PARAMETER ESTIMATION IN GENERALIZED PARTIAL LINEAR MODELS WITH TIKHANOV REGULARIZATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

BELGİN KAYHAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
SCIENTIFIC COMPUTING

SEPTEMBER 2010

Approval of the thesis:

PARAMETER ESTIMATION IN GENERALIZED PARTIAL LINEAR MODELS WITH TIKHANOV REGULARIZATION

submitted by **BELGİN KAYHAN** in partial fulfillment of the requirements for the degree of **Master of Science in Department of Scientific Computing, Middle East Technical University** by,

Prof. Dr. Ersan Akyıldız _____
Director, Graduate School of **Applied Mathematics**

Prof. Dr. Bülent Karasözen _____
Head of Department, **Scientific Computing**

Prof. Dr. Bülent Karasözen _____
Supervisor, **Department of Mathematics, METU**

Prof. Dr. Gerhard-Wilhelm Weber _____
Co-supervisor, **Institute of Applied Mathematics, METU**

Examining Committee Members:

Committee Member 1 Assoc. Prof. Dr. İnci Batmaz _____
Department of Statistics, METU

Committee Member 2 Prof. Dr. Bülent Karasözen _____
Department of Mathematics, METU

Committee Member 3 Assist. Prof. Dr. Cem İyigün _____
Department of Industrial Engineering, METU

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: BELGİN KAYHAN

Signature :

ABSTRACT

PARAMETER ESTIMATION IN GENERALIZED PARTIAL LINEAR MODELS WITH TIKHANOV REGULARIZATION

Kayhan, Belgin

M.S., Department of Scientific Computing

Supervisor : Prof. Dr. Bülent Karasözen

Co-Supervisor : Prof. Dr. Gerhard-Wilhelm Weber

September 2010, 101 pages

Regression analysis refers to techniques for modeling and analyzing several variables in statistical learning. There are various types of regression models. In our study, we analyzed *Generalized Partial Linear Models* ($GPLMs$), which decomposes input variables into two sets, and additively combines classical linear models with nonlinear model part. By separating linear models from nonlinear ones, an inverse problem method Tikhonov regularization was applied for the nonlinear submodels separately, within the entire GPLM. Such a particular representation of submodels provides both a better accuracy and a better stability (regularity) under noise in the data.

We aim to smooth the nonparametric part of GPLM by using a modified form of *Multiple Adaptive Regression Spline* ($MARS$) which is very useful for high-dimensional problems and does not impose any specific relationship between the predictor and dependent variables. Instead, it can estimate the contribution of the basis functions so that both the additive and interaction effects of the predictors are allowed to determine the dependent variable. The MARS algorithm has two steps: the forward

and backward stepwise algorithms. In the first one, the model is built by adding basis functions until a maximum level of complexity is reached. On the other hand, the backward stepwise algorithm starts with removing the least significant basis functions from the model.

In this study, we propose to use a penalized residual sum of squares (PRSS) instead of the backward stepwise algorithm and construct PRSS for MARS as a Tikhonov regularization problem. Besides, we provide numeric example with two data sets; one has interaction and the other one does not have. As well as studying the regularization of the nonparametric part, we also mention theoretically the regularization of the parametric part. Furthermore, we make a comparison between Infinite Kernel Learning (IKL) and Tikhonov regularization by using two data sets, with the difference consisting in the (non-)homogeneity of the data set. The thesis concludes with an outlook on future research.

Keywords: Generalized Partial Linear Model, Tikhonov Regularization, CMARS, Iteratively Reweighted Penalty Methods, Kernel Learning

ÖZ

GENELLEŞTİRİLMİŞ PARÇALI DOĞRUSAL MODELLERDE TİKHANOV DÜZENLEME İLE PARAMETRE TAHMİNİ

Kayhan, Belgin
Yüksek Lisans, Bilimsel Hesaplama
Tez Yöneticisi : Prof. Dr. Bülent Karasözen
Ortak Tez Yöneticisi : Prof. Dr. Gerhard-Wilhelm Weber

Eylül 2010, 101 sayfa

Regresyon analizi, istatistiksel öğrenmede çok sayıda bağımsız değişkenin modellendiği ve analiz edildiği bir yöntemdir. Birçok regresyon model çeşidi vardır. Bu çalısmada biz, genelleştirilmiş parçalı doğrusal modelleri inceledik. Genelleştirilmiş parçalı doğrusal modeller bağımsız değişkenleri iki kısma ayırarak, klasik doğrusal modellerle doğrusal olmayan modelleri eklemeli olarak birleştirir. Doğrusal modelleri doğrusal olmayan modellerden ayırarak, tüm genelleştirilmiş parçalı doğrusal modeller arasında, doğrusal olmayan kısım için bir ters problem yöntemi olan Tikhonov duzenlemesi uygulanmıştır. Alt modellerin bu şekilde gösterimi gürültü içeren verilerde daha iyi bir tutarlılık ve doğruluk sağlamaktadır.

