_{small} = [t_1, t_{17}]$
105 (6.94)	$\{d^{Eucl}$ with $x_i(t)$ centered, $a = 0$, $k = 5\}$
241 (4.43)	$\{d^{Eucl}$ with $x_i(t)$ centered, $a = 0$, $k = 21\}$

coefficient ID (estimated RFI values)	parameter tuple
csCATS-Lasso	
2 (10.04)	$\{d^{shortEucl}$, $a = 0$, $k = 1\}$, $\mathbb{D}_{small} = [t_1, t_{17}]$
241 (9.22)	$\{d^{Eucl}$ with $x_i(t)$ centered, $a = 0$, $k = 21\}$
205 (7.87)	$\{d^{Eucl}$, $a = 0$, $k = 21\}$
252 (5.58)	$\{d^{Max}$ with $x_i(t)$ centered, $a = 0$, $k = 21\}$
adaptive csCATS-Lasso	
241 (13.45)	$\{d^{Eucl}$ with $x_i(t)$ centered, $a = 0$, $k = 21\}$
2 (10.46)	$\{d^{shortEucl}$, $a = 0$, $k = 1\}$, $\mathbb{D}_{small} = [t_1, t_{17}]$
137 (6.35)	$\{d^{Eucl}$, $a = 0$, $k = 11\}$
252 (5.38)	$\{d^{Max}$ with $x_i(t)$ centered, $a = 0$, $k = 21\}$

Table 4.6: Selection results for the single penalties. First column: the IDs of the four estimated coefficients that show the largest RFI values, across all categories where appropriate (in brackets, in decreasing order). Second column: The chosen ensemble coefficients are decoded, with value a indicating the order of derivation and k indicating the number of nearest neighbors used.

205, 206, 214, and 241 (Lasso)/ 1, 2, 69, and 70 (cs-Lasso)/ 2, 70, 205, and 241 (csCATS-Lasso) for both the ordinary as well as the adaptive penalties, partly overlapping with the coefficients that have been estimated from the whole data shown in Table 4.6. The tuples corresponding to the unlisted IDs are $1 \hat{=} \{d^{Eucl},\ a = 0,\ k = 1\}$, $70 \hat{=} \{d^{shortEucl},\ a = 0,\ k = 5\}$ with $\mathbb{D}_{small} = [t_1, t_{17}]$, $206 \hat{=} \{d^{shortEucl},\ a = 0,\ k = 21\}$ with $\mathbb{D}_{small} = [t_1, t_{17}]$, and $214 \hat{=} \{d^{relAreas},\ a = 0,\ k = 21\}$ with $\mathbb{D}_1 = [t_{57}, t_{76}]$. It thus seems that the Euclidian distance between the raw or centered curves, eventually using only the very first part of the signal, and with a relatively high number of nearest neighbors $k \geq 5$, is the most important curve characteristic for this discrimination task.
When estimating the k-nearest-neigbor ensemble by means of minimizing the Brier score, selection results are identical to those in Chapter 3, Section 3.6. The coefficient IDs showing the five highest mean RFI values (in decreasing order) are 137, 205, 69, 105, and 1. Hence, the selection results of the two estimation approaches, i.e. Brier score minimization and the penalized cMLM approach, are consistent, differing solely in the RFI values.

Figure 4.11 gives the test results for all classification approaches on the basis of the 100 draws. The upper panel shows the Brier scores (for RDA, the estimated probabilities were not accessible), the lower panel the misclassification rates. To achieve a better resolution concerning the best performing methods' boxes, the y-scale has been pruned. The mean Brier scores of the methods LDA, SVM-FPCs, SVM-cov., and RF-FPCs are 0.089, 0.17, 0.29, and 0.089. The mean MCRs of LDA, SVM-FPCs, RF-FPCs, RDA, and SDA are 0.22, 0.8, 0.38, 0.81, and 0.95, respectively. As can be seen, the penalized cMLM is competitive compared to the other methods for all penalties, with the lowest Brier scores and MCR values among the best methods. Especially for the ordinary penalties, an improvement of prediction accuracy compared to the k-nearest-neighbor ensemble is revealed, emphasizing the advantages discussed in Section 4.2. The performance between the three penalty options Lasso, cs-Lasso and csCATS-Lasso shows that the csCATS-Lasso tends to somewhat lower values than the other penalties.

4.6 Discussion

We propose a functional classification approach that includes interpretable feature and variable selection by estimating a functional k-nearest-neighbor ensemble within a penalized and constrained multinomial logit model. The approach represents a synthesis of the methods introduced in Tutz et al. (2015) and Fuchs et al. (2015a). The strong performance is only obtained by the combination of the methods.
Setting up the functional ensemble can be seen as a dimension reduction approach that allows to detect the relevant features for classification given functional covariates. It makes efficient use of established tools and reliable and fast software. A large variety of penalties enables the user to define additional benefits arising with the discrimination itself. For ex-

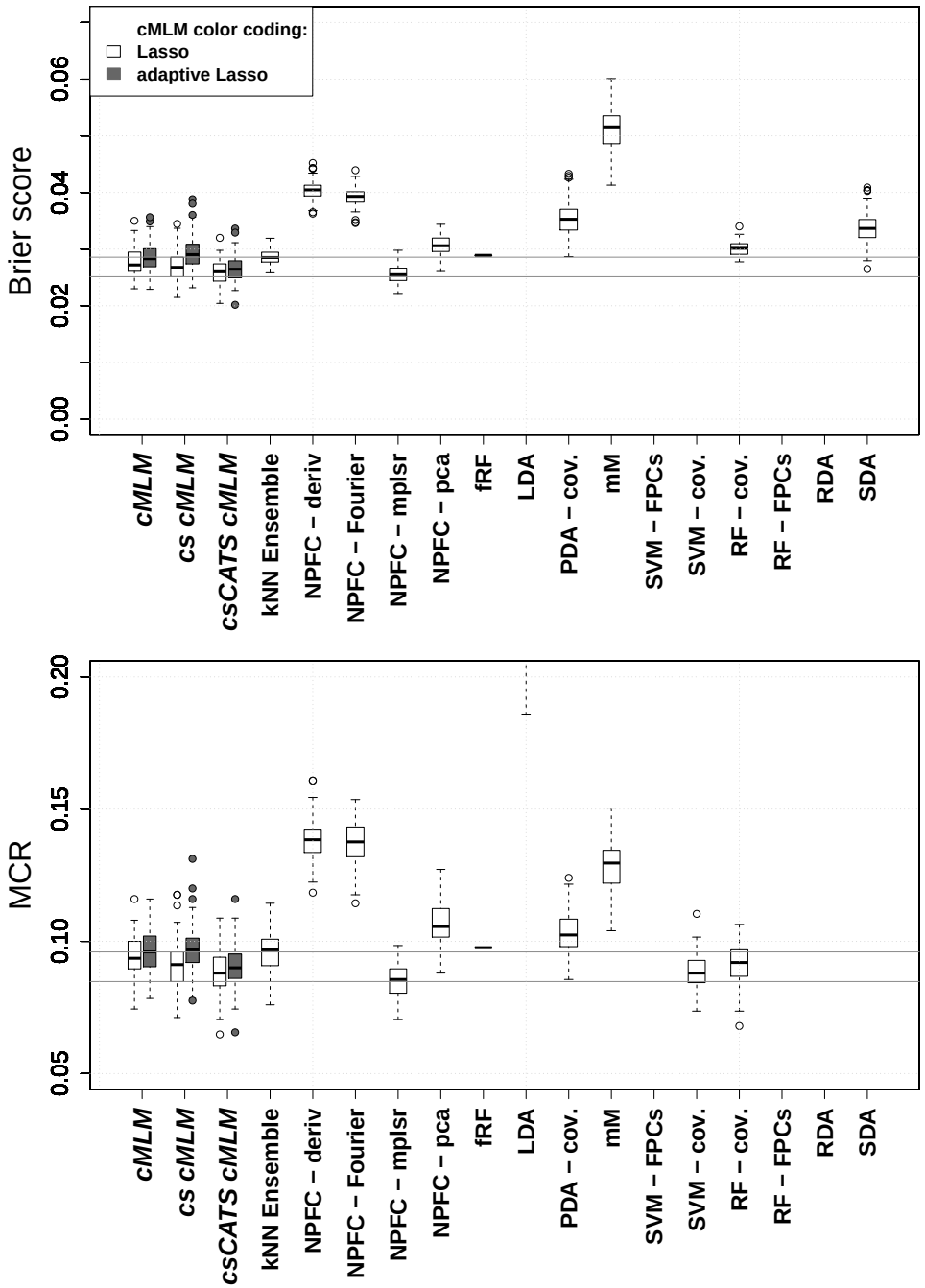


Figure 4.11: Test results of the phoneme data for all classification approaches on basis of 100 draws from the data. The upper panel shows the Brier scores. The boxes of the methods LDA, SVM-FPCs, SVM-cov., and RF-FPCs (mean values 0.089, 0.17, 0.29, and 0.089) are not shown due to y-axis pruning. In the lower panel the misclassification rates (MCR) are shown. The boxes of LDA, SVM-FPCs, RF-FPCs, RDA, and SDA are not shown due to y-axis pruning. They yield mean values of 0.22, 0.8, 0.38, 0.81, and 0.95, respectively. The horizontal lines indicate the values of the first and third quartiles of the category-specific cMLM box.

ample, the choice of a Lasso-type penalty results in feature selection. Since the functional k-nearest-neighbor ensemble is estimated within the MLM framework the estimated probabilities π_g of unknown observations $x^*(t)$ are automatically scaled to their natural domain $[0, 1]$, and the constraint $c_l \geq 0$ suffices to make the ensemble coefficients interpretable. This allows the use of category-specific coefficients if necessary. Moreover, the method can readily be adapted to ordinal responses. To summarize, the ensemble step enables a MLM to classify functional covariates, providing inputs that represent a wide variety of curve features. Meanwhile, the MLM step speeds up the estimation of the ensemble weights and allows for more sophisticated coefficient specification. The prediction results of especially the second generating process in the simulation study and both, the cell chip as well as the phoneme data, support the advantages of the method.

Since the number of data sets including multiple data is growing, the capability of the penalized cMLM to weight differing covariate types differently, up to selecting some types and dispense others, is of increasing interest in the context of functional classification. This was exemplified by the cell chip data. Although both functional covariate types, i.e. the IDES- and ISFET-signals, were selected, the chosen coefficients represented different curve characteristics. Thus, even if no variable selection takes place, one gets information concerning which features of which signal provide discriminative power, allowing for insight in the processes underlying the data.

We have shown that the estimation of a functional k-nearest-neighbor ensemble via a penalized cMLM is a powerful tool for the discrimination of functional data. Nevertheless, further extensions seem interesting. Concerning the ensemble, an especially worthwhile point to be considered is the choice of the semi-metrics. To be able to achieve data-driven feature selection, a large variety of semi-metrics should be incorporated in the ensemble. The semi-metrics and respective parameters have to be chosen by the user. A randomized choice of both, semi-metrics and semi-metric parameters where appropriate, possibly from predefined sets, seems a sensible enhancement of the k-nearest-neighbor ensemble. Also, multivariate and non-functional covariates could be included by suitable semi-metrics, as was already discussed in the conclusions of Chapter 3.
Concerning the penalized cMLM, the method can handle additional covariates corresponding to (random) batch or time independent effects by adapting the grouped Lasso penalty. There are also penalties that account for highly correlated inputs, see Tutz and Ulbricht (2009) or Bondell and Reich (2008) for exemplarily penalties applied to non-functional data. However, the combination of all issues mentioned, i.e. ordinal responses, random and time independent effects as well as highly correlated data, being incorporated into the penalized cMLM approach will probably rise other challenges that could be topics for future research.

Chapter 5

Discussion and Outlook

Especially during recent years, the optimization of already intergrated industrial components or the development of new markets results in a constantly growing amount of data. This includes data from experiments, e.g. gene experiments, as well as from empirical research of various kinds (social sciences, clinical trials, etc.) and automatically registered data, as for example in social media. Although data, depending on its' origin, can be of very different structures, problems occurring in data generation or interpretation often are similar: In experiments and empirical trials, the costs and accessibility of material as well as man power are a limiting factor for data generation. Additionally, the extraction of valid, reproducible, generalizing, and interpretable results becomes more difficult with increasing experiment complexity. This is also an important issue when handling automatically recorded or "big" data. Thus, the necessity of adequate statistical models increased with the amount and complexity of data. If a data set can be taken for the realization of underlying (quasi-) continuous functions, functional data analysis is a suitable tool here.

Ullah and Finch (2013) give a good overview of the palette of functional data and related methods. In this thesis, the data sets introduced in Chapter 1 are some of the functional data that is used to evaluate the performance of the developed models. The main interest concerning this data lies in regression and/or classification, and the variable and feature selection of functional covariates and their characteristics.
The cell chip data, yielding multiple functional covariates, inspired the extension of functional generalized linear models to include functional covariate interaction terms. In Chapter 2, such interaction terms are introduced. They are used to test the additivity assumption in a scalar-on-functions regression model context for simulated data and when modeling the cell chip and the spectroscopic fossil fuel data sets. The prediction performances of models with and without an interaction term are compared. Chapter 2 shows that the inclusion of an interaction term is worthwhile if the data comprises functional covariate interactions, and can enhance prediction performance. Our approach is suitable for models

including multiple functional and non-functional covariates as well as several error distributions (cf. for example Wood, 2017, entry "family.mgcv" for details). This covers many data sets, but one can think of various further developments. As already mentioned, the interaction term of two functional covariates can easily be extended to higher orders, i.e. to functional n-way interactions. Another obvious modification is the use of bases other than B-spline bases where it is more suitable for the data at hand, for example a Fourier basis for periodic data. Generally, an unresolved problem here is the choice of the number of basis functions, which is why we recommend a sensitivity analysis. An interesting point concerning the functional interaction term would be its extension to functional covariates of higher order, for instance, surfaces or images. Lastly, much research has to be done concerning the identifiability of functional interaction effect models, especially if containing functional interactions of higher order, or comprising high-dimensional functional covariates. For many data situations, the number of observations presumably will be too small to generate an identifiable model. After the development of diagnostic criteria for identifiability (following the example in Scheipl and Greven, 2016), the definition of some kind of uniform estimation procedure or a standard constraint for non-identifiable models would be sensible.

While Chapter 2 focuses on scalar-on-functions regression, the k-nearest-neighbor ensemble introduced in Chapter 3 can be used for feature and, in the presence of multiple functional covariates, variable selection. As a first step in the ensemble, a set of semi-metrics is defined in such a way that the semi-metrics represent characteristics of the covariates. Then, for each observation and semi-metric, the posterior probability estimate per class is calculated from the k nearest neighbors. "Nearness" here is specified by the respective semi-metrics, which serve as distance measures. Each single posterior probability estimate is multiplied with an unknown ensemble coefficient, i.e. weight, and all these products are combined linearly. The sum of the products yields the final class probability estimate per observation. The estimation of the ensemble coefficients is fulfilled via minimizing the Brier score, subject to coefficient constraints.
In Chapter 3, the k-nearest-neighbor ensemble is used for classification only. With appropriate modifications, it could also be used for regression tasks. No distribution assumptions are needed here, since the k-nearest-neighbor approach is non-parametric and the estimation is based on loss optimization. For classification tasks, so far, the overall probability that observation $x^*(t)$ belongs to class g is determined by the highest overall posterior probability across classes, $y^* = \text{argmax}_g\,(\hat{\pi}_g)$. This decision criterion could be reconsidered. For example, if G denotes the number of classes, a threshold $y^* = \text{argmax}_g\left(\hat{\pi}'_g\right)$ *with* $\left(\hat{\pi}'_g \in \{\hat{\pi}_g : |\hat{\pi}_g - 1/G| \geq \text{threshold}\}\right)$ could be implemented for very inhomogeneous data. The ensemble itself could be expanded to include members other than semi-metric based posterior probabilities, e.g. estimation results from other classification (or regression) models. However, the interpretability will become difficult after such an expansion, since the ensemble weights are no longer restricted to curve characteristics,

but also refer to results of afore estimated models. The modification of the presented linear ensemble to a non-linear ensemble of the form $\hat{\pi}_g = \sum_{l=1}^{p} f_l(c_l)\hat{\pi}_{g(l)}$, with $f_l(\cdot)$ being a smooth function of c_l, bears the risk of non-identifiability. Suitable constraints would have to be put on the functions $f_l(\cdot)$ to ensure unique estimation results. Further, the estimation of the ensemble coefficients via a loss minimization becomes a challenging task. A demanding and at the same time very interesting topic is the development of semi-metrics operating on higher dimensional functional data, such as surfaces. Another direction for future reseach is the development of adequate distance measures for non-functional covariates, so that the latter can be included in the k-nearest-neighbor ensemble. Although such distances have already been studied e.g. by the machine learning community, their choice should primarily be made with respect to the interpretability relative to the respective covariates.

Chapter 4 of this thesis deals with an alternative estimation approach for the k-nearest-neighbor ensemble. As before, semi-metrics are defined with respect to functional covariate characteristics, and corresponding posterior probabilities are calculated. In contrast to Chapter 3, the estimation is not accomplished by minimizing a loss, but by using the single posterior probabilities as inputs in a multinomial logit model and maximizing the respective log-likelihood. Thus, the domain of predicted values is inherently restricted to $[0, 1]$. This allows to relax the constraints put on the ensemble coefficients. The constraint $c_l \geq 0 \; \forall l$ is retained in such a way that the coefficients again can be interpreted as weights. In Chapter 4, it is illustrated that both approaches, the minimization of the (Brier) loss as well as a constraint and penalized MLM, are suitable for the estimation of the functional k-nearest-neighbor ensemble. It depends on the data at hand which method is to be preferred. In contrast to the loss approach, the penalized cMLM approach can readily be adapted to ordinal responses. Also, due to the above mild coefficient constraint, the penalized cMLM approach enables the user to choose from a broader variety of penalties, including class-specific penalties. Apart from those used in Chapter 4, the performance of further penalties could be examined. For both estimation approaches, the definition of interpretable semi-metrics for sparse (in the sense of few available observation points) functional data is an open problem. A simple workaround here is to smooth the sparse data or approximate it by some basis representation prior to employing the semi-metrics. However, information is lost in this approach if the missing data is informative, as for example in medical applications when a missing measurement is correlated to the health status of a patient. A different direction for research is the investigation of further estimation approaches for the k-nearest-neighbor ensemble. For example, the ensemble could be divided in several sums, each comprising only one semi-metric with several parameter choices. These sums could be used as (non-) linear base learners in a boosting approach. Another idea is to use the posterior probabilities as inputs in regression, classification or clustering approaches other than a MLM.

Appendix A

Appendices – Functional Covariate Interaction

A.1 Influence of Preprocessing – Detailed Plots

The following plots give details on the preprocessings introduced and disussed in Sections 2.5.2 and 2.6.2. The sequence of plots is as follows:
A first series of figures shows, for exemplarily preprocessings, boxplots of the adjusted R^2 (first panels) as well as the relative mean squared errors of prediction (relMSE$_{\hat{y}}$, second panels) and calibration (relMSE$_y$, third panels) across 25 random draws from the preprocessed data (cf. Sections 2.4.1 and 2.5.1 for the calculation of the relMSE). These boxplots are given for models with up to three main effects. Color coding is with respect to the numbers nk of marginal bases functions used in the modeling. The boxes for different nk per model type are superimposed. For each box, the respective mean value is plotted as a diamond in the same color.
After this series of plots, we determine, for each preprocessing, which number nk of marginal bases functions most frequently yields the smallest median relMSE$_{\hat{y}}$ across all model types. The corresponding figure shows boxplots of the R^2_{adj}, relMSE$_{\hat{y}}$, and relMSE$_y$ across the mentioned 25 random draws for the determined nk's. Here, color coding is with respect to the preprocessing.
The following figure shows the same, except that nk is chosen such that it corresponds to the nk which most frequently yields the smallest mean relMSE$_{\hat{y}}$ across all model types.

First, all these plots are given for the spectra data, with exemplifying preprocessing options 3 and 6 (cf. Table 2.2, Chapter 2.5.2).

R^2_{adj}
0.0
0.2
0.4
0.6
0.8
1.0
UV–VIS
NIR
UV–VIS + NIR
interaction: UV–VIS + NIR
full model: UV–VIS + NIR
relMSE$_{\hat{y}}$
relMSE$_y$
nk=4
nk=5
nk=6
nk=7
nk=8
nk=9
nk=10
mean values
Spectra data – preprocessing 3

Spectra data – preprocessing 6
R^2_{adj}
$relMSE_{\hat{y}}$
$relMSE_y$
0.0
0.2
0.4
0.6
0.8
1.0
UV–VIS
NIR
UV–VIS + NIR
interaction: UV–VIS + NIR
full model: UV–VIS + NIR
nk=4
nk=5
nk=6
nk=7
nk=8
nk=9
nk=10
mean values

R^2_{adj}
0.0
0.2
0.4
0.6
0.8
1.0
UV–VIS
NIR
UV–VIS + NIR
interaction: UV–VIS + NIR
full model: UV–VIS + NIR

nk–choice with respect to minimal median relMSE of prediction
$relMSE_{\hat{y}}$
0.0
0.2
0.4
0.6
0.8
1.0
UV–VIS
NIR
UV–VIS + NIR
interaction: UV–VIS + NIR
full model: UV–VIS + NIR

$relMSE_y$
0.0
0.2
0.4
0.6
0.8
1.0
UV–VIS
NIR
UV–VIS + NIR
interaction: UV–VIS + NIR
full model: UV–VIS + NIR
Preprocessing 1, nk=10
Preprocessing 2, nk=7
Preprocessing 3, nk=10
Preprocessing 4, nk=9
Preprocessing 5, nk=6
Preprocessing 6, nk=9
Preprocessing 7, nk=10
mean values

R^2_{adj}
0.0
0.2
0.4
0.6
0.8
1.0
UV–VIS
NIR
UV–VIS + NIR
interaction: UV–VIS + NIR
full model: UV–VIS + NIR

nk−choice with respect to minimal mean relMSE of prediction
$\text{relMSE}_{\hat{y}}$
0.0
0.2
0.4
0.6
0.8
1.0
UV–VIS
NIR
UV–VIS + NIR
interaction: UV–VIS + NIR
full model: UV–VIS + NIR

relMSE_y
0.0
0.2
0.4
0.6
0.8
1.0
Preprocessing 1, nk=10
Preprocessing 2, nk=10
Preprocessing 3, nk=8
Preprocessing 4, nk=9
Preprocessing 5, nk=6
Preprocessing 6, nk=6
Preprocessing 7, nk=6
mean values
UV–VIS
NIR
UV–VIS + NIR
interaction: UV–VIS + NIR
full model: UV–VIS + NIR

The following sequence of plots shows the same as before, but for the cell chip data and exemplifying preprocessing options 4 and 5 (cf. Table 2.3, Chapter 2.6.2).
Additionally, the analogs to Figure 2.18 are given for the covariate combinations ISFET and IDES as well as IDES and CLARK.

Cell chip data – preprocessing 4
R^2_{adj}
$relMSE_{\hat{y}}$
$relMSE_y$
0.0
0.2
0.4
0.6
0.8
1.0
ISFET
IDES
CLARK
ISFET + IDES
interaction: ISFET + IDES
full model: ISFET + IDES
ISFET + CLARK
interaction: ISFET + CLARK
full model: ISFET + CLARK
IDES + CLARK
interaction: IDES + CLARK
full model: IDES + CLARK
ISFET + IDES + CLARK
nk=4
nk=5
nk=6
nk=10
mean values

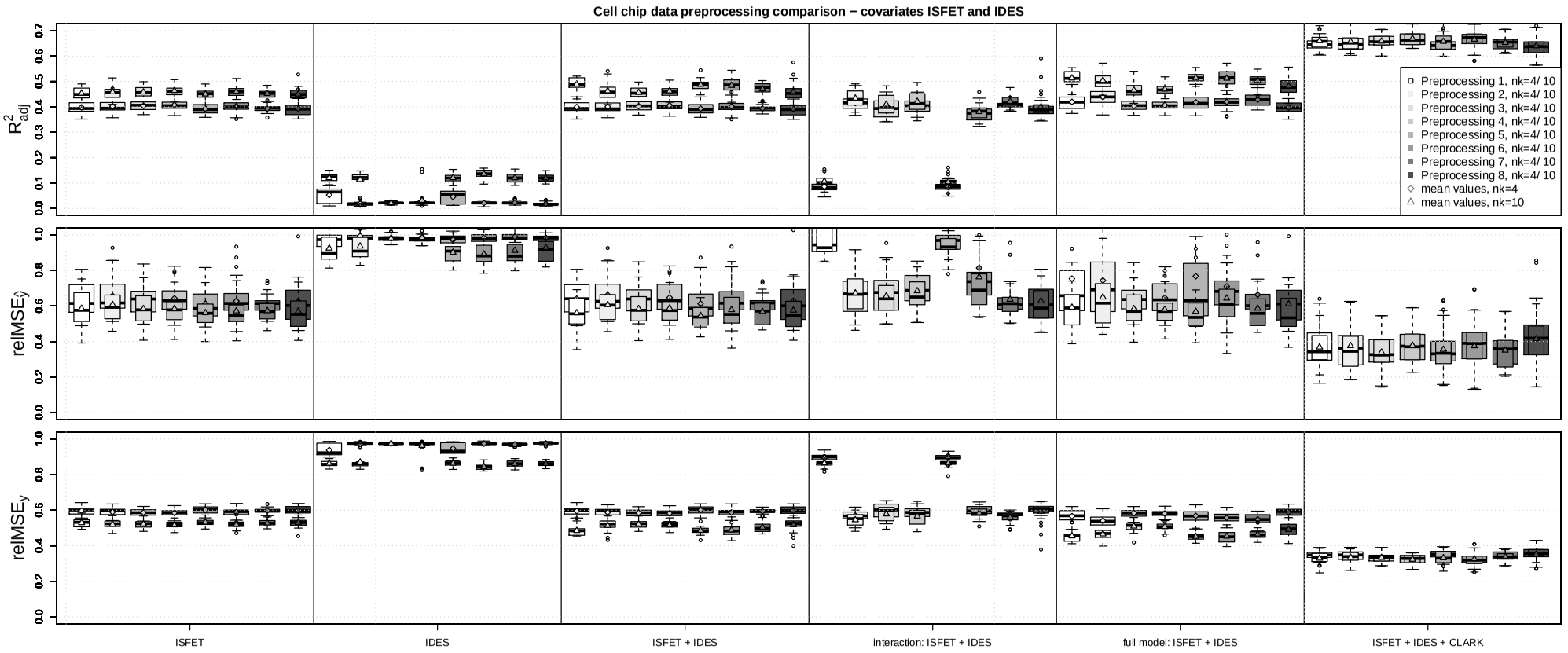
Cell chip data – preprocessing 5
R^2_{adj}
$relMSE_{\hat{y}}$
$relMSE_y$
0.0
0.2
0.4
0.6
0.8
1.0
ISFET
IDES
CLARK
ISFET + IDES
interaction: ISFET + IDES
full model: ISFET + IDES
ISFET + CLARK
interaction: ISFET + CLARK
full model: ISFET + CLARK
IDES + CLARK
interaction: IDES + CLARK
full model: IDES + CLARK
ISFET + IDES + CLARK
nk=4
nk=5
nk=6
nk=10
mean values

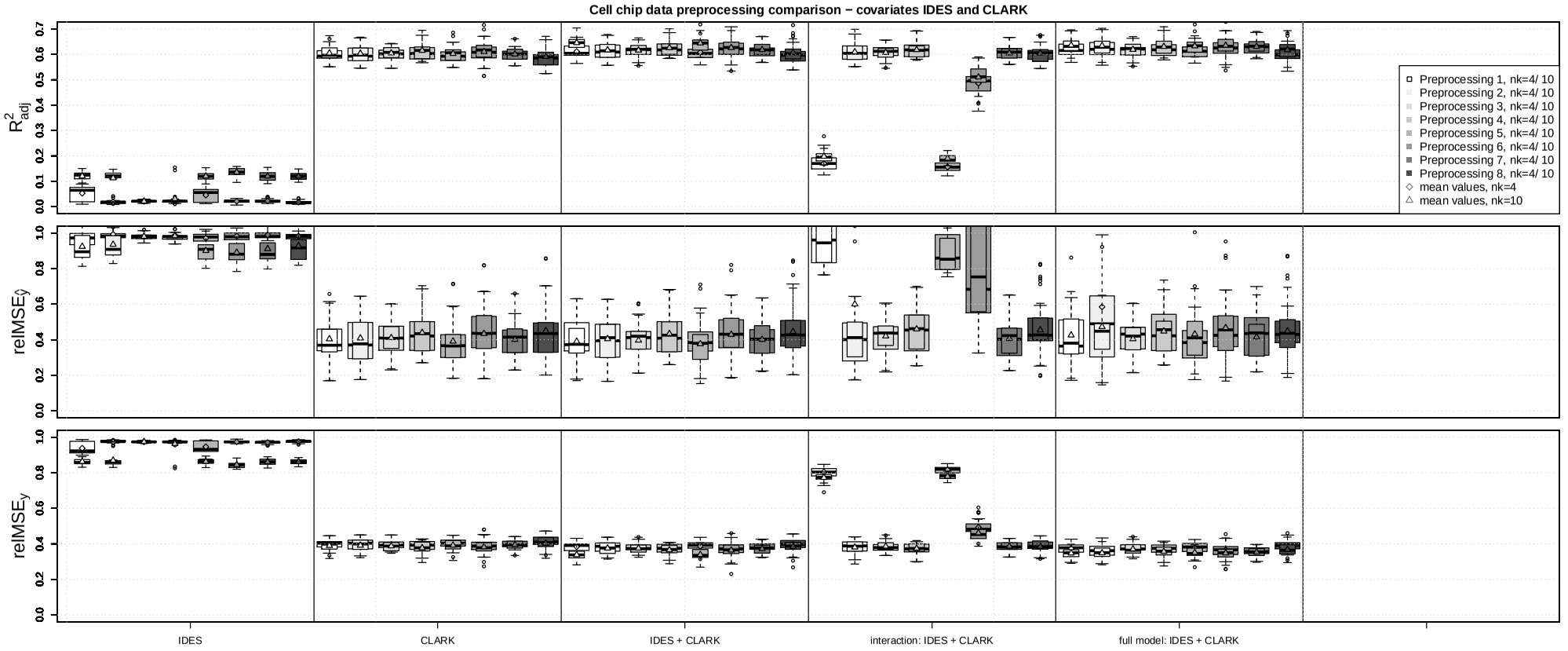
R^2_{adj}
0.0
0.2
0.4
0.6
0.8
1.0
ISFET
IDES
CLARK
ISFET + IDES
interaction: ISFET + IDES
full model: ISFET + IDES
ISFET + CLARK
interaction: ISFET + CLARK
full model: ISFET + CLARK
IDES + CLARK
interaction: IDES + CLARK
full model: IDES + CLARK
ISFET + IDES + CLARK

nk−choice with respect to the minimal median relMSE of prediction
$relMSE_{\hat{y}}$
0.0
0.2
0.4
0.6
0.8
1.0
ISFET
IDES
CLARK
ISFET + IDES
interaction: ISFET + IDES
full model: ISFET + IDES
ISFET + CLARK
interaction: ISFET + CLARK
full model: ISFET + CLARK
IDES + CLARK
interaction: IDES + CLARK
full model: IDES + CLARK
ISFET + IDES + CLARK

$relMSE_y$
0.0
0.2
0.4
0.6
0.8
1.0
Preprocessing 1, nk=10
Preprocessing 2, nk=10
Preprocessing 3, nk=10
Preprocessing 4, nk=10
Preprocessing 5, nk=10
Preprocessing 6, nk=10
Preprocessing 7, nk=10
Preprocessing 8, nk=10
mean values
ISFET
IDES
CLARK
ISFET + IDES
interaction: ISFET + IDES
full model: ISFET + IDES
ISFET + CLARK
interaction: ISFET + CLARK
full model: ISFET + CLARK
IDES + CLARK
interaction: IDES + CLARK
full model: IDES + CLARK
ISFET + IDES + CLARK

R^2_{adj}
relMSE$_{\hat{y}}$
relMSE$_y$
nk−choice with respect to the minimal mean relMSE of prediction
0.0
0.2
0.4
0.6
0.8
1.0
ISFET
IDES
CLARK
ISFET + IDES
interaction: ISFET + IDES
full model: ISFET + IDES
ISFET + CLARK
interaction: ISFET + CLARK
full model: ISFET + CLARK
IDES + CLARK
interaction: IDES + CLARK
full model: IDES + CLARK
ISFET + IDES + CLARK
Preprocessing 1, nk=10
Preprocessing 2, nk=10
Preprocessing 3, nk=10
Preprocessing 4, nk=6
Preprocessing 5, nk=10
Preprocessing 6, nk=10
Preprocessing 7, nk=10
Preprocessing 8, nk=10
mean values


Cell chip data preprocessing comparison − covariates ISFET and IDES
R^2_{adj}
$relMSE_{\hat{y}}$
$relMSE_y$
Preprocessing 1, nk=4/ 10
Preprocessing 2, nk=4/ 10
Preprocessing 3, nk=4/ 10
Preprocessing 4, nk=4/ 10
Preprocessing 5, nk=4/ 10
Preprocessing 6, nk=4/ 10
Preprocessing 7, nk=4/ 10
Preprocessing 8, nk=4/ 10
mean values, nk=4
mean values, nk=10
ISFET
IDES
ISFET + IDES
interaction: ISFET + IDES
full model: ISFET + IDES
ISFET + IDES + CLARK


Cell chip data preprocessing comparison − covariates IDES and CLARK
R^2_{adj}
$relMSE_{\hat{y}}$
$relMSE_y$
Preprocessing 1, nk=4/ 10
Preprocessing 2, nk=4/ 10
Preprocessing 3, nk=4/ 10
Preprocessing 4, nk=4/ 10
Preprocessing 5, nk=4/ 10
Preprocessing 6, nk=4/ 10
Preprocessing 7, nk=4/ 10
Preprocessing 8, nk=4/ 10
mean values, nk=4
mean values, nk=10
IDES
CLARK
IDES + CLARK
interaction: IDES + CLARK
full model: IDES + CLARK

A.2 Detailed Derivations of the Equations Concerning the Identifiability in the Context of Scalar-on-Functions Regression

In the following, the derivations of the equations in Section 2.7 are given in detail. The definitions and nomenclature of Section 2.7 are retained.

The integral of the product of two functional covariates $X_1(s)$, $X_2(t)$ and a surface $o(s,t)$ can be simplified to

$$\begin{aligned}
\int_{\mathcal{T}}\int_{\mathcal{S}} X_1(s)X_2(t)o(s,t)dsdt &= \int_{\mathcal{T}}\int_{\mathcal{S}} \sum_{q_1=1}^{Q_1} \iota_{q_1}\chi_{q_1}(s) \sum_{q_2=1}^{Q_2} \iota_{q_2}\chi_{q_2}(t) \sum_{q_s=1}^{\infty}\sum_{q_t=1}^{\infty} \kappa_{q_sq_t}\chi_{q_s}(s)\chi_{q_t}(t)dsdt \\
&= \sum_{q_1=1}^{Q_1} \iota_{q_1} \sum_{q_2=1}^{Q_2} \iota_{q_2} \int_{\mathcal{T}}\int_{\mathcal{S}} \chi_{q_1}(s)\chi_{q_2}(t) \sum_{q_s=1}^{\infty}\sum_{q_t=1}^{\infty} \kappa_{q_sq_t}\chi_{q_s}(s)\chi_{q_t}(t)dsdt \\
&= \sum_{q_1=1}^{Q_1} \iota_{q_1} \sum_{q_2=1}^{Q_2} \iota_{q_2} \sum_{q_s=1}^{\infty}\sum_{q_t=1}^{\infty} \kappa_{q_sq_t} \int_{\mathcal{T}}\int_{\mathcal{S}} \chi_{q_1}(s)\chi_{q_2}(t)\chi_{q_s}(s)\chi_{q_t}(t)dsdt \\
&= \sum_{q_1=1}^{Q_1} \iota_{q_1} \sum_{q_2=1}^{Q_2} \iota_{q_2} \sum_{q_s=1}^{\infty}\sum_{q_t=1}^{\infty} \kappa_{q_sq_t} \int_{\mathcal{T}} \chi_{q_2}(t)\chi_{q_t}(t) \int_{\mathcal{S}} \chi_{q_1}(s)\chi_{q_s}(s)dsdt \\
&\quad \textit{$\chi_{q_1}(s)$ and $\chi_{q_s}(s)$ are both orthonormal basis functions of $L^2(\mathcal{S})$,} \\
&\quad \textit{thus } \int_{\mathcal{S}} \chi_{q_1}(s)\chi_{q_s}(s)ds = \delta_{q_1q_s} \\
&= \sum_{q_1=1}^{Q_1} \iota_{q_1} \sum_{q_2=1}^{Q_2} \iota_{q_2} \sum_{q_t=1}^{\infty} \kappa_{q_1q_t} \int_{\mathcal{T}} \chi_{q_2}(t)\chi_{q_t}(t)dt \\
&\quad \textit{analogously } \int_{\mathcal{T}} \chi_{q_2}(t)\chi_{q_t}(t)dt = \delta_{q_2q_t} \\
&= \sum_{q_1=1}^{Q_1}\sum_{q_2=1}^{Q_2} \iota_{q_1}\iota_{q_2}\kappa_{q_1q_2}.
\end{aligned}$$

In equation (2.9), the surface itself is split into finite and infinite sums of bases coefficients and functions,

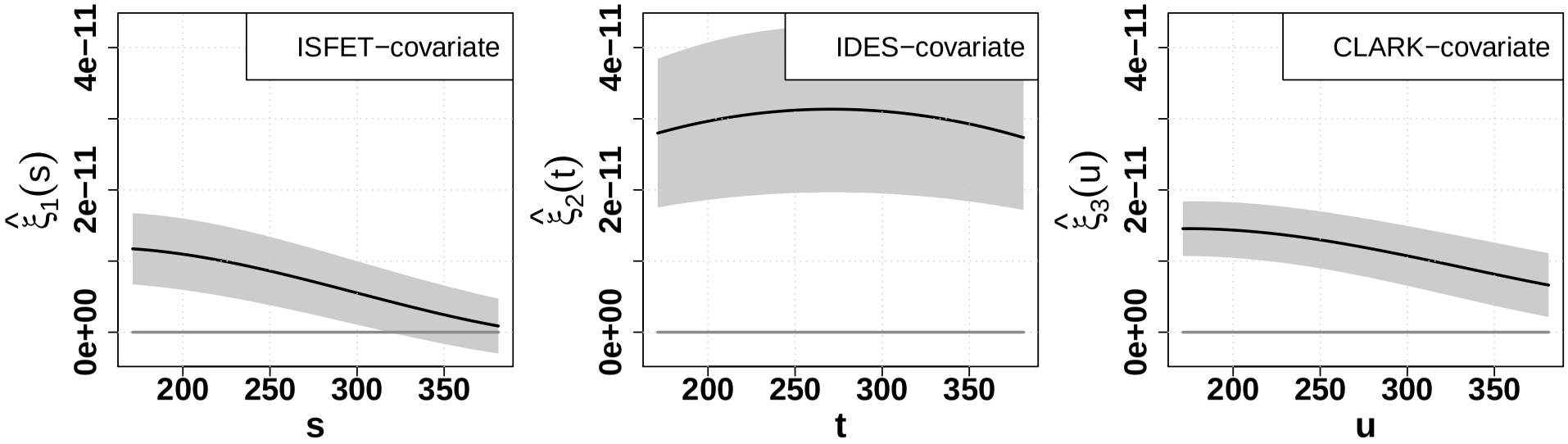
$$
\begin{aligned}
o(s,t) &= \sum_{q_s=1}^{\infty}\sum_{q_t=1}^{\infty}\kappa_{q_sq_t}\chi_{q_s}(s)\chi_{q_t}(t) \\
&\quad \textit{split the first sum} \\
&= \sum_{q_t=1}^{\infty}\kappa_{1q_t}\chi_1(s)\chi_{q_t}(t) + \sum_{q_t=1}^{\infty}\kappa_{2q_t}\chi_2(s)\chi_{q_t}(t) + \ldots \\
&\quad \textit{let there be } \infty > Q_2 \in \mathbb{N} \\
&= \left(\sum_{q_t=1}^{Q_2}\kappa_{1q_t}\chi_1(s)\chi_{q_t}(t) + \sum_{q_t=Q_2+1}^{\infty}\kappa_{1q_t}\chi_1(s)\chi_{q_t}(t)\right) + \\
&\quad \left(\sum_{q_t=1}^{Q_2}\kappa_{2q_t}\chi_2(s)\chi_{q_t}(t) + \sum_{q_t=Q_2+1}^{\infty}\kappa_{2q_t}\chi_2(s)\chi_{q_t}(t)\right) + \ldots \\
&\quad \textit{rearrange summands} \\
&= \sum_{q_t=1}^{Q_2}\kappa_{1q_t}\chi_1(s)\chi_{q_t}(t) + \sum_{q_t=1}^{Q_2}\kappa_{2q_t}\chi_2(s)\chi_{q_t}(t) + \ldots \\
&\quad \sum_{q_t=Q_2+1}^{\infty}\kappa_{1q_t}\chi_1(s)\chi_{q_t}(t) + \sum_{q_t=Q_2+1}^{\infty}\kappa_{2q_t}\chi_2(s)\chi_{q_t}(t) + \ldots = \sum_{q_s=1}^{\infty}\sum_{q_t=1}^{Q_2}\kappa_{q_sq_t}\chi_{q_s}(s)\chi_{q_t}(t) + \\
&\qquad\qquad \sum_{q_s=1}^{\infty}\sum_{q_t=Q_2+1}^{\infty}\kappa_{q_sq_t}\chi_{q_s}(s)\chi_{q_t}(t) \\
&\quad \textit{let there be } \infty > Q_1 \in \mathbb{N} \\
&= \sum_{q_t=1}^{Q_2}\kappa_{1q_t}\chi_1(s)\chi_{q_t}(t) + \sum_{q_t=1}^{Q_2}\kappa_{2q_t}\chi_2(s)\chi_{q_t}(t) + \ldots + \\
&\quad \sum_{q_t=1}^{Q_2}\kappa_{Q_1q_t}\chi_{Q_1}(s)\chi_{q_t}(t) + \sum_{q_t=1}^{Q_2}\kappa_{(Q_1+1)q_t}\chi_{Q_1+1}(s)\chi_{q_t}(t) + \ldots \\
&\quad \sum_{q_t=Q_2+1}^{\infty}\kappa_{1q_t}\chi_1(s)\chi_{q_t}(t) + \sum_{q_t=Q_2+1}^{\infty}\kappa_{2q_t}\chi_2(s)\chi_{q_t}(t) + \ldots \\
&\quad \textit{summarize sums} \\
&= \sum_{q_s=1}^{Q_1}\sum_{q_t=1}^{Q_2}\kappa_{q_sq_t}\chi_{q_s}(s)\chi_{q_t}(t) + \sum_{q_s=Q_1+1}^{\infty}\sum_{q_t=1}^{Q_2}\kappa_{q_sq_t}\chi_{q_s}(s)\chi_{q_t}(t) + \\
&\quad \sum_{q_s=1}^{Q_1}\sum_{q_t=Q_2+1}^{\infty}\kappa_{q_sq_t}\chi_{q_s}(s)\chi_{q_t}(t) + \sum_{q_s=Q_1+1}^{\infty}\sum_{q_t=Q_2+1}^{\infty}\kappa_{q_sq_t}\chi_{q_s}(s)\chi_{q_t}(t).
\end{aligned}
$$

The covariance operator of $X(s,t)$ applied to $o(s,t)$ is derived via

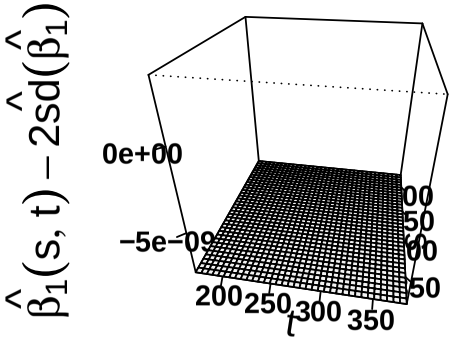
$$
\begin{aligned}
(V_X o)(s,t) &= \int_{\mathcal{T}}\int_{\mathcal{S}} \mathbb{E}\left\{X(s,t)X(u,v)\right\} o(u,v)dudv \\
&\quad \textit{basis expansion of } X(\cdot,\cdot) \\
&= \int_{\mathcal{T}}\int_{\mathcal{S}} \mathbb{E}\left\{\sum_{q_s=1}^{Q_s} \iota_{q_s}\chi_{q_s}(s) \sum_{q_t=1}^{Q_t} \iota_{q_t}\chi_{q_t}(t) \sum_{q_u=1}^{Q_u} \iota_{q_u}\chi_{q_u}(u) \sum_{q_v=1}^{Q_v} \iota_{q_v}\chi_{q_v}(v)\right\} o(u,v)dudv \\
&= \int_{\mathcal{T}}\int_{\mathcal{S}} \sum_{q_s=1}^{Q_s}\sum_{q_t=1}^{Q_t}\sum_{q_u=1}^{Q_u}\sum_{q_v=1}^{Q_v} \mathbb{E}\left\{\iota_{q_s}\iota_{q_t}\iota_{q_u}\iota_{q_v}\right\} \chi_{q_s}(s)\chi_{q_t}(t)\chi_{q_u}(u)\chi_{q_v}(v) o(u,v)dudv \\
&\quad X_1(s),\ X_2(t) \textit{ independent} \\
&= \int_{\mathcal{T}}\int_{\mathcal{S}} \sum_{q_s=1}^{Q_s}\sum_{q_t=1}^{Q_t}\sum_{q_u=1}^{Q_u}\sum_{q_v=1}^{Q_v} \mathbb{E}\left\{\iota_{q_s}\iota_{q_u}\right\}\mathbb{E}\left\{\iota_{q_t}\iota_{q_v}\right\} \chi_{q_s}(s)\chi_{q_t}(t)\chi_{q_u}(u)\chi_{q_v}(v) o(u,v)dudv \\
&\quad \mathbb{E}(\iota_{q_s}\iota_{q_u}) = \delta_{q_s q_u}\nu_{q_s},\ \textit{since } \iota_{q_s} \textit{ and } \iota_{q_u} \textit{ are per definition uncorrelated} \\
&= \int_{\mathcal{T}}\int_{\mathcal{S}} \sum_{q_s=1}^{Q_s}\sum_{q_t=1}^{Q_t}\sum_{q_u=1}^{Q_u}\sum_{q_v=1}^{Q_v} \delta_{q_s q_u}\nu_{q_s}\delta_{q_t q_v}\nu_{q_t} \chi_{q_s}(s)\chi_{q_t}(t)\chi_{q_u}(u)\chi_{q_v}(v) o(u,v)dudv \\
&= \int_{\mathcal{T}}\int_{\mathcal{S}} \sum_{q_s=1}^{Q_s}\sum_{q_t=1}^{Q_t} \nu_{q_s}\nu_{q_t} \chi_{q_s}(s)\chi_{q_t}(t)\chi_{q_s}(u)\chi_{q_t}(v) o(u,v)dudv \\
&\quad \textit{basis expansion of } o(u,v) \\
&= \int_{\mathcal{T}}\int_{\mathcal{S}} \sum_{q_s=1}^{Q_s}\sum_{q_t=1}^{Q_t} \nu_{q_s}\nu_{q_t} \chi_{q_s}(s)\chi_{q_t}(t)\chi_{q_s}(u)\chi_{q_t}(v) \sum_{q_u=1}^{\infty}\sum_{q_v=1}^{\infty} \kappa_{q_u q_v}\chi_{q_u}(u)\chi_{q_v}(v) dudv \\
&= \sum_{q_s=1}^{Q_s}\sum_{q_t=1}^{Q_t} \nu_{q_s}\nu_{q_t} \chi_{q_s}(s)\chi_{q_t}(t) \sum_{q_u=1}^{\infty}\sum_{q_v=1}^{\infty} \kappa_{q_u q_v} \int_{\mathcal{T}}\int_{\mathcal{S}} \chi_{q_s}(u)\chi_{q_t}(v)\chi_{q_u}(u)\chi_{q_v}(v) dudv \\
&= \sum_{q_s=1}^{Q_s}\sum_{q_t=1}^{Q_t} \nu_{q_s}\nu_{q_t} \chi_{q_s}(s)\chi_{q_t}(t) \sum_{q_u=1}^{\infty}\sum_{q_v=1}^{\infty} \kappa_{q_u q_v} \int_{\mathcal{T}} \chi_{q_t}(v)\chi_{q_v}(v) \int_{\mathcal{S}} \chi_{q_s}(u)\chi_{q_u}(u) dudv \\
&\quad \chi_{q_s}(u) \textit{ and } \chi_{q_u}(u) \textit{ are both orthonormal basis functions of } L^2(\mathcal{S}) \\
&= \sum_{q_s=1}^{Q_s}\sum_{q_t=1}^{Q_t} \nu_{q_s}\nu_{q_t} \chi_{q_s}(s)\chi_{q_t}(t) \sum_{q_v=1}^{\infty} \kappa_{q_s q_v} \int_{\mathcal{T}} \chi_{q_t}(v)\chi_{q_v}(v) dv \\
&\quad \chi_{q_t}(v) \textit{ and } \chi_{q_v}(v) \textit{ are both orthonormal basis functions of } L^2(\mathcal{T}) \\
&= \sum_{q_s=1}^{Q_s}\sum_{q_t=1}^{Q_t} \nu_{q_s}\nu_{q_t}\kappa_{q_s q_t} \chi_{q_s}(s)\chi_{q_t}(t).
\end{aligned}
$$

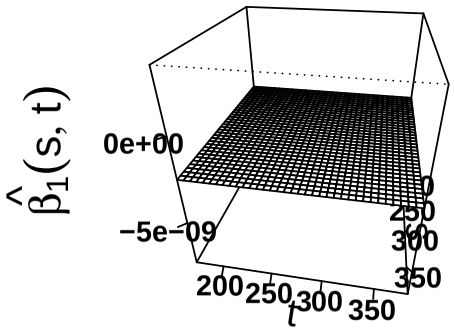
A.3 Estimates of the Three-Way Interaction Model Applied to the Cell Chip Data

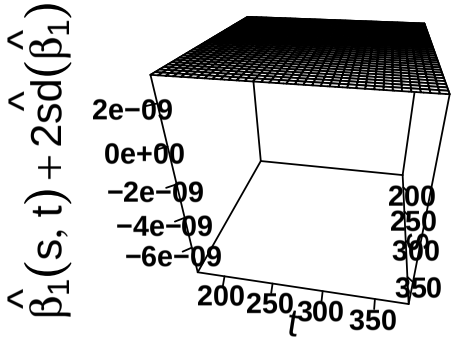
The following plots show the estimates of the three-way interaction model (2.14) applied to the full cell chip data, being typical also for the single replicates.
The three main effect estimates are depicted as lines (black), with pointwise confidence bands (gray). The three two-way interaction effect estimates are depicted as surfaces (middle panels), together with the estimated surfaces ± two times the estimated standard errors (left and right panels).
Since the coefficient $\beta(s,t,u)$ of the three-way interaction effect is three-dimensional, its estimate is visualized in two different ways. First, for each observation point $s_z \in \{s_1, \dots s_{36}\}$ of the ISFET covariate, the means $\int \hat{\beta}(s_z,t,u)dt$ and $\int \hat{\beta}(s_z,t,u)du$ across the t- and u-directions are shown. One can find that the orientation of $\hat{\beta}(s,t,u)$ does not change for both directions, but the absolute level increases with increasing s. In the second illustration of $\hat{\beta}(s,t,u)$, this result is confirmed. Here, the estimated surfaces $\hat{\beta}(s_z,t,u)$ (middle panels) as well as $\hat{\beta}(s_z,t,u)$ ± two times the estimated standard errors (left and right) are shown for $s_z \in \{s_1, s_{18}, s_{36}\}$. Browsing along increasing values of s_z shows that the estimated surface remains a negative plane, tilted downwards with decreasing values of u and increasing absolute level when s_z increases.


ISFET−covariate
IDES−covariate
CLARK−covariate
$\hat{\xi}_1(s)$
$\hat{\xi}_2(t)$
$\hat{\xi}_3(u)$
0e+00
2e−11
4e−11
200
250
300
350
s
t
u

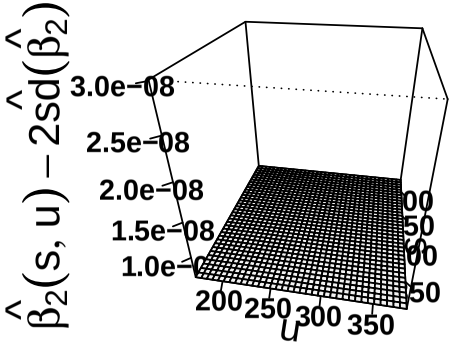
covariates ISFET/ IDES

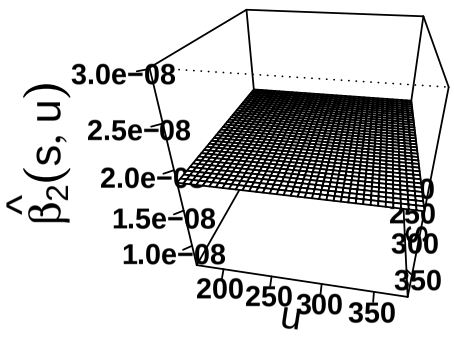


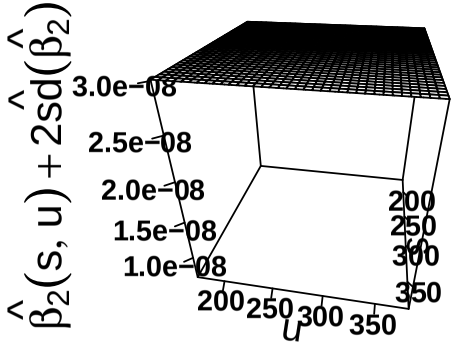




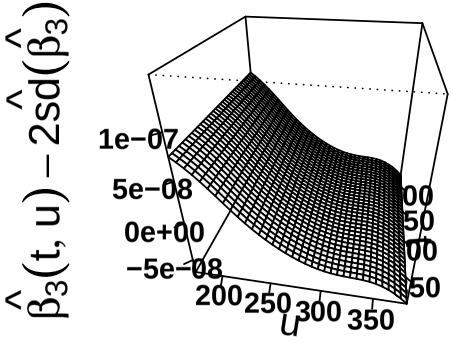
covariates ISFET/ CLARK

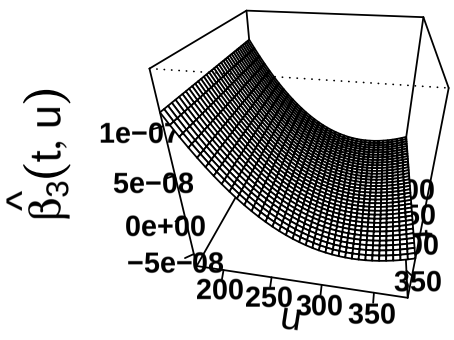


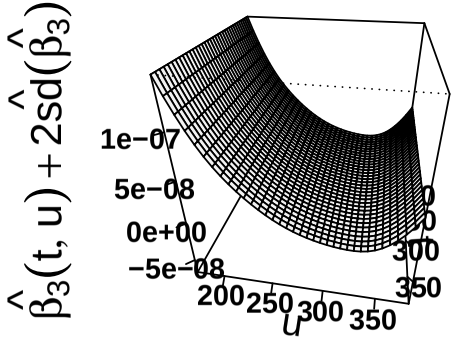




covariates IDES/ CLARK

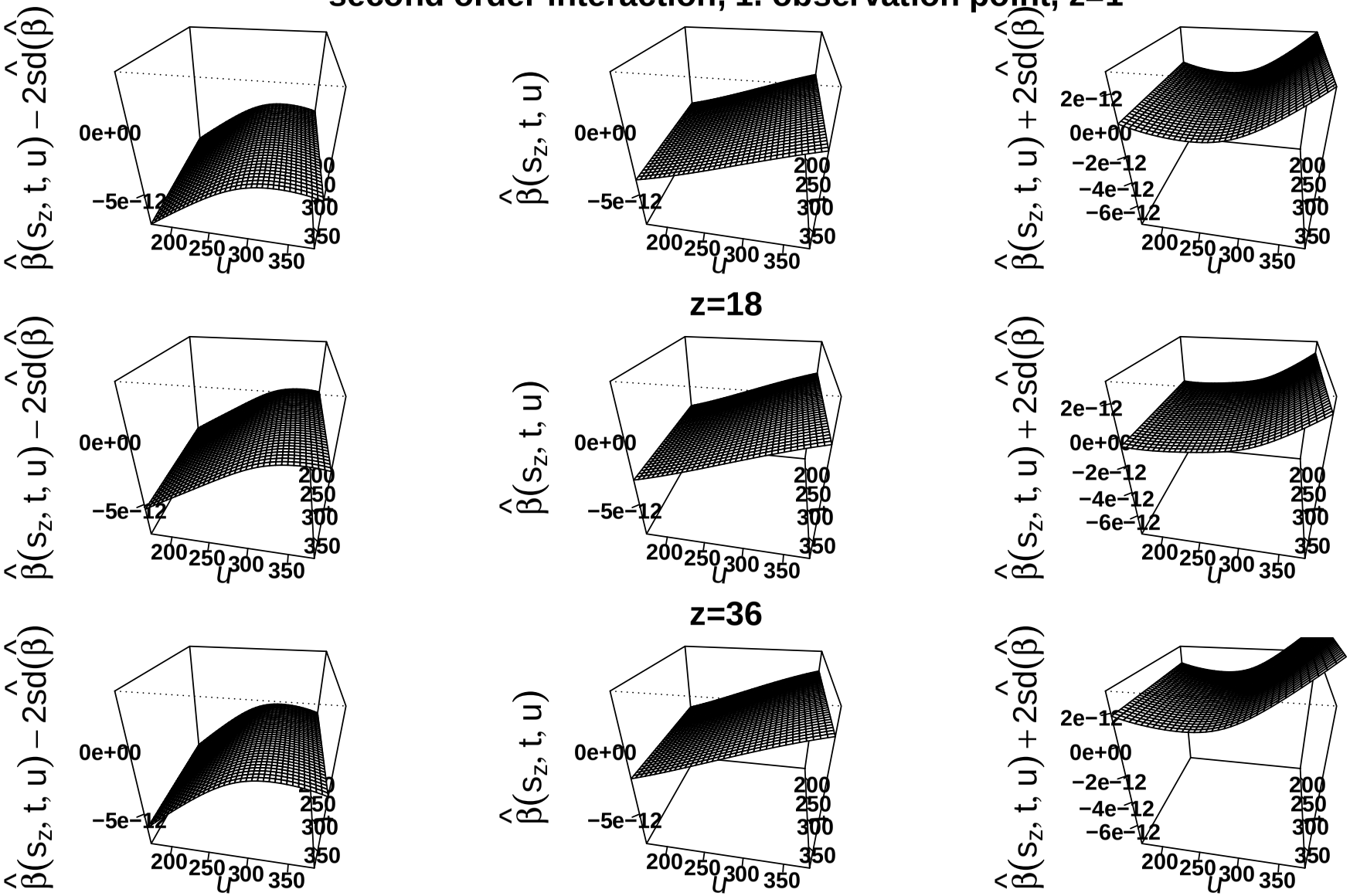






$\int \hat{\beta}(s_z, t, u)\, dt$
−3e−12
0e+00
200
250
300
350
z=1
...
z=20
...
z=36
u
$\int \hat{\beta}(s_z, t, u)\, du$
−2.0e−12
−5.0e−13
200
250
300
350
t

second order interaction, 1. observation point, z=1



Appendix B

Appendices – Functional Nearest Neighbor Ensembles

B.1 Functional Principal Components

Analogously to the multivariate case, the functional principle components (FPC) decomposition is a method to project the feature space on the eigenfunction space of the covariates' covariance matrix. The implementation of the `fpca.sc`-function of the `R`-package `refund` (Crainiceanu et al., 2013; R Core Team, 2017) calculates the functional principal component decomposition as follows:
The overall mean function $\mu(t)$ of all functional covariates $x_i(t)$, $i = 1, \dots N$, is estimated via a penalized spline fit, with the smoothing parameter estimated using restricted maximum likelihood (REML). The covariance matrices $\mathbf{\Sigma}_i(t, t_0) = \text{cov}\left(x_i(t), x_i(t_0)\right)$ per curve $x_i(t)$ are estimated in two steps. First, a so-called raw estimate is directly computed from the observed curves at the Q discrete observation points $t, t_0 \in \{t_1, \dots t_Q\} \subseteq \mathbb{D}$, where $\mathbb{D}$ denotes the domain of definition of $x_i(t)$ $\forall i$. This raw surface is smoothed using bivariate penalized splines, where the smoothing parameters again are estimated via REML.
The E eigenfunctions $\Phi_e(t)$ and eigenvalues λ_e, $e = 1 \dots E$, of this smoothed covariance matrix constitute the functional principal component basis functions and score variances. If the underlying process is known, one can even give the formulae of the eigenfunctions (which is for example possible for a Wiener process). E is chosen such that at least 95% of the curves' variability is explained. This means that E is the minimal value for which $\sum_{e=1}^{E} \lambda_e \geq 0.95$, with $E \leq \min(N, Q)$ and $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{\min(N,Q)+1} = 0$. The functional principal component scores ξ_{ie} are estimated from the curves $x_i(t)$ and the estimated quantities via best linear unbiased predictors basing on a mixed model of the form

$$x_i(t_q) = \mu(t_q) + \sum_{e=1}^{E} \xi_{ie} \Phi_e(t_q) + \varepsilon_i(t_q),$$

assuming $\boldsymbol{\xi}_i = (\xi_{i1}, \dots \xi_{iE}) \underset{iid}{\sim} N(\mathbf{0}, \boldsymbol{\Lambda})$, $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \dots \lambda_E)$,
$\varepsilon_i(t_q) \underset{iid}{\sim} N(\mathbf{0}, \boldsymbol{\sigma}^2)$ and $t_q \in \mathbb{D}$ (Di et al., 2009; Goldsmith et al., 2013).
With that, one gets E-sized, curve-specific vectors of FPC scores, which can be taken as multivariate representations of the functional covariates. The FPC score vectors are often used as covariates in a multivariate model.
For a short review on FPCA, see Shang (2014); for details on non-functional PCA see, for example, Jolliffe et al. (1996) and Jolliffe (2002).

B.2 Effect of the Number of Principal Components on Prediction

The principal components that explain the largest proportion of variance in the predictors are not necessarily those with highest discriminative power. Therefore, Epifanio and Ventura-Campos (2011) compute all functional principal component scores (one should keep in mind that, in the functional context, "all" means a number $\nu = \min(N, Q)$, with N denoting the number of functional observations and Q denoting the number of observation points). In the following analysis they include only those scores that exhibit significant differences (i.e., a p-value≤ 0.05 of a t-test) between the classes of their two-class response. On the other hand, however, it is not clear either that these components with largest bivariate effects are those with best performance in a multivariate setting.
That is why we chose to use a number of functional principal components that explains a considerably large proportion, at least 95%, of the variability in the data. Figure B.1 gives the frequencies of the numbers of the FPC of all replications of training data, for all examined data sets. For most data sets, more than 3 components were chosen. Only in the first generating process and the waveform data, ≤ 2 components seemed enough to capture the main curve characteristics.
In our analysis, we also calculated the functional principal components that explain at least 99.9% of the variability and compared the number of scores used in the multivariate methods to the predictive performances, see Figures B.2 to B.4. For all data sets and multivariate methods, it is obvious that results including components with small eigenvalues do not notably reduce the Brier scores and misclassification rates compared to a low number of components. Thus, the components explaining at least 95% of the variability (typically 1-6 components) seemed sufficient in the comparison of the classification methods.
Nonetheless, Figures B.2 to B.4 give a hint concerning the relatively high Brier score values of the SVM approach. While the misclassification rates behave similar across all multivariate methods for all data sets, the Brier scores of the SVM often show an opposing trend or an offset for more than one component. It seems that SVM can not reflect the underlying true probability distributions as well as the other multivariate methods.

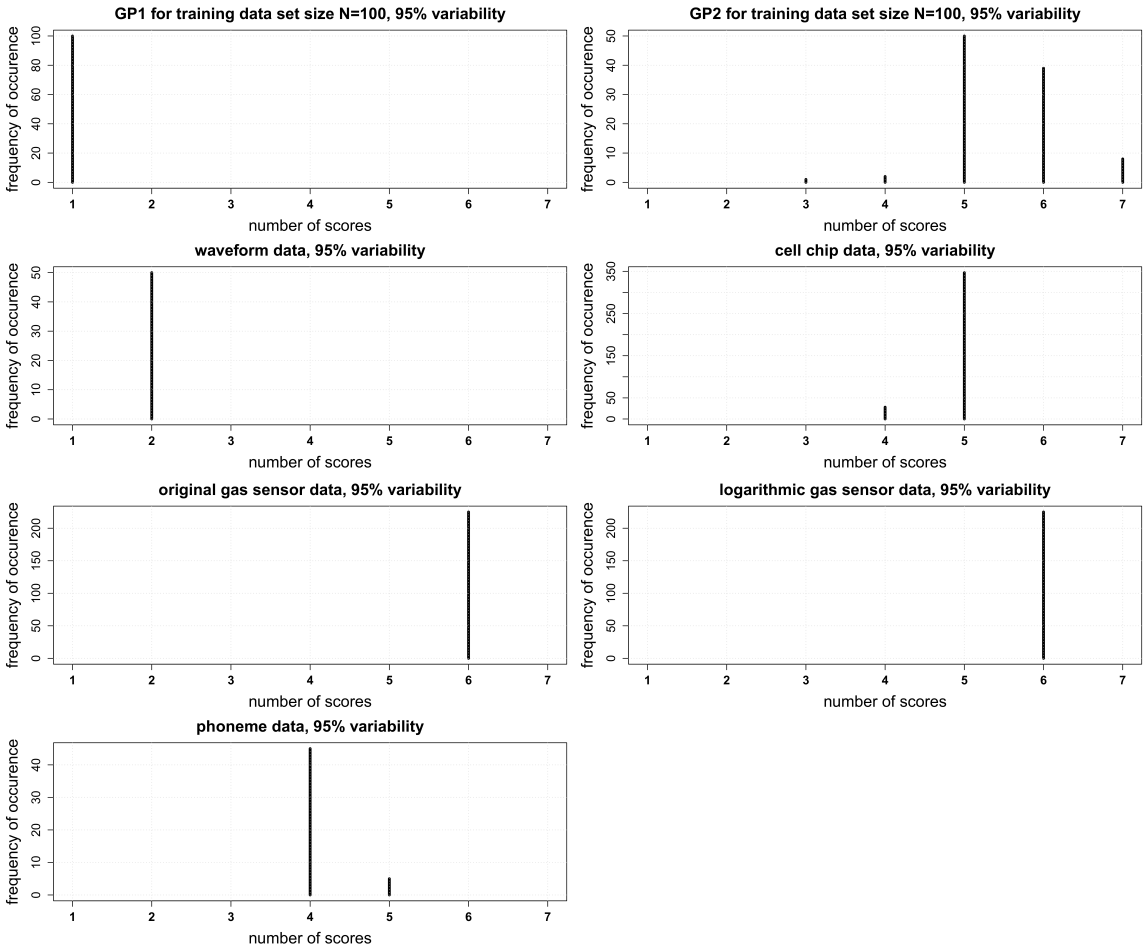


Figure B.1: The frequency of numbers of principal components when explaining $\geq 95\%$ variability in the training data sets, for all examined data sets.

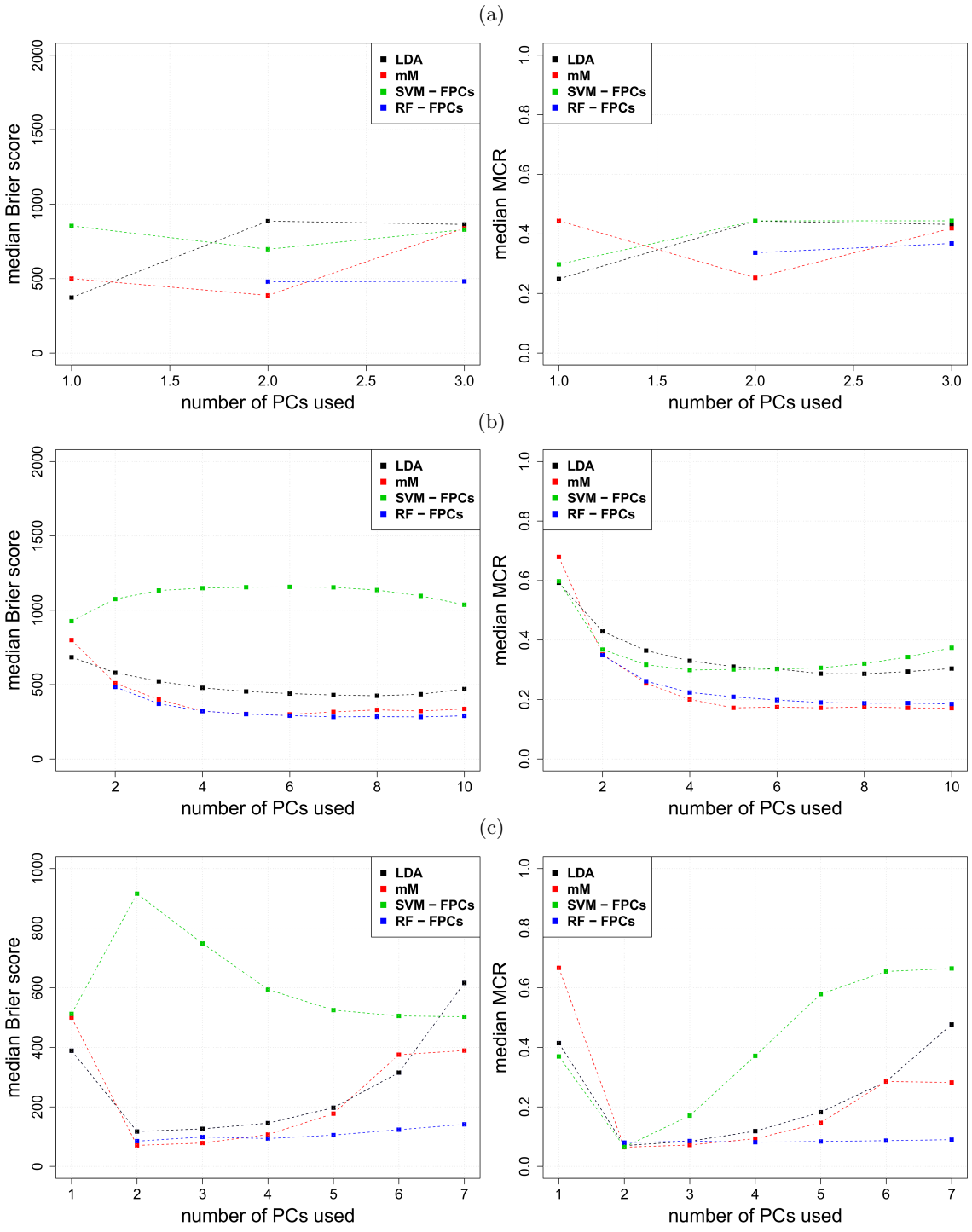


Figure B.2: (a) Median Brier scores and misclassification rates of the validation data set ($N_{val} = 1000$) of the two-class generating process of simulation study A, in dependence of the number of scores used in the respective multivariate methods. (b) The same for the five-class generating process of simulation study A. (c) The same for the waveform data (50 replications, $N_{val} = 250$).

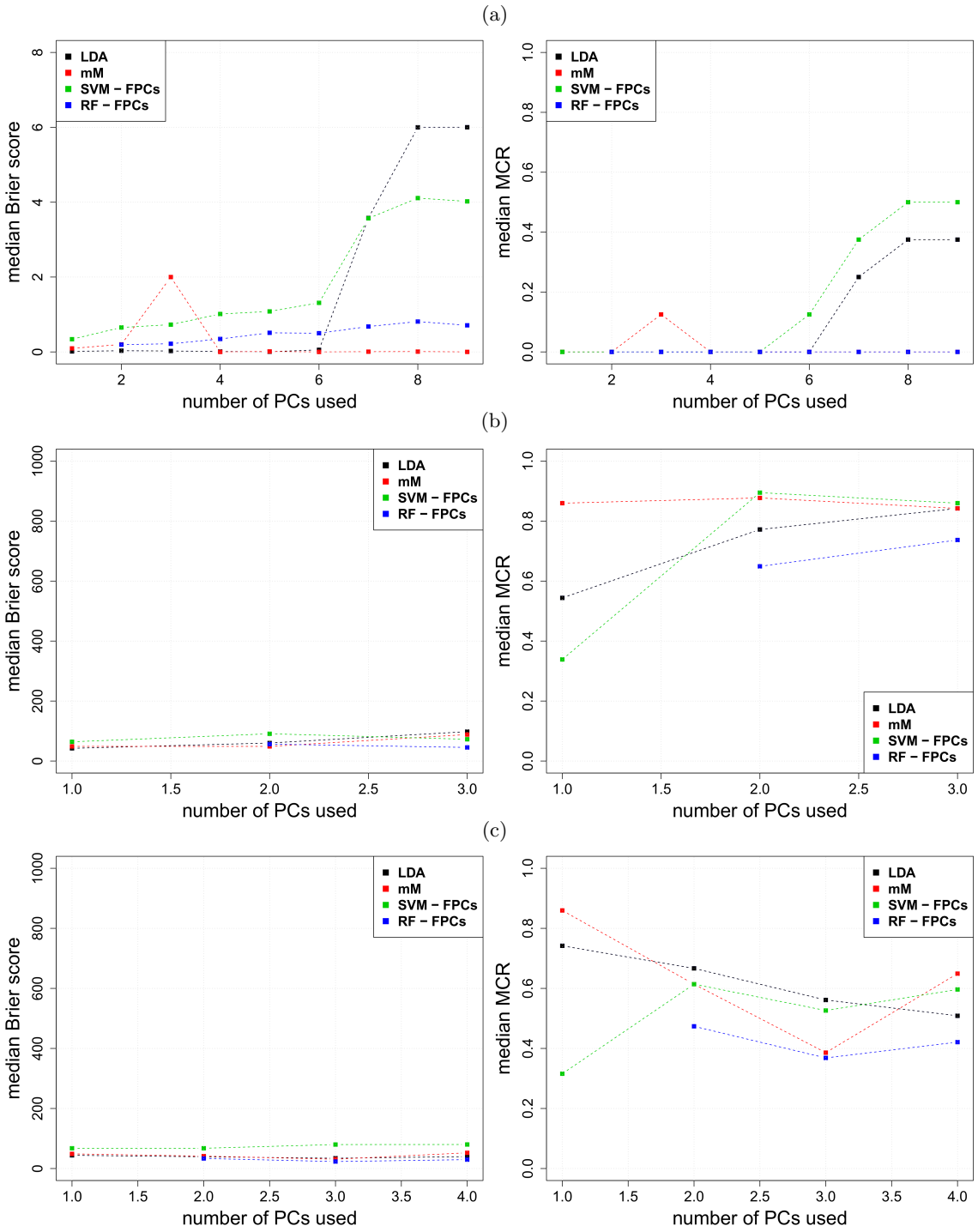


Figure B.3: (a) Median Brier scores and misclassification rates of the validation data sets of the cell chip data, in dependence of the number of scores used in the respective multivariate methods. (b) The same for the original gas sensor data. (c) The same for the logarithmic gas sensor data.

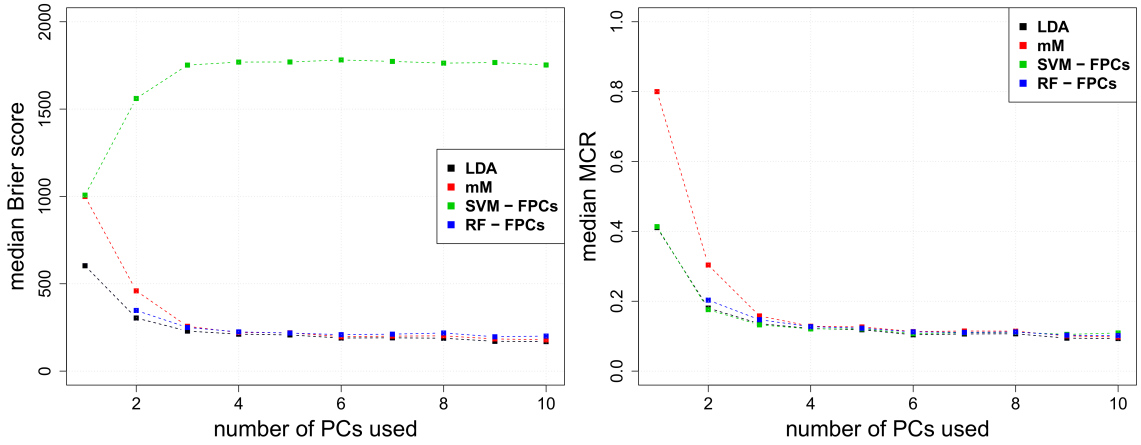


Figure B.4: Median Brier scores and misclassification rates of the validation data sets of the phoneme data, in dependence of the number of scores used in the respective multivariate methods.

B.3 Coefficient IDs and Frequencies of Occurence of Mirroring Coefficients

Table B.1: IDs and frequencies of occurence of mirroring coefficients per generated data set. Those coefficients with the 5 highest frequencies are marked in bold numbers.

Generating process	**IDs of mirroring coefficients**			**Occurence across replications of estimation [%]**		
GP 2	N=100	N=300	N=1000	N=100	N=300	N=1000
	–	–	–	–	–	–
GP 1	5	1	1	1	2	2
	6	2	2	2	1	2
	8	3	3	**65**	2	2
	30	4	4	1	1	2
	33	5	5	1	2	2
	35	**6**	6	1	**4**	2
	37	7	7	2	1	2
	56	**8**	**8**	1	**61**	**60**
	66	18	18	1	2	2
	252	19	19	1	2	2
	279	20	20	1	1	2
	468	21	21	1	3	2
	472	22	22	1	2	2
	491	23	23	1	2	2
	495	24	24	1	1	2
	499	25	25	1	1	2
	503	29	26	1	2	2

Continued on next page

Generating process	IDs of mirroring coefficients			Occurence across replications of estimation [%]		
GP 1	N=100	N=300	N=1000	N=100	N=300	N=1000
	518	30	27	1	1	2
	522	31	28	1	1	1
		32	29		1	2
		33	30		2	2
		34	31		2	2
		35	32		2	2
		36	33		2	2
		37	34		2	2
		38	35		1	2
		45	36		2	2
		47	37		1	2
		48	38		3	2
		49	45		2	2
		50	47		3	2
		51	48		2	2
		53	49		1	2
		56	50		2	2
		57	51		2	2
		66	52		2	2
		233	53		3	2
		234	54		3	2
		235	55		1	1
		236	56		2	2
		237	57		2	2
		238	58		2	2
		239	**66**		2	**4**
		250	233		1	2
		251	234		1	2
		252	235		3	2
		253	236		2	2
		254	237		2	2
		255	238		2	2
		256	239		2	2
		257	250		1	2
		258	251		1	2
		259	252		2	2
		261	253		3	2
		262	254		2	2
		263	255		1	2
		264	256		3	2
		265	257		3	2
		266	258		1	2
		267	259		2	2
		268	261		2	2
		269	262		2	2
		270	263		2	2
		277	264		1	2

Continued on next page

Generating process	IDs of mirroring coefficients			Occurence across replications of estimation [%]		
GP 1	N=100	N=300	N=1000	N=100	N=300	N=1000
		278	265		1	2
		279	266		3	2
		280	267		2	2
		281	268		2	2
		282	269		2	2
		283	270		2	2
		284	277		1	2
		285	278		1	2
		286	279		2	2
		288	280		3	2
		289	281		2	2
		290	282		1	2
		465	283		2	2
		466	284		2	2
		467	285		2	2
		468	286		2	2
		469	288		2	2
		470	289		2	2
		471	290		2	2
		482	465		3	2
		483	466		2	2
		484	467		2	2
		485	468		2	2
		486	469		2	2
		487	470		3	2
		488	471		1	2
		490	482		2	2
		491	483		2	2
		493	484		1	2
		494	485		2	2
		495	486		2	2
		496	487		2	2
		497	488		2	2
		498	489		2	2
		499	490		2	2
		500	491		2	3
		501	493		2	2
		502	494		2	2
		509	495		3	2
		510	496		2	2
		511	497		2	2
		512	498		2	2
		513	499		2	2
		514	500		3	2
		515	501		1	2
		517	502		2	2
		518	509		2	2

Continued on next page

Generating process	IDs of mirroring coefficients			Occurence across replications of estimation [%]		
GP 1	N=100	N=300	N=1000	N=100	N=300	N=1000
		520	510		1	2
		521	511		2	2
		522	512		2	2
			513			2
			514			2
			515			2
			516			2
			517			2
			518			3
			520			2
			521			2
			522			2

Table B.2: IDs and frequencies of occurence of mirroring coefficients per data set. Those coefficients with the 5 highest frequencies are marked in bold numbers.

Data sets	**IDs of mirroring coefficients**	**Occurence across replications of estimation** [%]
Waveform data	–	–
Gas sensor data	–	–
Phoneme data	–	–
Cell chip data	1251	0.80
	1252	0.27
	1258	1.33
	1262	0.27
	1268	**92.27**
	1270	0.27
	1271	0.27
	1278	0.8
	1304	0.27
	1322	0.27
	1323	0.27
	1459	**94.40**
	1466	1.33
	1476	**94.40**
	1486	**94.40**
	1501	**94.40**

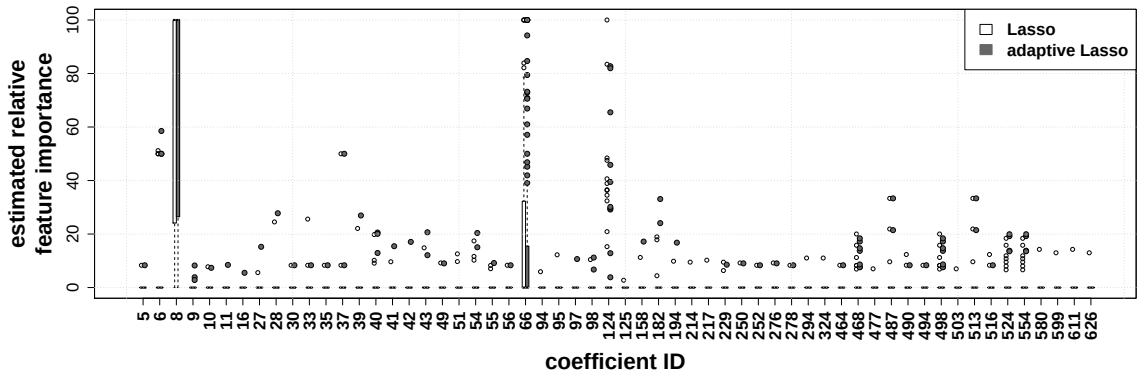
Appendix C

Appendices – Classification of Functional Data by Fitting Penalized and Constrained Multinomial Logit Models

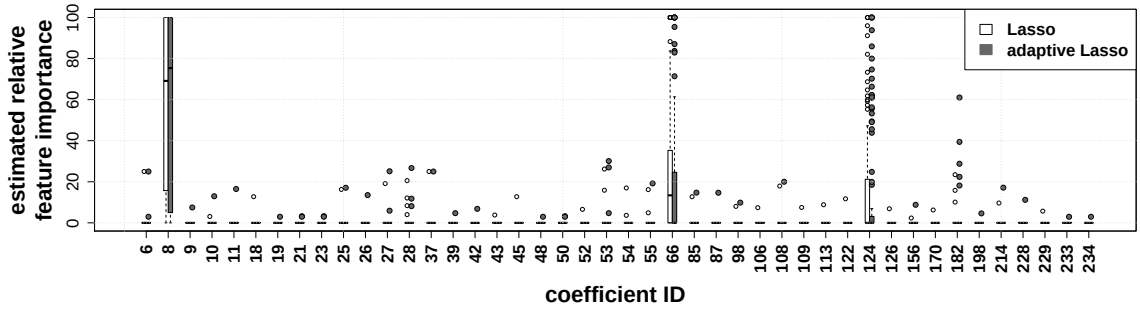
C.1 Estimated Coefficients of the Two-Class Generating Process across Replication Splits

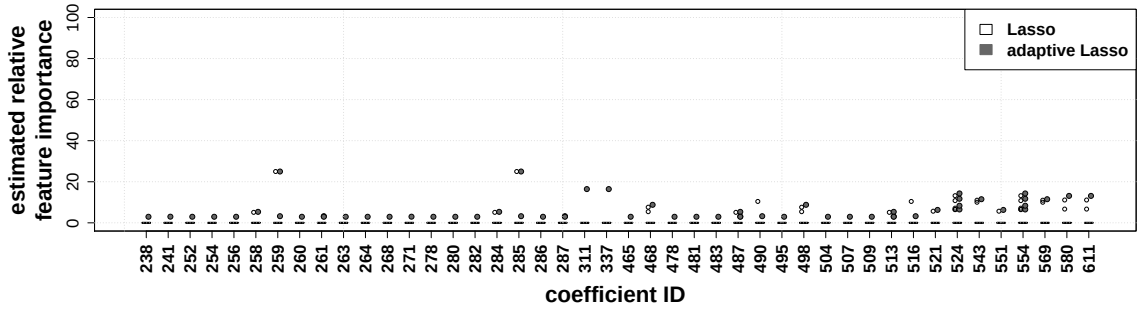
In Chapter 4, the simulation study of Chapter 3 was re-estimated to be able to compare the two proposed estimation approaches for the k-nearest-neighbor ensemble, i.e. the minimization of the Brier score and the penalized cMLM approach. While the prediction results of all competing methods were given in Figure 4.4, the following figure gives the estimated RFI of the coefficients selected by the penalized cMLM for the two-class generation process. All coefficients that are estimated unequal to zero across classes for at least one replicate and penalty version are given as boxplots across the 100 replications. The white boxes show the results of the ordinary, the gray boxes those of the adaptive Lasso penalty. The generated training data sets are (a) of size $n = 100$, (b) of size $n = 300$, and (c) of size $n = 1000$. Depending on the training sample size, up to 53 (Lasso)/ 76 (adaptive Lasso) coefficient IDs across all replications were chosen, and most were selected very seldomly.

(a)

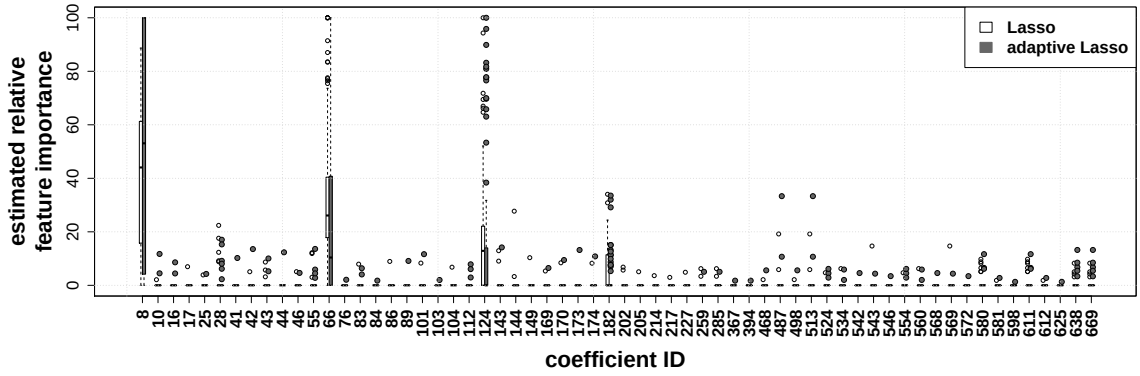

Lasso
adaptive Lasso
estimated relative feature importance
coefficient ID

(b)


Lasso
adaptive Lasso
estimated relative feature importance
coefficient ID

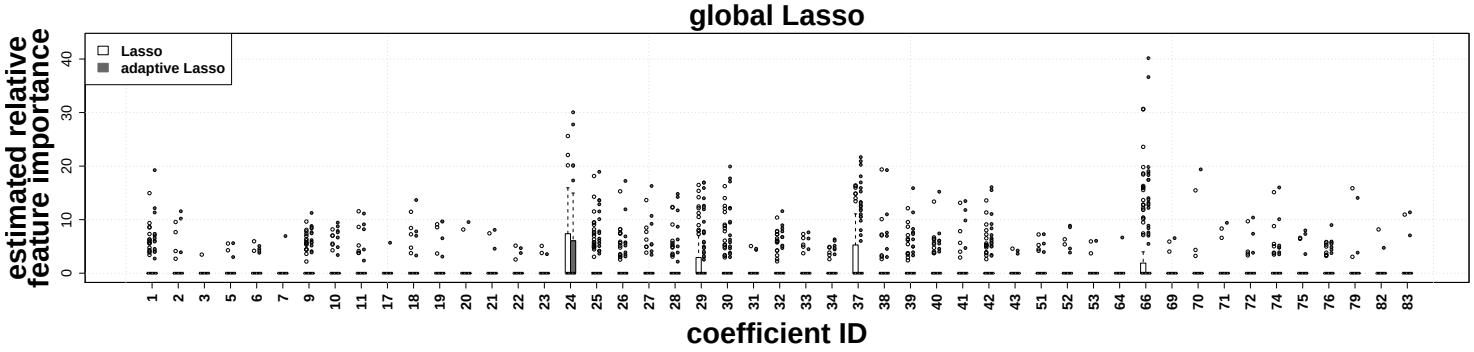

Lasso
adaptive Lasso
estimated relative feature importance
coefficient ID

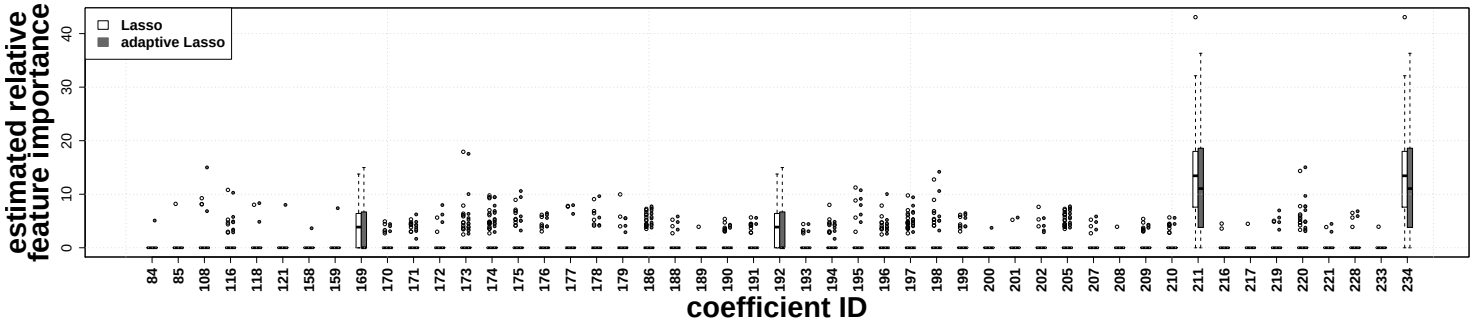
(c)

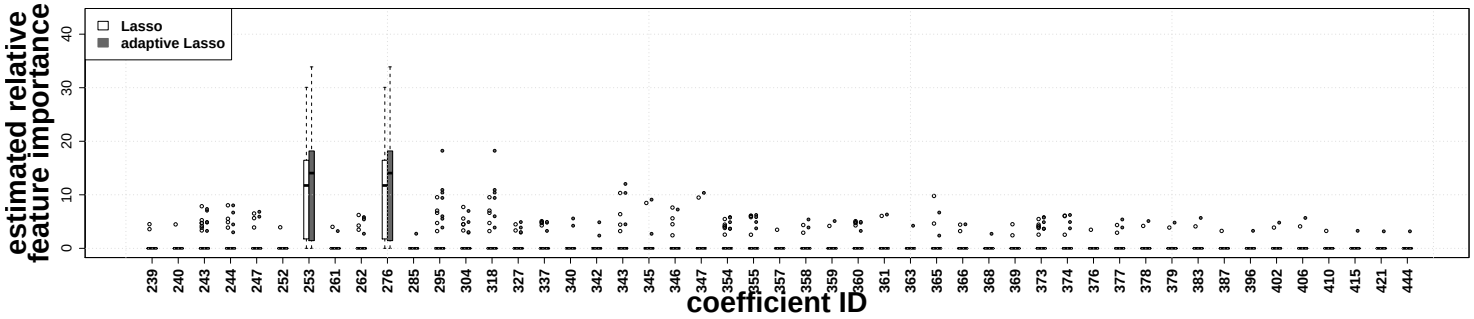

Lasso
adaptive Lasso
estimated relative feature importance
coefficient ID

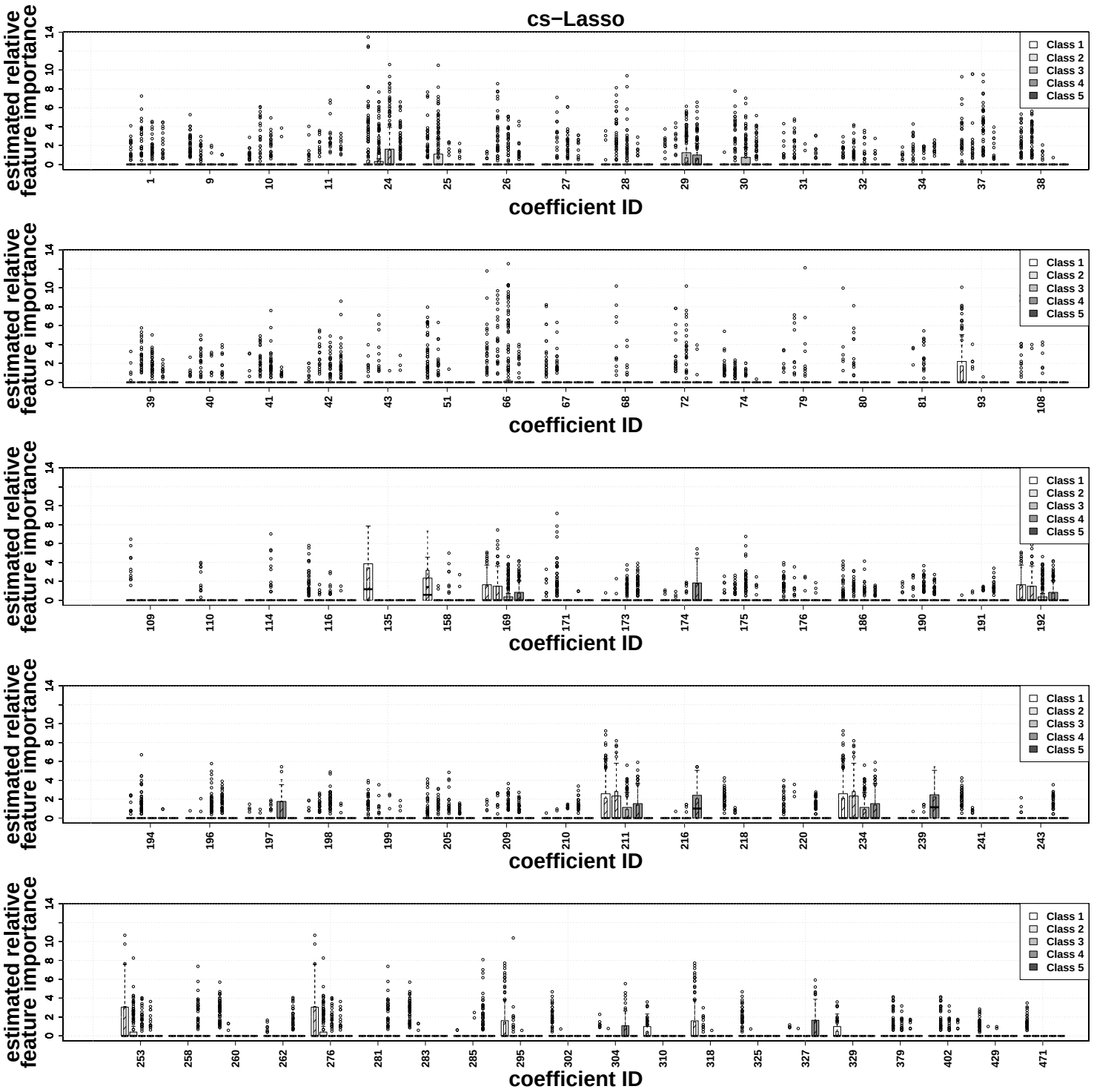
C.2 Estimated Coefficients of the Multi-Class Generating Process across Replication Splits

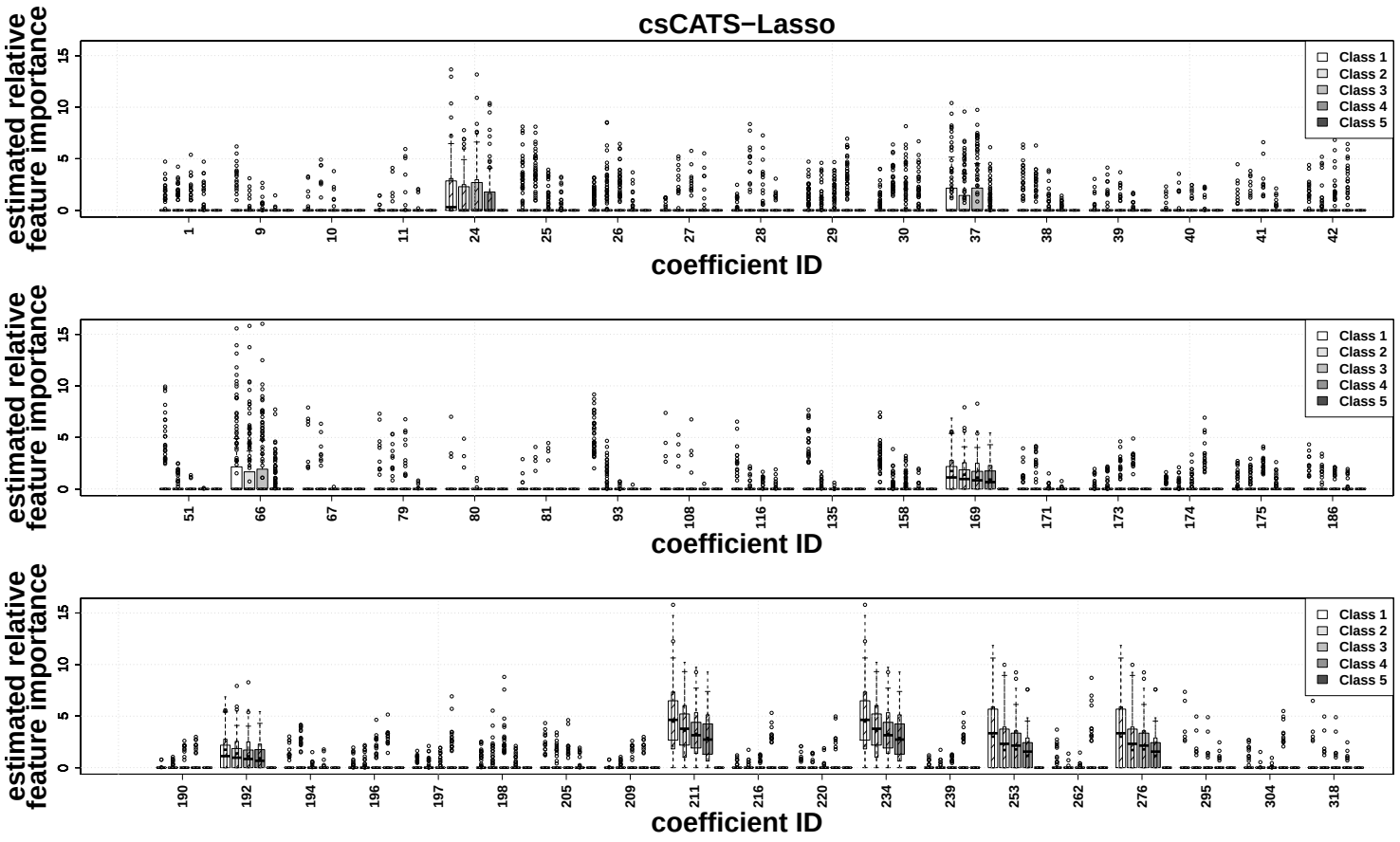
Analogously to the previous Appendix C.1, the following figures give the estimated RFI of the coefficients selected by the penalized cMLM for the multi-class generation process. The coefficients are given as boxplots across all 100 replications. The generated training data sets are of size $n = 300$. Results for $n = 100$/ $n = 1000$ are less/ more pronounced.
The first figure gives the selection results when using the (adaptive) Lasso penalty. Here, the white boxes show the results of the ordinary, the gray boxes those of the adaptive Lasso penalty. All coefficients that are estimated unequal to zero across classes for at least one replicate and penalty version are given. Overall, 130 (Lasso)/ 132 (adaptive Lasso) coefficient IDs across all replications and classes were chosen.
The second figure gives the selection results when using the (adaptive) cs-Lasso penalty. Here, the color coding is with respect to the class. The wider boxes show the results of the ordinary, the superimposed narrow, shaded boxes those of the adaptive cs-Lasso penalty. All coefficients that are estimated to have mean RFI values of above 0.2 (across all splits, for at least one class and penalty version) are given. Overall, 368 (cs-Lasso)/ 349 (adaptive cs-Lasso) from 504 coefficient IDs across all replications and classes were chosen.
The last figure gives the selection results when using the (adaptive) csCATS-Lasso penalty. All coefficients that are estimated to have mean RFI values of above 0.1 (across all splits, for at least one class and penalty version) are given. For this penalty version, overall 115 (csCATS-Lasso)/ 110 (adaptive csCATS-Lasso) coefficient IDs across all replications and classes were chosen.
It can be seen that, for each penalty version, most coefficient IDs were chosen very seldomly.


global Lasso
Lasso
adaptive Lasso
estimated relative feature importance
0 10 20 30 40
1 2 3 5 6 7 9 10 11 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 37 38 39 40 41 42 43 51 52 53 64 66 69 70 71 72 74 75 76 79 82 83
coefficient ID


Lasso
adaptive Lasso
estimated relative feature importance
0 10 20 30 40
84 85 108 116 118 121 158 159 169 170 171 172 173 174 175 176 177 178 179 186 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 205 207 208 209 210 211 216 217 219 220 221 228 233 234
coefficient ID


Lasso
adaptive Lasso
estimated relative feature importance
0 10 20 30 40
239 240 243 244 247 252 253 261 262 276 285 295 304 318 327 337 340 342 343 345 346 347 354 355 357 358 359 360 361 363 365 366 368 369 373 374 376 377 378 379 383 387 396 402 406 410 415 421 444
coefficient ID


cs−Lasso
estimated relative feature importance
coefficient ID
Class 1
Class 2
Class 3
Class 4
Class 5
0 2 4 6 8 10 12 14
1 9 10 11 24 25 26 27 28 29 30 31 32 34 37 38
39 40 41 42 43 51 66 67 68 72 74 79 80 81 93 108
109 110 114 116 135 158 169 171 173 174 175 176 186 190 191 192
194 196 197 198 199 205 209 210 211 216 218 220 234 239 241 243
253 258 260 262 276 281 283 285 295 302 304 310 318 325 327 329 379 402 429 471


csCATS−Lasso
estimated relative feature importance
0
5
10
15
Class 1
Class 2
Class 3
Class 4
Class 5
1
9
10
11
24
25
26
27
28
29
30
37
38
39
40
41
42
coefficient ID
estimated relative feature importance
0
5
10
15
Class 1
Class 2
Class 3
Class 4
Class 5
51
66
67
79
80
81
93
108
116
135
158
169
171
173
174
175
186
coefficient ID
estimated relative feature importance
0
5
10
15
Class 1
Class 2
Class 3
Class 4
Class 5
190
192
194
196
197
198
205
209
211
216
220
234
239
253
262
276
295
304
318
coefficient ID

C.3 Estimated Coefficients of the Cell Chip Data across Replication Splits

In Chapter 4, an important issue was the prediction performance comparison of different methods. To this end, the cell chip data was divided randomly 100 times into learning sets comprising 90 curves and test sets of size 30. While the prediction results of all competing methods were given in Figure 4.7, the following figure C.1 gives the RFI of those penalized cMLM coefficients that are estimated unequal to zero for at least one split and penalty version, as boxplots across all 100 splits. Here, the white boxes show the results of the ordinary, the gray boxes those of the adaptive Lasso penalty. As can be seen, 84 (Lasso)/ 94 (adaptive Lasso) coefficient IDs across all replications were chosen, and most were selected very seldomly.

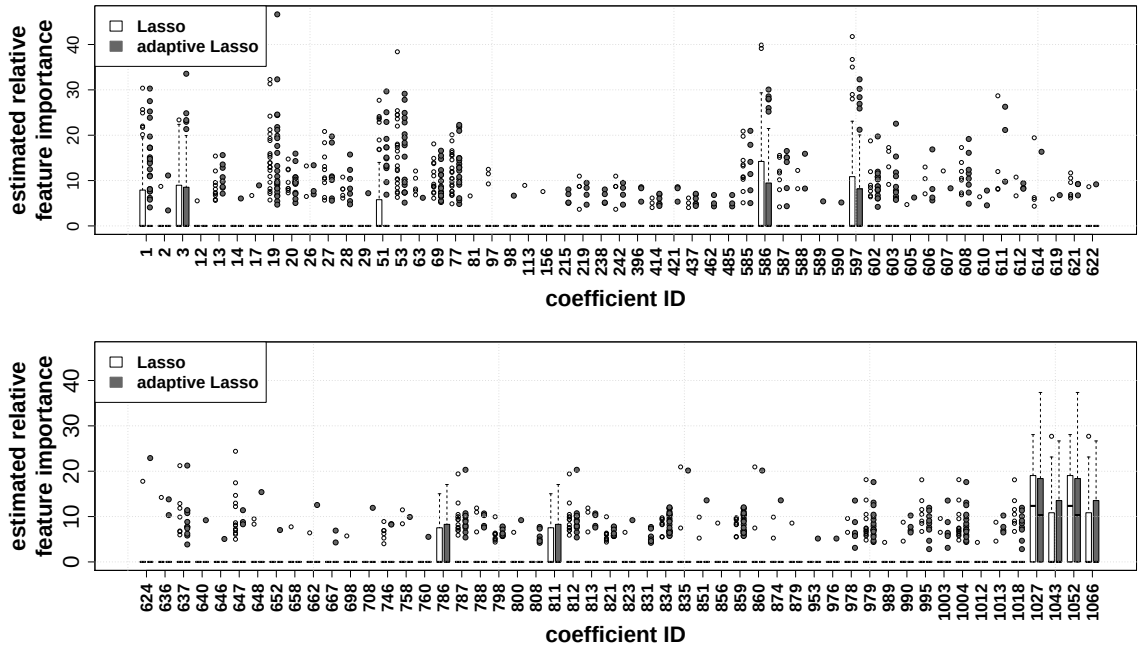


Figure C.1: Coefficients of the cell chip data that are unequal to zero, as boxplots across all 100 splits into learning and test data sets. The white boxes give the results of the ordinary, the gray boxes those of the adaptive Lasso penalty.

C.4 Estimated Coefficients of the Phoneme Data across Replication Splits

Analogously to the cell chip data, modeling and test steps were repeated 100 times to evaluate the prediction performance of the different classification methods. Here, each random draw used 150 curves per class as a learning data set. The test sample contained another 250 randomly drawn curves per class. The prediction results of all competing methods were given in Figure 4.11. The following figure gives the RFI of those globally penalized cMLM coefficients that are estimated to have mean RFI values of above 0.25 (across all splits, for at least one penalty version), as boxplots across all 100 splits. Here, the white boxes show the results of the ordinary, the gray boxes those of the adaptive Lasso penalty. Overall, 487 (Lasso)/ 529 (adaptive Lasso) from 800 coefficient IDs across all replications were chosen, most very seldomly, as, for example, the coefficients 7 - 10 exemplify.

estimated relative feature importance
0
5
10
15
20
25
Lasso
adaptive Lasso
1
2
3
7
8
9
10
17
19
21
26
27
28
30
31
34
37
38
39
41
43
45
coefficient ID

estimated relative feature importance
0
5
10
15
20
25
Lasso
adaptive Lasso
46
47
48
49
50
52
53
54
57
59
60
62
63
64
66
68
69
70
71
72
76
77
coefficient ID

estimated relative feature importance
0
5
10
15
20
25
Lasso
adaptive Lasso
78
79
84
86
87
89
92
97
98
105
106
107
111
112
113
114
116
117
118
121
125
126
coefficient ID

estimated relative feature importance
0
5
10
15
20
25
Lasso
adaptive Lasso
129
130
137
138
143
145
146
147
152
173
174
175
179
181
182
183
184
186
189
193
199
202
coefficient ID

estimated relative feature importance
0
5
10
20
Lasso
adaptive Lasso
205
206
210
211
213
214
215
220
223
224
236
241
242
243
247
248
250
252
254
coefficient ID

C.5 Estimated Coefficients of the Phoneme Data across Replication Splits, Using a Category-Specific Penalty

Analogously to the cell chip data, modeling and test steps were repeated 100 times to evaluate the prediction performance of the different classification methods. Here, each random draw used 150 curves per class as a learning data set. The test sample contained another 250 randomly drawn curves per class. The prediction results of all competing methods were given in Figure 4.11. The following figure gives the RFI of those category-specific penalized cMLM coefficients that are estimated to have mean RFI values of above 0.25 (across all splits, for at least one class and penalty version), as boxplots across all 100 splits. Here, the color coding is with respect to the class. The wider boxes show the results of the ordinary, the superimposed narrow, shaded boxes those of the adaptive cs-Lasso penalty. Overall, 689 (cs-Lasso)/ 717 (adaptive cs-Lasso) from 800 coefficient IDs across all replications were chosen, most very seldomly, as, for example, the coefficients 18 or 31 exemplify.

estimated relative feature importance
0
2
4
6
8
10
Class 1
Class 2
Class 3
Class 4
Class 5
1
2
18
20
31
37
38
39
52
coefficient ID

estimated relative feature importance
0
2
4
6
8
10
Class 1
Class 2
Class 3
Class 4
Class 5
69
70
88
105
106
116
118
121
137
coefficient ID

estimated relative feature importance
0
2
4
6
8
10
Class 1
Class 2
Class 3
Class 4
Class 5
138
143
146
152
173
179
184
186
189
coefficient ID

estimated relative feature importance
0
2
4
6
8
10
Class 1
Class 2
Class 3
Class 4
Class 5
205
206
207
208
209
211
212
214
220
coefficient ID

estimated relative feature importance
0
2
4
6
8
10
Class 1
Class 2
Class 3
Class 4
Class 5
224
225
236
241
243
245
247
250
252
256
257
270
coefficient ID

C.6 Estimated Coefficients of the Phoneme Data across Replication Splits, Using a Category-Specific CATS Penalty

Analogously to the cell chip data, modeling and test steps were repeated 100 times to evaluate the prediction performance of the different classification methods. Here, each random draw used 150 curves per class as a learning data set. The test sample contained another 250 randomly drawn curves per class. The prediction results of all competing methods were given in Figure 4.11. The following figure gives the RFI of those categorically structured (CATS) penalized cMLM coefficients that are estimated to have mean RFI values of above 0.25 (across all splits, for at least one class and penalty version), as boxplots across all 100 splits. Here, the color coding is with respect to the class. The wider boxes show the results of the ordinary, the superimposed narrow, shaded boxes those of the adaptive cs-Lasso penalty. Overall, only 315 (csCATS-Lasso)/ 340 (adaptive csCATS-Lasso) from 800 coefficient IDs across all replications were chosen, most very seldomly, as, for example, the coefficients 20 or 146 exemplify.

estimated relative feature importance
0
2
4
6
8
1
2
20
37
38
39
69
coefficient ID
Class 1
Class 2
Class 3
Class 4
Class 5

estimated relative feature importance
0
2
4
6
8
70
105
106
116
118
137
138
coefficient ID
Class 1
Class 2
Class 3
Class 4
Class 5

estimated relative feature importance
0
2
4
6
8
143
146
152
173
179
184
186
coefficient ID
Class 1
Class 2
Class 3
Class 4
Class 5

estimated relative feature importance
0
2
4
6
8
205
206
211
214
224
225
241
247
coefficient ID
Class 1
Class 2
Class 3
Class 4
Class 5

Appendix D

Motivating Data Sets – Details

In the following, more details on the data sets examined in this thesis are given, especially on their technical background and data acquisition. Also, the most common evaluation methods are outlined where appropriate.
Many thanks to Ulrich Bohrn, Christian Guijarro and Evamaria Stütz for providing parts of the cell chip data.
Many thanks to Remigiusz Pastusiak for providing the spectroscopic data. I would also like to thank him and Artur Pastusiak for the support in designing the gas measurement chamber.

D.1 Cell Chip Data

D.1.1 Materials

Cell Culture
For the data, Chinese hamsterlung fibroblast cells are used. They were purchased from DSMZ (German Collection of Microorganisms and Cell Lines, Braunschweig, Germany, ACC-No. 335).

Chemicals
Paracetamol (Acetaminophen 99%, short: AAP, catalog #7085) was purchased from Sigma-Aldrich (Seelze, Germany), Triton-X (TX, catalog #93418) from Fluka (Steinheim, Germany), Hepes (catalog #17-737F), streptomycin and penicillin from BioWhittaker (Heidelberg, Germany), phosphate buffered saline (PBS, catalog #L1820), fetal bovine serum (FBS, catalog #S3113) and Dulbecco's modified eagle medium (DMEM, catalog #F0455) from Biochrom (Berlin, Germany).

Micronas Cell Chips
Eight sensors are distributed on the chip surface of the Micronas metabolic chip SC1000

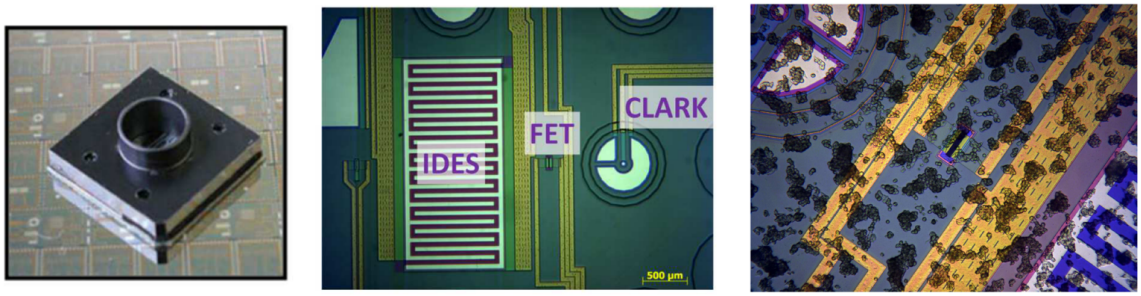


Figure D.1: A metabolic cell chip SC1000 from the Micronas GmbH in its housing (left). Chip surface with the three sensor types (middle). Chip surface covered with cells (right).

used in our study. Five ion sensitive field effect transistors (ISFET), one interdigitated electrode structure (IDES) and two oxygen sensitive electrodes based on CLARK-electrodes, i.e. amperometric transduction sensors, are used in each experiment (Thévenot et al., 2001; Ramamoorthy et al., 2003).
The metabolism of the cells involves the production and excretion of acid waste products like lactate and CO_2, which are the results of oxidative and non-oxidative cellular pathways. ISFET-signals relate to the acidification rate of the medium in which the cells are contained. The acidification rate alters due to the cells' excretion of acidic metabolites. At the same time, the cells' mitochondrial activity is reflected by their oxygen consumption (see e.g. Wolf et al., 1998). CLARK-electrodes measure the oxygen (O_2) contained in the medium via amperometric transduction sensors, with the O_2 content being a proxy for the respiration activity of the cells (Thedinga et al., 2007; Ceriotti et al., 2007). IDES-signals measure the cellular impedance (the cell membran is an electrical insulator, cf. Ehret et al., 1998) and can be used to draw conclusions about the cell morphology and cell adhesion on the surface of the sensor chip. Figure D.1 shows a cell chip in its housing (black cavity and socket) and its surface with the three sensor types.

The Bionas 2500 Analyzing System
The Bionas 2500 Analyzing System provides six so-called bio-modules, each serving as a test point for one chip. A fluidic head connects the chip with a medium reservoir via polyether ether ketone tubes. The Bionas system provides an automated perfusion system for ensuring constant medium supply and the simultaneous drain of consumed medium. Six devices *à* 6 reservoirs can be used for storing test substances, the so-called autosampler (cf. Figure D.2). Apart from recording, the software of the Bionas system also allows for preprocessing the data, see also Section D.1.2.

Software
The operation system Windows XP and the software included by the Bionas 2500 Analy-

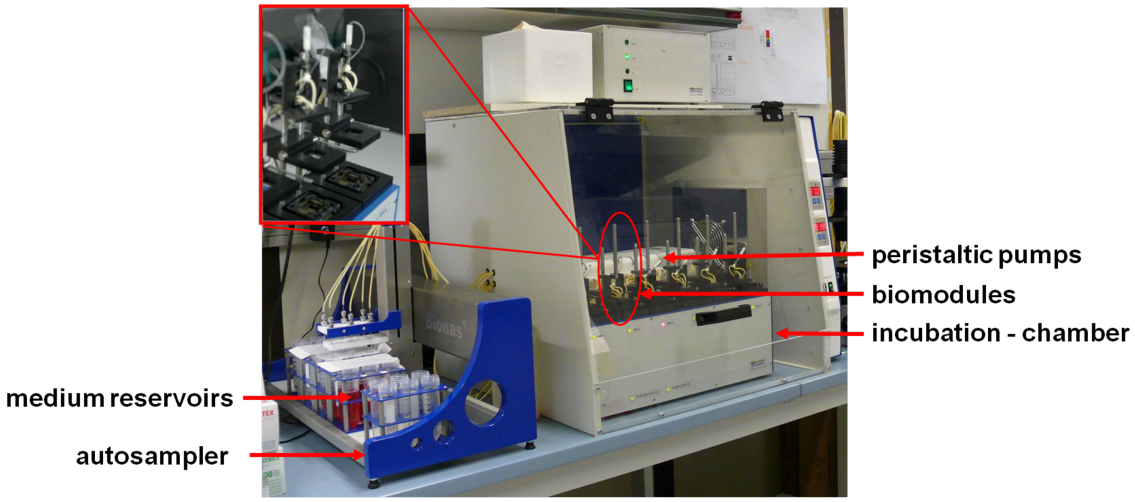


Figure D.2: The Bionas system.

zing System (Bionas 2500 Data Analyzer V 1.52) are used.

D.1.2 Data Acquisition

Measurement Cycles
All measurements follow the same protocol:

I) The cells are grown at 37°C in a humidified incubation chamber in a 5% CO_2 atmosphere while cultivated in DMEM. The latter is supplemented with 10% heat-inactivated FBS, 100 units/ml penicillin, and 100 μg/ml streptomycin. In the following, this mixture is called "nutrient medium". The cells are left to adhere and to proliferate in the incubation chamber.

II) When confluency is reached, the cells are detached using 0.05% trypsin (w/v) and 0.02% ethylenediaminetetraacetic acid (w/v) in PBS and are redistributed in a new cell culture flask (according to standard procedures, described for example in Brent et al., 1988).

III) Before each measurement, plain chips are cleaned with 70% ethanol for 10 minutes and flushed with PBS. About 175,000 cells (confirmed by microscopy and manual counting) are seeded on the chip in 400 μl nutrient medium and incubated overnight at 37°C in humidified atmosphere with 5% CO_2. The next day, confluency is verified by microscopy.

IV) Before each measurement, the Bionas 2500 Analyzing System and its test points (biomodules) are prepared as follows:

70% ethanol is pumped through the system for 3 minutes (pump rate: 100%) and 10 minutes (pump rate: 4%) to dissolve possible deposits and to disinfect the system. Afterwards, PBS is pumped through the system for 2 minutes (pump rate: 100%) to wash away possible residuals of ethanol and acidic waste products. To clean the system from the PBS in the last preparation step, running medium (weakly buffered DMEM with 1mM Hepes, supplemented with 1% FBS and penicillin/ streptomycin) is pumped through the system for 2 minutes (pump rate: 100%). The system is warmed up to 37°C.

V) The chips are placed in the biomodules and overlaid with the fluidic head. The medium reservoirs containing the testing substances are placed into the respective devices of the automated perfusion system, and the measurement cycle is started. All phases are executed in the three minute stop-go mode. This means that the test substance is pumped over the cells for three minutes, followed by three minutes with paused pumps, and so on. Only in the stop phase the consumption of oxygen and the acidification of the extracellular medium can be observed and the obtained chip signals increase. In the subsequent go-phase, fresh medium with a normal oxygen content and a defined pH is pumped on the cells, exchanging the consumed medium and setting the chip signals back to a baseline level. In the next stop phase, the signals of the CLARK-like electrodes and the ISFETs increase again due to the cells' consumption of oxygen and the excretion of acidic metabolites in the extracellular medium. The flow rate was set to 56 ml/min. The following phases are performed:

Phase 1: To equilibrate the cells to the new environment and the medium flow, only running medium is pumped over the cells for three hours, and the signals stabilize.

Phase 2: For the test phase, AAP of different concentrations, dissolved in running medium, is pumped over the cells for three to twelve hours. Depending on the concentration, the cells react to the AAP, i.e. the signals alter.

Phase 3: In order to obtain a base line without cellular activity, the system is flushed with a 0.2% TX-solution. This devitalizes the cells and dissolves them from the chip surface. The signals collapse.

VI) The chips are cleaned from residuals, checked for signs of dysfunction, and stored in 70% ethanol, until preparing them for the next measurement.

Design of Experiments
Apart from the reaction of the cells the cell chip signals are potentially influenced by a number of other factors, which in part can be included in the design of the experiment. The aim was to perform measurements following a randomized, full-factorial design including the following controllable factors:

- measurement position, i.e. bio-module

- chip number
- AAP concentration.

Uncontrollable factors are, amongst others,

- the composition of the individually prepared running medium, especially the pH-values altering with age,
- natural slight drift in cell culture: too many cell divisions may result in slightly altered performance of a cell population in an experiment
- the actual confluency rate of a single chip per measurement: its verification by microscopy is prone to user subjectivity, and
- alterations of the read-out software in-line with system updates.

Since one measurement takes at least three days including chip and system preparations, a randomized, full-factorial design was not feasible for all planned AAP concentrations. Also, the life cycle of the chips is limited due to chemical and physical influences. Experiments following a randomized, full-factorial design are performed for 10 chips with AAP concentrations 0.5mM and 2.5mM, and for 10 chips with AAP concentrations 0mM and 5mM. A design including 10 chips and AAP concentrations 0mM, 1.5mM, 3.5mM, and 6mM could not be completed due to the above mentioned reasons.
All in all, $N = 280$ measurements per signal type of usable quality are recorded, see also Figure D.3.

Data Preprocessing
The data shown in Figure D.3 is preprocessed by the Bionas 2500 Analyzing System. Here, each data point is calculated from the gradient of the linear fit of the first and last test point of the stop-phase in the three minute stop-go mode. Thus every six minutes a test point is generated, representing the cellular metabolic activity. This is due to the assumption that the cells have maximally reacted to the test substance at the end of the stop-phase, before fresh medium and test substance is again flushed over them in the go-phase. The linear fit is an approximation to the measured signal representing the reaction of the cells, whose progression depends on the test substance.
At the end of a measurement cycle, the data point at which the test substance reaches the cells is defined. Just before the test substance is applied, one expects the cells to exhibit maximal viability, i.e. 100%. All signals are standardized in such a way that at the respective data point (at about 215 minutes), the signals have a value of 100.

D.1.3 Evaluation Techniques

There are no specialized evaluation tools for in-vivo measurements of cell viability (see also Lagarde and Jaffrezie-Renault, 2011; Eltzov and Marks, 2011, for a review of cell

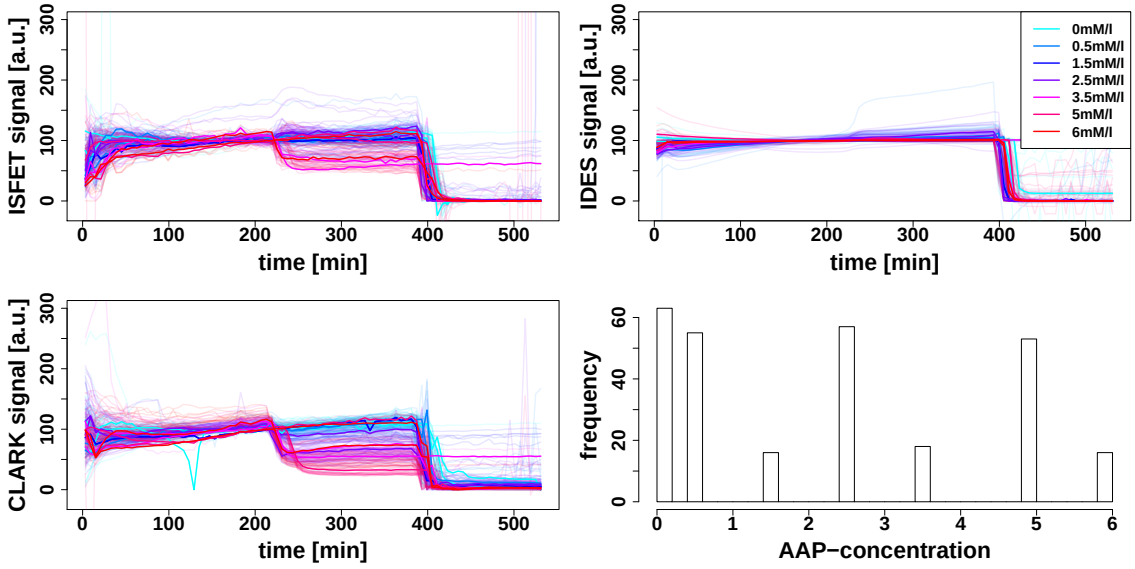


Figure D.3: First three panels: all available cell chip data, after outlier correction and elimination of erroneous measurements, per signal type. Lower right panel: respective histogram of the AAP concentrations [mM].

based biosensors). Common evaluation techniques include the interpretation of the signal progression, both mutual to each signal type as well as to the signal before and after test substances have been applied (see e.g. Bohrn et al., 2012). Also quite widespread is the comparison of the mean values ± the standard deviation of signals at certain data points or per test substance concentration or other key measures in order to assess the impact of the test substance on the respective viability parameter (see e.g. Huang and Pang, 2012). Rarely, multivariate methods as for example principal component analysis are applied (see e.g. Lovelady et al., 2007).
To our knowledge, the present thesis is the first that has examined cell chip data by means of functional data analysis.

D.2 Gas Sensor Data

D.2.1 Materials

Gas Sensors
Four AS-MLV metal oxid gas sensors with a tin dioxid based sensitive layer are used, which are specialized for volatile organic compounds. They were bought from AppliedSensor GmbH (Reutlingen, Germany). The sensitive layer was deposited on a miniaturized hotplate (heater).

It is known that reactions between a sensitive layer and the atmosphere are, among other things, depending on the composition of the ambient air and the type of sensitive material as well as on the layers' temperature (see e.g. Lee and Reedy, 1999). Thus, applying a certain gas to the gas sensor at four different temperatures simulates four different sensitive layers.

Gases
The in-house gases nitrogen (N_2), nitrogen oxid (NO_2), carbon monoxid (CO) and oxygen (O_2) are used, as well as gas containers with ethanol (EtOH, 5200 parts per million (ppm) in N_2), pentanal (Pent, 320ppm in N_2), acetaldehyde (Acetal, 560ppm in NO_2) and acetone (Acet, 1000ppm in NO_2).

Gas Measurement Test Stand
The gas measurement test rig at Siemens CT laboratories consists of eight mass flow controllers (MFC) of various flow rates and provides four gas container inputs as well as seperate in-house pipe inputs for nitrogen, nitrogen oxid, carbon monoxid, and oxygen. The system includes a control unit to operate the MFCs and to set up measurement cycles.
For the single gases used in this study, the following MFCs are employed: Pent - 300 standard cubic centimeters per minute (sccm) MFC, EtOH - 50sccm MFC, Acetal - 17sccm MFC, Acet - 20sccm MFC.

Gas Measurement Chamber
To be able to run a simultaneous measurement with all four gas sensors, a cylindric measurement chamber was constructed, where each two sensors are positioned face to face. Head and tail of the cylinder provide gas inlet and outlet. The measurement chamber, cf. the computer-aided design in Figure D.4, was lathed from teflon, which is not prone to gas species inclusion, is chemically inert and can be heated out.

Software
The board controlling the gas sensors is a MSP430 F2619 board from Texas Instruments (Freising, Germany).
The data is recorded and stored by the multi-session terminal emulation application FoxTerm 1.4.3.0.

D.2.2 Data Acquisition

The four tin dioxide gas sensors are placed in the gas measurement chamber. Each sensor is connected to a control board, which in turn is connected to a computer via USB. The gas in- and outlets of the gas measurement chamber are connected to the respective gas sources of the measurement test stand.

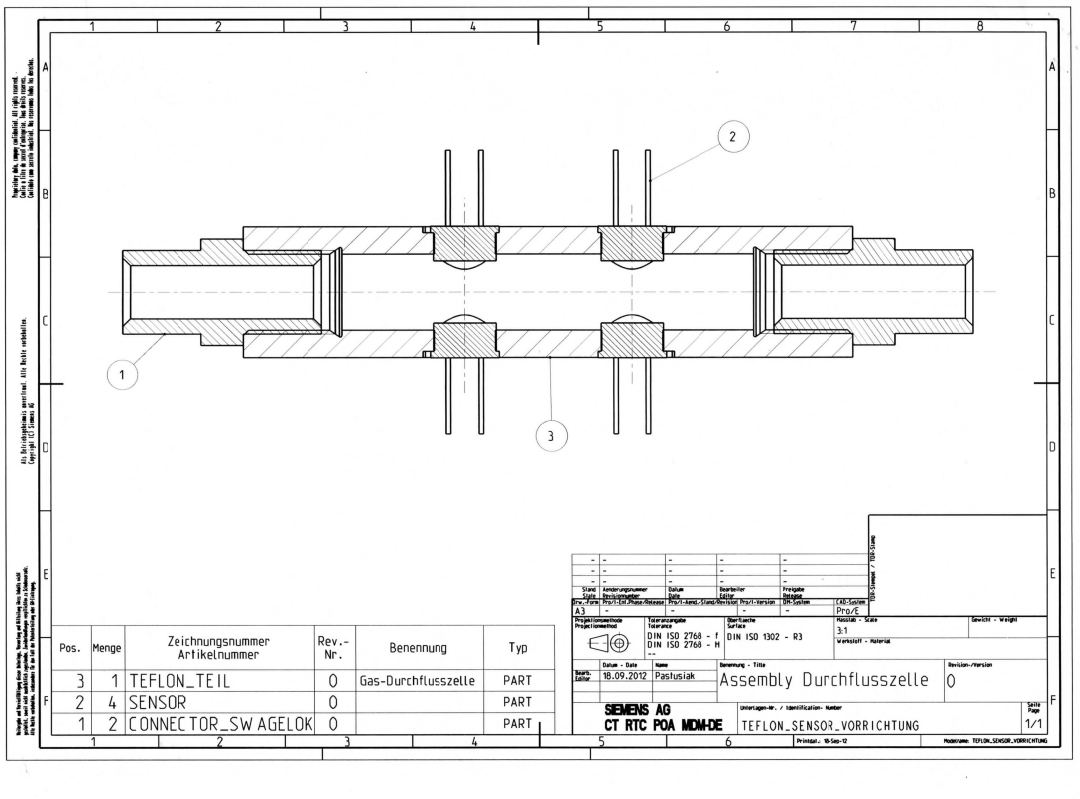


Figure D.4: Computer-aided design scheme of the constructed gas measurement chamber.

The gas sensor control boards applied a five-step temperature cycle to the hotplates to heat up the sensitive layers. The temperatures are indirectly defined by the applied voltages, namely 0V, 3.9V, 3.1V, 3.3V, and 3.5V. The first voltage represents the cycle-initialization. The second voltage defines the bake-out step that expurgates the sensitive layer from residual gas molecules. The last three voltages represent different temperatures, i.e. sensitive layers. The gas flow, also shown in Figure D.5, is such that the sensors are exposed to the gas species sequentially, and in turn with synthetic air (SA). The SA is composed of 25 Vol.-% (i.e. 40% relative) humidity (added to the gas flow by a bubbler), 60 Vol.-% nitrogen and 20 Vol.-% oxygen. The concentrations of the single gas species can be found in Table D.1.
The data read-out from the gas sensors is performed every five seconds, each read-out is providing a gas sensor signal of 89 equidistant data points including a whole temperature cycle. A plot of all measurements can be found in Figure D.6, where the initialization step of 0V (data points 1-13) has been omitted for clarity. The signal steps resulting from temperature changes can be seen clearly.

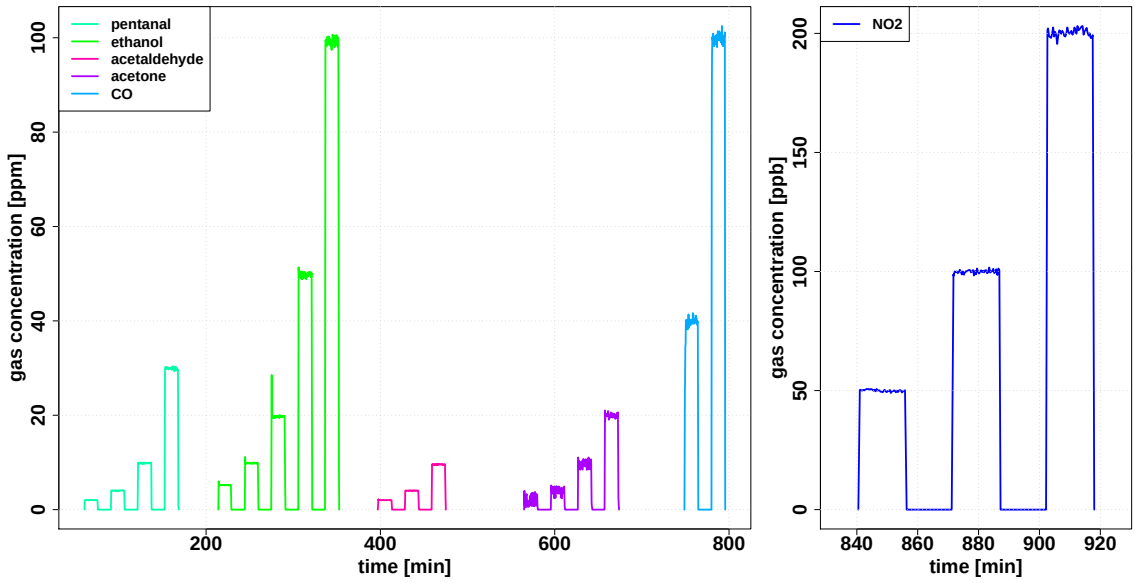


Figure D.5: Gasflow applied to the gas sensors. Pure synthetic air is applied where no other gases are shown.

D.2.3 Evaluation Techniques

The chemical sensor community uses and studies a broad spectrum of sensor evaluation techniques, ranging from explanatory tools based on physics to multivariate and functional approaches, often closely related to machine learning techniques. An exhaustive survey would fill a whole book. Thus, only some exemplary references are given here, as the review of Gutierrez-Osuna et al. (2011) and some examples of evaluation approaches like Bandyopadhyay et al. (2011); Carlo et al. (2011); Pashami et al. (2013); Gosangi and Gutierrez-Osuna (2014).

Gas Species	Concentrations	Unit
Pentanal	2, 4, 10, 30	[ppm]
Ethanol	5, 10, 20, 50 100	[ppm]
Acetaldehyde	2, 4, 10	[ppm]
Aceton	2, 4, 10, 20	[ppm]
CO	40, 100	[ppm]
NO_2	50, 100, 200	[ppb]

Table D.1: Gas species with respective concentrations applied to the metal oxid gas sensors.

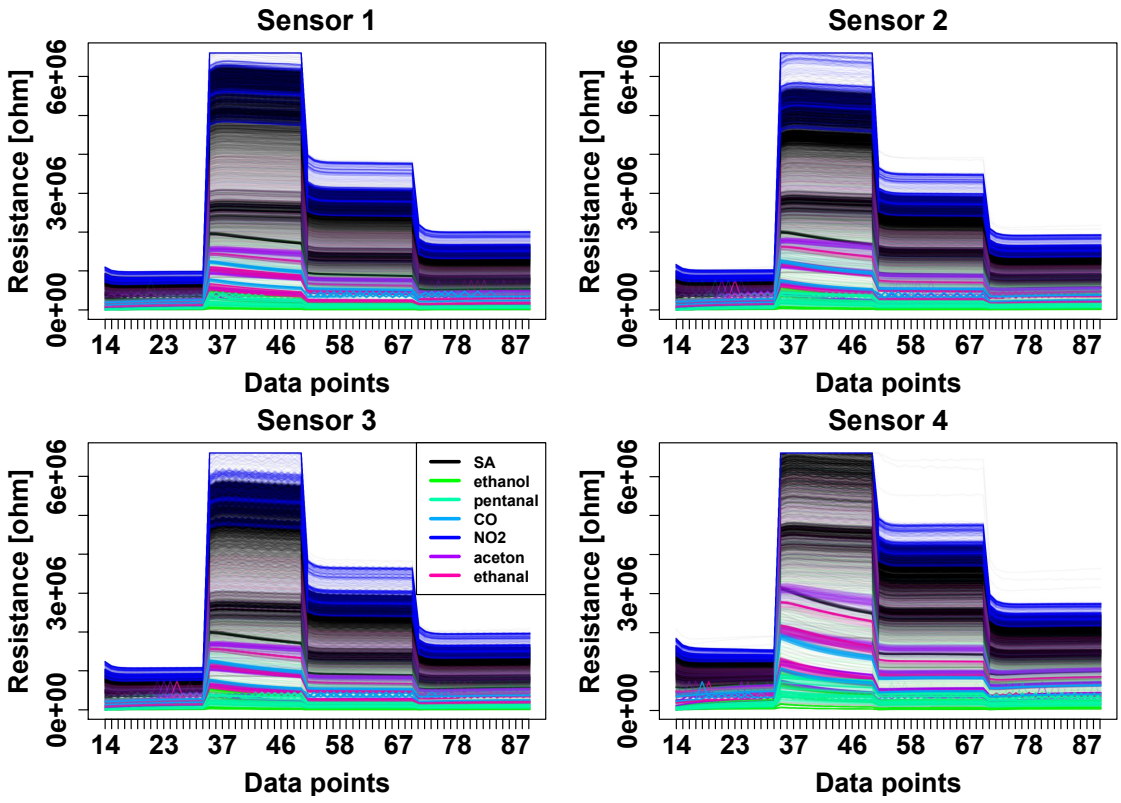


Figure D.6: Gas measurements for all gas species and concentration per sensor. The initialization step of 0V (data points 1-13) has been omitted for clarity.

D.3 Spectroscopic Data

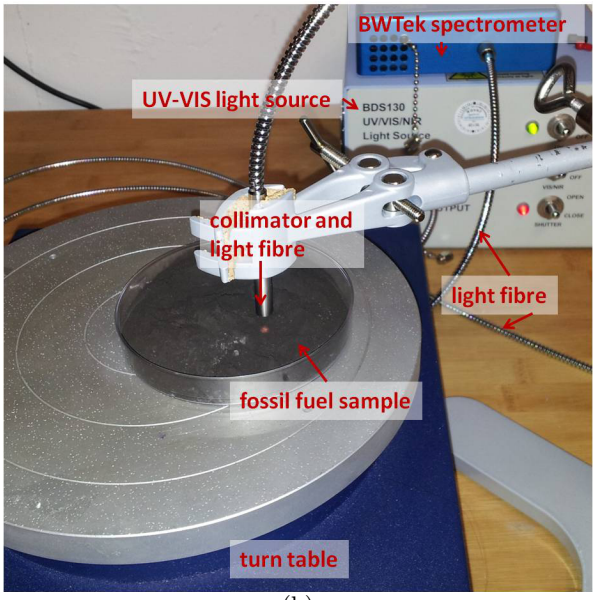
D.3.1 Materials

There are two data sets containing spectra of fossil fuels. They are recorded by two spectrometers which operate on different spectral domains. The first spectrometer, the MPA (from Bruker, Ettlingen, Germany), is a near infrared (IR) spectrometer providing a measurement range between 800nm and 2780nm. The second set of data is measured with a BWTek Compass X (BTC 112E, from Polytec, Waldbronn, Germany), measuring in the ultraviolet-visible (UV-VIS) range between 250nm and 880nm.
For the measurement setup, the following additional components are employed: One turn table and one light source (Polytec BDS 130) (both from Polytec, Waldbronn, Germany), one fiber collimator (F240SMA-A, from Thorlabs GmbH, New Jersey, USA), a spectralon calibrated diffuse reflectance standard (5%, from Labsphere Inc., North Sutton, USA), and one 9-to-1-channel light fiber (from Hellma GmbH & Co. KG, Müllheim, Germany).

D.3.2 Data Acquisition

The measurement setup differed for the two spectrometers. Carrying out the UV-VIS measurements, the sample is placed on the turn table. The collimator is arranged perpendicularly to the smoothed sample surface. The 9-channel connector of the light fiber is connected to the light source, the 1-channel connector to the spectrometer. The fiber channel combining both is plugged in the collimator (see also Figure D.7(a)). The whole setup is placed in a container to exclude ambient light from the measurements. The turn table simulated a moving sample. The NIR measurements are performed using the integrating sphere, probe and halogen light source implemented in the MPA. The sample is put on a protective glass, which in turn is placed on the integrating sphere (see also Figure D.7(b)). To simulate a moving sample, the protective glass under the fossil fuel sample is moved between measurements.
For every measurement series, two spectra have to be taken in advance before the measurements of the fossil fuels. The first spectrum is called the dark spectrum. It is taken with the light source turned off and a capped light fiber or integration sphere. It measures the dark current of the detector of the spectrometer, and is subtracted from all further measurements, including the second preprocess spectrum called the reference spectrum. The latter is taken with the light source turned on, and the reflectance standard is used for a sample. This reference spectrum conveys all information about the optical setup, i.e. all possible absorbances that occur on the light's way from the light source through the light fiber and collimator, or integration sphere and the probe, to the spectrometer. The MPA spectrometer settings were 10 accumulations per issued spectrum, those of the BWTek spectrometer 500ms integration time and 10 accumulations per issued spectrum.
After acquiring these two spectra, the actual measurements can be done. The light from the light source is led through the light fiber and the collimator, or is diffusively reflected

(a)



(b)

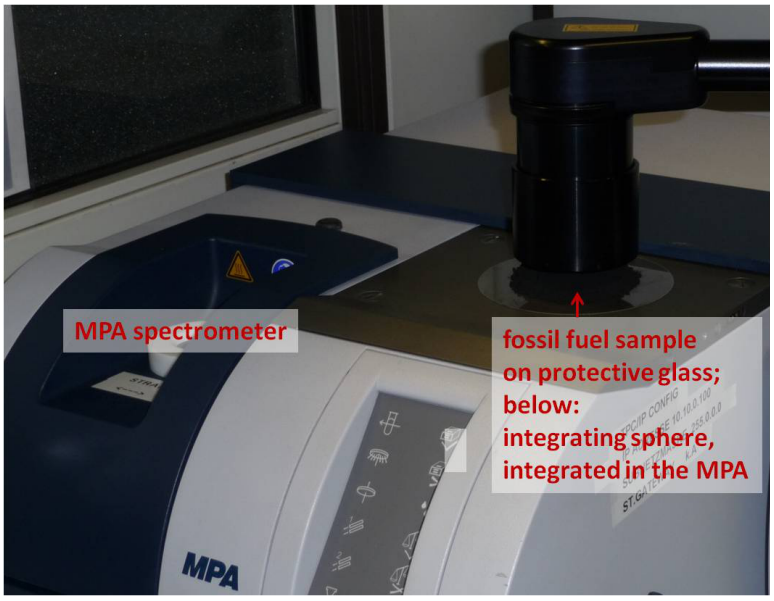


Figure D.7: (a) Measurement setup for the UV-VIS spectra. The container protecting the measurements from ambient light is not shown. (b) Shows the same for the NIR spectra.

into the integration sphere, respectively, and illuminates the fossil fuel. A part of the light, which, from a physical point of view, is energy, is diffusely reflected from the sample surface back to the collimator (or probe). Another part of the energy is diffusely reflected in the upper layer (typically few μm) of the sample some times and then reflected back to the collimator (or probe). Some of the energy is absorbed in the sample. The way in which the light is reflected and the amount that is absorbed depends on the combination of light, i.e. energy levels, and the chemical composition of the sample. Thus, the measurements are material-specific. The reflected light is collected by the collimator (or probe) and sent to the spectrometer. The spectrometer splits the light into its different wavelengths, storing the relative intensity per wavelength.

With the measurement approach above, one data set per spectrometer is recorded. From the reference spectrum and a sample spectrum, the referenced spectrum (used in the final data sets) is calculated by Lambert-Beer's Law. There are 129 spectra of the MPA, stored as digitized signals of 2307 equidistant data points. The 129 spectra of the BWTek spectrometer consist of 1335 equidistant data points. As responses for this data set, sample heat values and percentages of humidity are determined by a chemical laboratory following respective norms. The spectra data is presented in the upper panels of Figure D.8. A histogram of the sample heat values and the percentages of humidity, respectively, can be found in the lower panels.

D.3.3 Evaluation Techniques

Due to the long history of spectroscopic data (for example, first measurements in the infrared range go back to Herschel, 1800), this kind of data is widespread in different areas of data evaluation. In chemometrics, many scientists use multivariate methods like partial least squares regression, principal component analysis or partial least squares discriminant analysis, to name but a few (Otto, 2007; Sattlecker et al., 2010). Also, artificial neural networks of various kinds are used for prediction purposes (Blanco et al., 2000; Balabin and Smirnov, 2012). In statistics, spectroscopic data is popular for illustrating new functional data methods. The Tecator data set, which is distributed through the `R` package `caret` (Kuhn et al., 2015), is a widespread example for this.

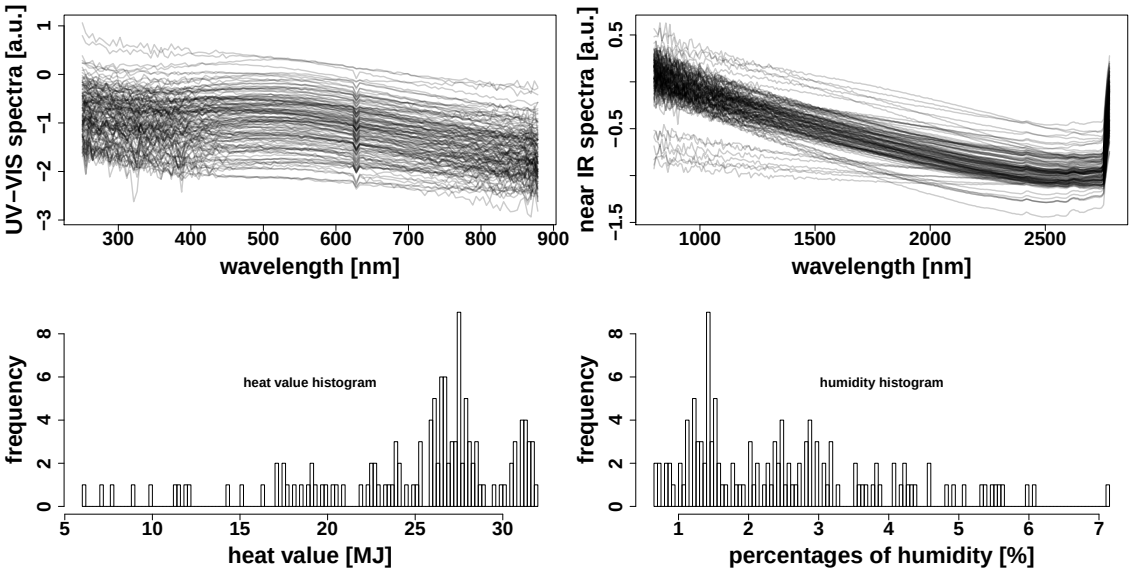


Figure D.8: Spectra of fossil fuel samples of the BWTek spectrometer (upper left) and the MPA spectrometer (upper right). In the lower panels, histograms of the respective responses, namely heat values and percentages of humidity, are shown.

References

Ahdesmaki, M., Zuber, V., Gibb, S., and Strimmer, K. (2015). *Shrinkage Discriminant Analysis and CAT Score Variable Selection*, R package version 1.3.7.

Alonso, A. M., Casado, D., and Romo, J. (2012). Supervised classification for functional data: A weighted distance approach. *Computational Statistics & Data Analysis*, 56:2334 – 2346.

Alonso-Salces, R. M., Guyot, S., Herrero, C., Berrueta, L. A., Drilleau, J.-F., Gallo, B., and Vicente, F. (2005). Chemometric classification of Basque and French ciders based on their total polyphenol contents and CIELab parameters. *Food Chemistry*, 91:91 – 98.

Aneiros-Perez, G. and Vieu, P. (2008). Nonparametric time series prediction: A semi-functional partial linear modeling. *Journal of Multivariate Analysis*, 99:834 – 857.

Antoch, J., Prchal, L., Rosa, M. D., and Sarda, P. (2010). Electricity consumption prediction with functional linear regression using spline estimators. *Journal of Applied Statistics*, 37 (12):2027 – 2041.

Araki, Y., Konishi, S., Kawano, S., and Matsui, H. (2009). Functional Logistic Discrimination Via Regularized Basis Expansions. *Communications in Statistics – Theory and Methods*, 38:2944 – 2957.

Argyriou, A., Evgeniou, T., and Pontil, M. (2007). Multi-task feature learning. In Schölkopf, B., Platt, J., and Hoffman, T., editors, *Advances in Neural Information Processing Systems 19*, pages 41 – 48. MIT Press.

Athitsos, V. and Sclaroff, S. (2005). Boosting Nearest Neighbor Classifiers for Multiclass Recognition. *IEEE Workshop on Learning in Computer Vision and Pattern Recognition.*

Bajtarevic, A., Ager, C., Pienz, M., Klieber, M., Schwarz, K., Ligor, M., Ligor, T., Filipiak, W., Denz, H., Fiegl, M., Hilbe, W., Weiss, W., Lukas, P., Jamnig, H., Hackl, M., Haidenberger, A., Buszewski, B., Miekisch, W., Schubert, J., and Amann, A. (2009). Noninvasive detection of lung cancer by analysis of exhaled breath. *BMC Cancer*, 9:1 – 16.

Balabin, R. and Smirnov, S. (2012). Interpolation and extrapolation problems of multivariate regression in analytical chemistry: benchmarking the robustness on near-infrared (NIR) spectroscopy data. *ANALYST*, 137 (7):1604 – 1610.

Bandyopadhyay, R., Bag, A., Tudu, B., and Bhattacharyya, N. (2011). Electronic nose sensor array optimization using rough set theory. *AIP Conf. Proc.*, 1362:64 – 65.

Berrueta, L. A., Alonso-Salces, R. M., and Héberger, K. (2007). Supervised pattern recognition in food analysis. *Journal of Chromatography A*, 1158:196 – 214.

Bischl, B., Schiffner, J., and Weihs, C. (2013). Benchmarking local classification methods. *Computational Statistics*, 28:2599 – 2619.

Blanco, M., Coello, J., Iturriaga, H., Maspoch, S., and Pages, J. (2000). NIR calibration in non-linear systems: different PLS approaches and artificial neural networks. *Chemometrics and Intelligent Laboratory Systems*, 50:75 – 82.

Bohrn, U., Mucha, A., Werner, C., Trattner, B., Bäcker, M., Krumbe, C., Schienle, M., Stütz, E., Schmitt-Landsiedel, D., Fleischer, M., Wagner, P., and Schöning, M. (2013). A critical comparison of cell-based sensor systems for the detection of Cr(VI) in aquatic environment. *Sensors and Actuators, B: Chemical*, 182:58 – 65.

Bohrn, U., Stütz, E., Fleischer, M., Schöning, M., and Wagner, P. (2011). Eukaryotic cell lines as a sensitive layer for rapid monitoring of carbon monoxide. *Physical Status Solidi A*, 208:1345 – 1350.

Bohrn, U., Stütz, E., Fuchs, K., Fleischer, M., Schöning, M., and Wagner, P. (2012). Monitoring of irritant gas using a whole-cell-based sensor system. *Sensors and Actuators, B: Chemical*, 175:208 – 217.

Bondell, H. D. and Reich, B. J. (2008). Simultaneous Regression Shrinkage, Variable Selection, and Supervised Clustering of Predictors with OSCAR. *Biometrics*, 64:115 – 123.

Bongiorno, E., Goia, A., Salinelli, E., and Vieu, P., editors (2014). *Contributions in infinite-dimensional statistics and related topics.* Societa Editrice Esculapio.

Bosq, D. (2000). *Linear Processes in Function Spaces. Theory and Applications. Lecture Notes in Statistics, vol. 149.* Springer.

Bosq, D. and Blanke, D. (2007). *Inference and Prediction in Large Dimensions.* Dunod and John Wiley & Sons, Ltd.

Breiman, L. (2001). Random Forests. *Machine Learning*, 45:5 – 32.

Breiman, L., Cutler, A., Liaw, A., and Wiener, M. (2012). *randomForest : Breiman and Cutler′s random forests for classification and regression*, R package version 4.6-7.

Brent, R., Kingston, R., and Moore, D., editors (1988). *Current Protocols in Molecular Biology*. John Wiley & Sons Inc.

Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78 (1):1 – 3.

Brockhaus, S., Melcher, M., Leisch, F., and Greven, S. (2017). Boosting flexible functional regression models with a high number of functional historical effects. *Statistics and Computing*, 27 (4):913 – 926.

Buja, A., Hastie, T., and Tibshirani, R. (1989). Linear smoothers and additive models. *The Annals of Statistics*, 17 (2):453 – 555.

Burba, F., Ferraty, F., and Vieu, P. (2009). k-nearest neighbour method in functional nonparametric regression. *Journal of Nonparametric Statistics*, 21 (4):453 – 469.

Cardot, H., Ferraty, F., and Sarda, P. (2003). Spline estimators for the functional linear model. *Statistica Sinica*, 13 (3):571 – 592.

Cardot, H. and Sarda, P. (2005). Estimation in generalized linear models for functional data via penalized likelihood. *Journal of Multivariate Analysis*, 92:24 – 41.

Carlo, S. D., Falasconi, M., Sanchez, E., Scionti, A., Squillero, G., and Tonda, A. (2011). Increasing pattern recognition accuracy for chemical sensing by evolutionary based drift compensation. *Pattern Recognition Letters*, 32 (13):1594 – 1603.

Cederbaum, J., Pouplier, M., Hoole, P., and Greven, S. (2016). Functional Linear Mixed Models for Irregularly or Sparsely Sampled Data. *Statistical Modelling*, 16 (1):67 – 88.

Ceriotti, L., Kob, A., Drechsler, S., Ponti, J., Thedinga, E., Colpo, P., Ehret, R., and Rossi, F. (2007). Online monitoring of BALB/3T3 metabolism and adhesion with multiparametric chip-based system. *Analytical Biochemistry*, 371:92 – 104.

Chen, D., Hall, P., and Müller, H. (2011). Single and multiple index functional regression models with nonparametric link. *The Annals of Statistics*, 39 (3):1720 – 1747.

Chen, J. and Li, H. (2013). Variable selection for sparse Dirichlet-Multinomial regression with an application to microbiome data analysis. *The Annals of Applied Statistics*, 7(1):418 – 442.

Clemmensen, L., Hastie, T., Witten, D., and Ersboll, B. (2011). Sparse Discriminant Analysis. *Technometrics*, 53 (4):406 – 413.

Crainiceanu, C., Reiss, P., Goldsmith, J., Huang, L., Huo, L., Scheipl, F., Swihart, B., Greven, S., Harezlak, J., Kundu, M. G., Zhao, Y., McLean, M., and Xiao, L. (2013). $refund$: *Regression with Functional Data*, R package version 0.1-9.

Cuevas, A. (2014). A partial overview of the theory of statistics with functional data. *Journal of Statistical Planning and Inference*, 147:1 – 23.

Delaigle, A. and Hall, P. (2010). Defining probability density for a distribution of random functions. *Annals of Statistics*, 38 (2):1171 – 1193.

Delaigle, A. and Hall, P. (2012). Achieving near perfect classification for functional data. *Journal of the Royal Statistical Society: Series B*, 74 (2):267 – 286.

Di, C., Crainiceanu, C. M., Caffo, B. S., and Punjabi, N. M. (2009). Multilevel functional principal component analysis. *The Annals of Applied Statistics*, 3(1):458 – 488.

Dymerski, T., Gebicki, J., Wisniewska, P., Sliwinska, M., Wardencki, W., and Namiesnik, J. (2013). Application of the Electronic Nose Technique to Differentiation between Model Mixtures with COPD Markers. *Sensors*, 13:5008 – 5027.

Ehret, R., Baumann, W., Brischwein, M., Schwinde, A., and Wolf, B. (1998). On-line control of cellular adhesion with impedance measurements using interdigitated electrode structures. *Medical & Biological Engineering & Computing*, 36 (3):365 – 370.

Eltzov, E. and Marks, R. (2011). Whole-cell aquatic biosensors. *Analytical & Bioanalytical Chemistry*, 400:895 – 913.

Epifanio, I. (2008). Shape Descriptors for Classification of Functional Data. *Technometrics*, 50(3):284 – 294.

Epifanio, I. and Ventura-Campos, N. (2011). Functional data analysis in shape analysis. *Computational Statistics & Data Analysis*, 55:2758 – 2773.

Escabias, M., Aguilera, A. M., and Valderrama, M. J. (2004). Principal component estimation of functional logistic regression: discussion of two different approaches. *Journal of Nonparametric Statistics*, 16 (3):365 – 384.

Fahrmeir, L., Kneib, T., and Lang, S. (2009). *Regression - Modelle, Methoden und Anwendungen.* New York: Springer.

Fdez-Ortiz de Vallejuelo, S., Arana, G., de Diego, A., and Madariaga, J. M. (2011). Pattern recognition and classification of sediments according to their metal content using chemometric tools. A case study: The estuary of Nerbioi-Ibaizabal River (Bilbao, Basque Country). *Chemosphere*, 85:1347 – 1352.

Febrero-Bande, M., de la Fuente, M. O., Galeano, P., Nieto, A., and Garcia-Portugues, E. (2013). *fda.usc :. Functional Data Analysis and Utilities for Statistical Computing*, R package version 1.1.0.

Febrero-Bande, M. and Gonzalez-Manteiga, W. (2013). Generalized additive models for functional data. *Test*, 22 (2):278 – 292.

Ferraty, F., Goia, A., Salinelli, E., and Vieu, P. (2013). Functional projection pursuit regression. *TEST*, 22 (2):293 – 320.

Ferraty, F. and Romain, Y., editors (2011). *The Oxford Handbook of Functional Data Analysis*. Oxford University Press.

Ferraty, F. and Vieu, P. (2003). Curves discrimination: a nonparametric functional approach. *Computational Statistics & Data Analysis*, 44:161 – 173.

Ferraty, F. and Vieu, P. (2006). *Nonparametric Functional Data Analysis*. Springer Science and Business Media, New York.

Ferraty, F. and Vieu, P. (2009). Additive prediction and boosting for functional data. *Computational Statistics & Data Analysis*, 53:1400 – 1413.

Ferre, L. and Yao, A. (2005). Smoothed functional inverse regression. *Statistica Sinica*, 15:665 – 683.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, pages 179 – 188.

Fix, E. and Hodges, J. L. (1951). Discriminatory analysis - nonparametric discrimination: Consistency properties. Technical report, US Air Force School of Aviation Medicine, Randolph Field Texas.

Fuchs, K., Gertheiss, J., and Tutz, G. (2015a). Nearest neighbor ensembles for functional data with interpretable feature selection. *Chemometrics and Intelligent Laboratory Systems*, 146:186 – 197.

Fuchs, K., Pößnecker, W., and Tutz, G. (2016). Classification of functional data with k-nearest-neighbor ensembles by fitting constrained multinomial logit models. *arXiv:1612.04710v2 [stat.ME]*. submitted to Chemometrics and Intelligent Laboratory Systems.

Fuchs, K., Scheipl, F., and Greven, S. (2015b). Penalized scalar-on-functions regression with interaction term. *Computational Statistics & Data Analysis*, 81:38 – 51.

Gertheiss, J., Maity, A., and Staicu, A.-M. (2013). Variable selection in generalized functional linear models. *Stat*, 2:86 – 101.

Gertheiss, J. and Tutz, G. (2009). Feature selection and weighting by nearest neighbor ensembles. *Chemometrics and Intelligent Laboratory Systems*, 99:30 – 38.

Gneiting, T. and Raftery, A. E. (2007). Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association*, 102 (477):359 – 378.

Goldsmith, J., Bobb, J., Crainiceanu, C., Caffo, B., and Reich, D. (2011). Penalized functional regression. *Journal of Computational and Graphical Statistics*, 20 (4):830 – 851.

Goldsmith, J., Greven, S., and Crainiceanu, C. (2013). Corrected Confidence Bands for Functional Data Using Principal Components. *Biometrics*, 69 (1):41 – 51.

Goldsmith, J. and Scheipl, F. (2014). Estimator selection and combination in scalar-on-function regression. *Computational Statistics & Data Analysis*, 70:362 – 372.

Gonzalez Manteiga, W. and Vieu, P. (2007). Editorial - statistics for functional data. *Computational Statistics & Data Analysis*, 51 (10):4788 – 4792.

Gosangi, R. and Gutierrez-Osuna, R. (2014). Active classification with arrays of tunable chemical sensors. *Chemometrics and Intelligent Laboratory Systems*, 132:91 – 102.

Greven, S. and Scheipl, F. (2017). A General Framework for Functional Regression Modelling. *Statistical Modelling*, 17 (1 – 2):1 – 35.

Guijarro, C., Fuchs, K., Bohrn, U., Stütz, E., and Wölfl, S. (2015). Simultaneous detection of multiple bioactive pollutants using a multiparametric biochip for water quality monitoring. *Biosensors and Bioelectronics*, 72:71 – 79.

Guo, Y., Hastie, T., and Tibshirani, R. (2007). Regularized linear discriminant analysis and its application in microarrays. *Biostatistics*, 8 (1):86 – 100.

Guo, Y., Hastie, T., and Tibshirani, R. (2012). *Shrunken Centroids Regularized Discriminant Analysis*, R package version 1.0.2-2.

Gutierrez-Osuna, R., Gosangi, R., and Hierlemann, A. (2011). Advances in Active and Adaptive Chemical Sensing. *AIP Conference Proceedings*, 1362:11 – 12.

Hall, P., Park, B. U., and Samworth, R. J. (2008). Choice of neighbor order in nearest-neighbor classification. *The Annals of Statistics*, 36 (5):2135 – 2152.

Hall, P., Poskitt, D. S., and Presnell, B. (2001). A Functional Data-Analytic Approach to Signal Discrimination. *Technometrics*, 43 (1):1 –9.

Happ, C. (2017). $funData : An\ S4\ Class\ for\ Functional\ Data$, *R* package version 1.0.

Happ, C. and Greven, S. (2016+). Multivariate Functional Principal Component Analysis for Data Observed on Different (Dimensional) Domains. *Journal of the American Statistical Association*, to appear.

Happ, C. M. (2013). Identifiability in scalar-on-functions regression. Master's thesis, Department of Statistics, Ludwig-Maximilians-Universität München, Germany.

Harville, D. A. (2000). *Matrix algebra from a statistician's perspective.* New York: Springer.

Hastie, T., Buja, A., and Tibshirani, R. (1995). Penalized Discriminant Analysis. *The Annals of Statistics*, 23 (1):73 – 102.

Hastie, T. and Tibshirani, R. (1986). Generalized additive models. *Statistical Science*, 1 (3):297 – 310.

Hastie, T., Tibshirani, R., and Friedman, J. (2011). *The Elements of Statistical Learning.* Springer Science and Business Media, New York.

Hastie, T., Tibshirani, R., Friedman, J., and Halvorsen, K. (2015a). *ElemStatLearn : Data sets, functions and examples from the book : "The Elements of Statistical Learning, Data Mining, Inference, and Prediction" by Trevor Hastie, Robert Tibshirani and Jerome Friedman*, R package version 2012-04-05.

Hastie, T., Tibshirani, R., Leisch, F., Hornik, K., and Ripley, B. D. (2015b). *mda : Mixture and flexible discriminant analysis*, R package version 0.4-4.

Hayat, M., Tahir, M., and Khan, S. A. (2014). Prediction of protein structure classes using hybrid space of multi-profile Bayes and bi-gram probability feature spaces. *Journal of Theoretical Biology*, 346:8 – 15.

He, G., Müller, H., and J.L.Wang. (2000). Extending correlation and regression from multivariate to functional data. In Puri, M., editor, *Asymptotics in Statistics and Probability*, pages 301 – 315. VSP International Science Publishers.

Herschel, W. (1800). Experiments on the Refrangibility of the Invisible Rays of the Sun. *Philosophical Transactions of the Royal Society of London*, 90:284 – 292.

Horvath, L. and Kokoszca, P. (2012). *Inference for Functional Data with Applications.* Springer, New York.

Huang, S. and Pang, L. (2012). Comparing Statistical Methods for Quantifying Drug Sensitivity based on in vitro doseresponse assays. *ASSAY and Drug Development Technologies*, 10 (1).

Ivanescu, A., Staicu, A., Scheipl, F., and Greven, S. (2015). Penalized function-on-function regression. *Computational Statistics*, 30 (2):539 – 568.

Jacques, J. and Preda, C. (2014). Functional Data Clustering: A Survey. *Advances in Data Analysis and Classification*, 8:231 – 255.

James, G. (2002). Generalized Linear Models With Functional Predictors. *Journal of the Royal Statistical Society, Series B*, 64:411 – 432.

James, G. M. (2001). Functional linear discriminant analysis for irregularly sampled curves. *J. R. Statist. Soc. B*, 63 (3):533 – 550.

Japon-Lujan, R., Ruiz-Jiménez, J., and de Castro, M. D. L. (2006). Discrimination and Classification of Olive Tree Varieties and Cultivation Zones by Biophenol Contents. *J. Agric. Food Chem.*, 54:9706 – 9712.

Jenatton, R., Mairal, J., Obozinski, G., and Bach, F. (2011). Proximal methods for hierarchical sparse coding. *Journal of Machine Learning Research*, 12:2297 – 2334.

Ji, J. and Zhao, Q. (2013). A hybrid SVM based on nearest neighbor rule. *International Journal of Wavelets, Multiresolution and Information Processing*, 11 (6).

Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer.

Jolliffe, I. T., Morgan, B. J. T., and Young, P. J. (1996). A simulation study of the use of principal components in linear discriminant analysis. *Journal of Statistical Computation and Simulation*, 55 (4):353 – 366.

Jurka, T. P. and Tsuruoka, Y. (2013). $maxent : Low - memory\ Multinomial\ Logistic\ Regression\ with\ Support\ for\ Text\ Classification$, R package version 1.3.3.1.

Kauermann, G. and Opsomer, J. (2011). Data-driven selection of the spline dimension in penalized spline regression. *Biometrika*, 98 (1):225 – 230.

Kim, N. H., Choi, S. J., Yang, D. J., Bae, J., Park, J., and Kim, I. D. (2014). Highly sensitive and selective hydrogen sulfide and toluene sensors using Pd functionalized WO_3 nanofibers for potential diagnosis of halitosis and lung cancer. *Sensors and Actuators B*, 193:574 – 581.

Kruzlicova, D., Mocak, J., Katsoyannos, E., and Lankmayr, E. (2008). Classification and characterization of olive oils by UV-Vis absorption spectrometry and sensorial analysis. *Journal of Food and Nutrition Research*, 47 (4):181 – 188.

Kubisch, R., Bohrn, U., Fleischer, M., and Stütz, E. (2012). Cell-Based Sensor System Using L6 Cells for Broad Band Continuous Pollutant Monitoring in Aquatic Environments. *Sensors*, 12 (3):3370 – 3393.

Kudraszow, N. and Vieu, P. (2013). Uniform consistency of knn regressors for functional variables. *Statistics and Probability Letters*, 83:1863 – 1870.

Kuhn, M., Wing, J., Weston, S., Williams, A., Keefer, C., Engelhardt, A., Cooper, T., Mayer, Z., and Kenkel, B. (2015). R package *caret*.

Lagarde, F. and Jaffrezie-Renault, N. (2011). Cell-based electrochemical biosensors for water quality assessment. *Analytical & Bioanalytical Chemistry*, 400:947 – 964.

LeBlanc, M. and Tibshirani, R. (1996). Combining Estimates in Regression and Classification. *Journal of the American Statistical Association*, 91 (436):1641 – 1650.

Lee, A. and Reedy, B. J. (1999). Temperature modulation in semiconductor gas sensing. *Sensors and Actuators B*, 60:35 – 42.

Li, B. and Yu, Q. (2008). Classification of functional data: A segmentation approach. *Computational Statistics & Data Analysis*, 52:4790 – 4800.

Li, D., Lei, T., Zhang, S., Shao, X., and Xie, C. (2015). A novel headspace integrated E-nose and its application in discrimination of Chinese medical herbs. *Sensors and Actuators B*, 221:556 – 563.

Lian, H. (2011). Functional partial linear model. *Journal of Nonparametric Statistics*, 23 (1):115 – 128.

Lian, H. and Li, G. (2014). Series expansion for functional sufficient dimension reduction. *Journal of Multivariate Analysis*, 124:150 – 165.

Lovelady, D., Richmond, T., Maggi, A., Lo, C., and Rabson, D. (2007). Distinguishing cancerous from noncancerous cells through analysis of electrical noise. *Physical Review E*, 76 (041908).

Lukasiak, B. M., Zomer, S., Brereton, R. G., Faria, R., and Duncan, J. C. (2007). Pattern recognition and feature selection for the discrimination between grades of commercial plastics. *Chemometrics and Intelligent Laboratory Systems*, 87:18 – 25.

Maierhofer, T. (2017). Classification of Functional Data – Interpretable Ensemble Approaches. Master's thesis, Department of Statistics, Ludwig-Maximilians-Universität München, Germany.

Maierhofer, T. and Fuchs, K. (2017). $classifunc : classification\ of\ functional\ data$, R package.

Makisimovich, N., Vorotyntsev, V., Nikitina, N., Kaskevich, O., Karabun, P., and Martynenko, F. (1996). Adsorption semiconductor sensor for diabetic ketoacidosis diagnosis. *Sensors and Actuators B*, 35 – 36:419 – 421.

Marx, B. and Eilers, P. (2005). Multidimensional penalized signal regression. *Technometrics*, 47 (1):13 – 22.

Masson, N., Piedrahita, R., and Hannigan, M. (2015). Approach for quantification of metal oxide type semiconductor gas sensors used for ambient air quality monitoring. *Sensors and Actuators B*, 208:339 – 345.

Matsui, H. (2014). Variable and boundary selection for functional data via multiclass logistic regression modeling. *Computational Statistics & Data Analysis*, 78:176 – 185.

McFadden, D. (1973). Conditional logit analysis of qualitative choice behaviour. In *Frontiers in Econometrics*. ed. by P. Zarembka, Academic Press, New York.

Melvin, I., Weston, J., Leslie, C. S., and Noble, W. S. (2008). Combining classifiers for improved classification of proteins from sequence or structure. *BMC Bioinformatics*, 9:389 – 397.

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F., Chang, C.-C., and Lin, C.-C. (2014). *e*1071 : *Misc Functions of the Department of Statistics* (*e*1071), *TU Wien*, R package version 0.1-9.

Millonig, G., Praun, S., Netzer, M., Baumgartner, C., Dornauer, A., Mueller, S., Villinger, J., and Vogel, W. (2010). Non–invasive diagnosis of liver diseases by breath analysis using an optimized ion–molecule reactionmass spectrometry approach: a pilot study. *Biomarkers*, 15 (4):297 – 306.

Möller, A., Tutz, G., and Gertheiss, J. (2016). Random forests for functional covariates. *Journal of Chemometrics*, 30 (12):715 – 725.

Morris, J. S. (2015). Functional Regression. *Annual Review of Statistics and Its Application*, 2:321 – 359.

Müller, H. G. and Stadtmüller, U. (2005). Generalized Functional Linear Models. *The Annals of Statistics*, 33 (2):774 – 805.

Nava, R., Escalante-Ramírez, B., Cristóbal, G., and Estépar, R. S. J. (2014). Extended Gabor approach applied to classification of emphysematous patterns in computed tomography. *Med. Biol. Eng. Comput.*, 52:393 – 403.

Nelder, J. and Wedderburn, R. (1972). Generalized linear models. *Journal of the Royal Statistical Society, Series A*, 3:370 – 384.

Nguyen, H. D., McLachlan, G. J., and Wood, I. A. (2016). Mixtures of spatial spline regressions for clustering and classification. *Computational Statistics & Data Analysis*, 93:76 – 85.

Otto, M. (2007). *Chemometrics: Statistics and Computer Application in Analytical Chemistry.* Wiley – VCH.

Pashami, S., Lilienthal, A. J., Schaffernicht, E., and Trincavelli, M. (2013). TREFEX: Trend estimation and change detection in the response of MOX gas sensors. *Sensors*, 13:7323 – 7344.

Peveler, W. J., Binions, R., Hailes, S. M. V., and Parkin, I. P. (2013). Detection of explosive markers using zeolite modified gas sensors. *Journal of Materials Chemistry A*, 1:2613 – 2620.

Piedrahita, R., Xiang, Y., Masson, N., Ortega, J., Collier, A., Jiang, Y., Li, K., Dick, R. P., Lv, Q., Hannigan, M., and Shang, L. (2014). The next generation of low-cost personal air quality sensors for quantitative exposure monitoring. *Atmospheric Measurement Techniques*, 7:3325 – 3336.

Pößnecker, W. (2015). *MRSP : Multinomial Response Models with Structured Penalties*, R package version 0.4.3.

Prchal, L. and Sarda, P. (2007). Spline estimator for functional linear regression with functional response. *unpublished. URL http://www.math.univ-toulouse.fr/staph/PAPERS/flm_prchal_sarda.pdf*.

Preda, C., Saporta, G., and Leveder, C. (2007). PLS classification of functional data. *Computational Statistics*, 22:223 – 235.

Prevolnik, M., Andronikov, D., Zlender, B., i Furnols, M. F., Novic, M., Skorjanc, D., and Candek-Potokar, M. (2014). Classification of dry-cured hams according to the maturing time using near infrared spectra and artificial neural networks. *Meat Science*, 96:14 – 20.

Przewozniczek, M., Walkowiak, K., and Wozniak, M. (2011). Optimizing distributed computing systems for k-nearest neighbours classifiers – evolutionary approach. *Logic Journal of the IGPL*, 19 (2):357 – 372.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Ramamoorthy, R., Dutta, P., and Akbar, S. (2003). Oxygen sensors: Materials, methods, designs and applications. *Journal of Materials Science*, 38 (21):4271 – 4282.

Ramsay, J. and Silverman, B. (2005). *Functional Data Analysis.* Springer: New York.

Ramsay, J. O. and Li, X. (1998). Curve registration. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60:351 – 363.

Ramsay, J. O. and Silverman, B. W. (2002). *Applied functional data analysis*. Springer-Verlag Inc., New York.

Rao, C. R. (1973). *Linear Statistical Inference and Its Applications.* John Wiley & Sons Inc.

Reiss, P. and Ogden, R. (2007). Functional principal component regression and functional partial least squares. *Journal of the American Statistical Association*, 102 (479):984 – 996.

Reiss, P. and Ogden, R. (2009). Smoothing parameter selection for a class of semiparametric linear models. *Journal of the Royal Statistical Society B*, 71 (2):505 – 523.

Ripley, B., Venables, B., Bates, D. M., Hornik, K., Gebhardt, A., and Firth, D. (2014). $MASS : Support\ Functions\ and\ Datasets\ for\ Venables\ and\ Ripley's\ MASS$, R package version 7.3-30.

Rossi, F. and Villa, N. (2006). Support vector machine for functional data classification. *Neurocomputing*, 69:730 – 742.

Ruppert, D. (2002). Selecting the number of knots for penalized splines. *Journal of Computational and Graphical Statistics*, 11 (4):735 – 757.

Ruppert, D., Wand, M., and Carroll, R. (2003). *Semiparametric Regression.* Cambridge University Press.

Sattlecker, M., Bessant, C., Smith, J., and Stone, N. (2010). Investigation of support vector machines and Raman spectroscopy for lymph node diagnostics. *ANALYST*, 135 (5):895 – 901.

Scheipl, F. and Greven, S. (2016). Identifiability in penalized function-on-function regression models. *Electronic Journal of Statistics*, 10:495 – 526.

Selten, R. (1998). Axiomatic Characterization of the Quadratic Scoring Rule. *Experimental Economics*, 1:43 – 62.

Shang, H. L. (2014). A survey of functional principal component analysis. *AStA Adv Stat Anal*, 98 (2):121 – 142.

Simon, N., Friedman, J., and Hastie, T. (2013). A blockwise descent algorithm for group-penalized multiresponse and multinomial regression. *arXiv:1311.6529 [stat.CO].*

Soetaert, K., den Meersche, K. V., and van Oevelen, D. (2013). $limSolve : Solving\ Linear\ Inverse\ Models$, R package version 1.5.5.

Sorensen, H., Goldsmith, J., and Sangalli, L. (2013). An introduction with medical applications to functional data analysis. *Statistics in Medicine*, 32:5222 – 5240.

Styan, G. P. H. (1973). Hadamard Products and Multivariate Statistical Analysis. *LINEAR ALGEBRA AND ITS APPLICATIONS*, 6:217 – 240.

Thedinga, E., Kob, A., Holst, H., Keuer, A., Drechsler, S., Niendorf, R., Baumann, W., Freund, I., Lehmann, M., and Ehret, R. (2007). Online monitoring of cell metabolism for studying pharmacodynamic effects. *Toxicology and Applied Pharmacology*, 220:33 – 44.

Thévenot, D., Toth, K., Durst, R., and Wilson, G. (2001). Electrochemical biosensors: recommended definitions and classification. *Biosensors and Bioelectronics*, 16 (1-2):121 – 131.

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *J. R. Statist. Soc. B*, 58 (1):267 – 288.

Tutz, G. (2012). *Regression for Categorical Data*. Cambridge University Press, New York.

Tutz, G. and Koch, D. (2016). Improved nearest neighbor classifiers by weighting and selection of predictors. *Statistics and Computing*, 26:1039 – 1057.

Tutz, G., Pößnecker, W., and Uhlmann, L. (2015). Variable selection in general multinomial logit models. *Computational Statistics & Data Analysis*, 82:207 – 222.

Tutz, G. and Ulbricht, J. (2009). Penalized regression with correlation-based penalty. *Statistical Computing*, 19:239 – 253.

Ullah, S. and Finch, C. (2013). Applications of functional data analysis: A systematic review. *BMC Medical Research Methodology*, 13 (43).

Usset, J., Staicu, A. M., and Maity, A. (2016). Interaction models for functional regression. *Computational Statistics & Data Analysis*, 94:317 – 329.

van der Laan, M. and Dudoit, S. (2003). Unified cross-validation methodology for selection among estimators and a general cross-validated adaptive epsilon-net estimator: Finite sample oracle inequalities and examples. Paper 130, Division of Biostatistics, University of California, Berkeley.

Vapnik, V. (1996). *The Nature of Statistical Learning Theory.* Springer, New York.

Vincent, M. and Hansen, N. (2014). Sparse group lasso and high dimensional multinomial classification. *Computational Statistics & Data Analysis*, 71:771 – 786.

Wang, G., Lin, N., and Zhang, B. (2012). Functional linear regression after spline transformation. *Computational Statistics & Data Analysis*, 56:587 – 601.

Wang, H. and Leng, C. (2008). A note on adaptive group lasso. *Computational Statistics & Data Analysis*, 52:5277 – 5286.

Wang, J.-L., Chiou, J.-M., and Müller, G. (2016). Functional Data Analysis. *Annual Review of Statistics and Its Application*, 3:257 – 295.

Wolf, B., Brischwein, M., Baumann, W., Ehret, R., and Kraus, M. (1998). Monitoring of cellular signalling and metabolism with modular sensor-technique: The PhysioControl-Microsystem (PCM®). *Biosens. Bioelectron.*, 13:501 – 509.

Wolpert, D. H. (1992). Stacked Generalization. *Neural Networks*, 5:241 – 259.

Wong, C., Li, Y., Lee, C., and Huang, C. H. (2010). Ensemble learning algorithms for classification of mtDNA into haplogroups. *Briefings in Bioinformatics*, 12 (1):1 – 9.

Wood, S. (2004). Stable and efficient multiple smoothing parameter estimation for generalized additive model. *Journal of the American Statistical Association*, 99 (467):673 – 686.

Wood, S. (2006). *Generalized Additive Models: an Introduction with R.* Boca Raton: CRC-Chapman and Hall.

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

Wood, S. (2013). Manual of R package *mgcv*. version 1.7-26.

Wood, S. (2014). *mgcv : Mixed GAM Computation Vehicle with GCV/ AIC/ REML Smoothness Estimation*, R package version 1.8-4.

Wood, S. (2017). Manual of R package *mgcv*. version 1.8-17.

Yang, W.-H., Wikle, C., Holan, S., and Wildhaber, M. (2013). Ecological prediction with nonlinear multivariate time-frequency functional data models. *Journal of Agricultural, Biological, and Environmental Statistics*, 18 (3):450 – 474.

Yao, F. and Müller, H. (2010). Functional quadratic regression. *Biometrika*, 97 (1):49 – 64.

Yao, F., Müller, H.-G., and Wang, J.-L. (2005). Functional Data Analysis for Sparse Longitudinal Data. *Journal of the American Statistical Association*, 100 (470):577 – 590.

Yassouridis, C. and Leisch, F. (2016). Benchmarking different clustering algorithms on functional data. *Advances in Data Analysis and Classification*, pages 1 – 26.

Zhang, X. and Wang, J.-L. (2016). From sparse to dense functional data and beyond. *The Annals of Statistics*, 44 (5):2281 – 2321.

Zhu, H., Vannucci, M., and Cox, D. D. (2010). A Bayesian Hierarchical Model for Classification with Selection of Functional Predictors. *Biometrics*, 66:463 – 473.

Zou, H. (2006). The Adaptive Lasso and its Oracle Properties. *Journal of the American Statistical Association*, 101 (476):1418 – 1429.

Eidesstattliche Versicherung

(Siehe Promotionsordnung vom 12. Juli 2011, §8 Abs. 2 Pkt. 5)

Hiermit erkläre ich an Eides statt, dass die Dissertation von mir selbstständig, ohne unerlaubte Beihilfe angefertigt ist.

München, den 30.08.2017 Karen Fuchs