Bu çalışmada, doğrusal olmayan kısmı düzenlemek için çok değişkenli uyarlanabilir regresyon eğrilerini (MARS) değiştirerek kullanmayı amaçlamaktayız. Çok boyutlu problemlerin çözümünde elverişli bir yöntem olan MARS, bağımsız değişkenlerle bağımlı değişken arasında belirli bir ilişki biçimi öngörmez. Onun yerine, bağımlı değişkeni tanımlamak için bağımsız değişkenlerin eklemeli ve etkileşimsel katkılarına yer verir.

MARS algoritması ekleyerek ve eleyerek ilerleyen iki aşamalı bir algoritmadan oluşmaktadır. İlk aşamada en yüksek karmaşıklık düzeyine ulaşıncaya kadar temel fonksiyonlar eklenerek model yapılandırılır. İkinci aşamada ise modele katkısı en az fonksiyonlar eklenir.

Bu çalısmada biz, MARS'in ikinci aşamasını oluşturan geriye doğru eleme yöntemi yerine penaltı yöntemini kullanmayı önermekteyiz. Bu amaçla, bir Tikhonov düzenlemesi problemi olarak MARS icin cezalandırılmıs hata kareler toplamı oluşturmaktadır. Bununla birlikte, etkileşimli ve etkileşimsiz iki veri kümesi kullanarak sayısal bir örnek vermekteyiz. Parametrik kısmın düzenlenmesi çalışmasına ek olarak parametrik olmayan kısmın düzenlenmesinden de teorik olarak bahsetmekteyiz. Ayrıca, sonsuz çekirdek öğrenimi (IKL) ile Tikhonov, homojen ve homojen olmayan iki kümesini seti kullanılarak karşılaştırılmaktayız. Tez, ileriki çalışmalara bir bakış açısı sağlayarak sonlanmaktadır.

Anahtar Kelimeler: Genelleştirilmiş parçalı doğrusal modeller, Tikhonov düzenleme, CMARS, Tekrarlı ve yeniden ağırlıklandırılan ceza yöntemi, Çekirdek öğrenimi

To my family and best friends

ACKNOWLEDGMENTS

First of all, I would like to thank to my supervisor, Prof. Dr. Bülent Karasözen, and co-supervisor Prof. Dr. Gerhard-Wilhelm Weber for their friendship, motivation and support during this study.

I would like to thank to Assoc. Prof. Dr. Inci Batmaz and Assist. Prof. Dr. Cem Iyigün for their guidance and valuable contribution to this thesis.

Also, I would like to give special thanks to Assist. Prof. Dr. Süreyya Özögür-Akyüz for her valuable contribution to this study. I especially deeply thank to MSc. Gürkan Üstünkar for his endless support, patience and help.

I would like to thank to Assist. Prof. Dr. Pakize Taylan for her valuable contribution to this study.

In this thesis, I and Gül Çelik studied together and I am thankful her for all her helps, not letting me feel alone and motivating me.

I am grateful to my friend Ayşe Özmen for her encouragement, understanding and guidance throughout this study.

I would like thank to MSc. Fatma Yerlikaya for her friendship and also for her help during this study.

I am grateful to my coworkers Yeşne Aren and Vuslat Sabah for their support and understanding.

Finally, I would like to special thanks to my family and friends for their continuous support, love and patience. They always believed and encouraged me.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Regression analysis refers to techniques for modeling and analyzing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables. There are various types of regression models. Familiar methods such as linear regression and ordinary least-squares regression are parametric ones, because it is possible to define the regression function in terms of a finite number of unknown parameters that are estimated from the data. Nonparametric regression, however, refers to techniques that allow the regression function to lie in a specified set of functions, which may be infinite-dimensional.

The most widely known regression model is *Linear Regression Models (LRM). Generalized Linear Models (GLM)* are an extension of the linear modeling process that allows models to be fit to data that follow probability distributions other than the Normal distribution, such as the Poisson, Binomial, Gamma, etc.. Generalized Linear Models also relax the requirement of the constant variance equality which is required for hypothesis tests in traditional linear models [61].

However, GLM being a linear technique shares the common shortcomings of the linear modeling (LM) approach. Firstly, both need the assumption that data has a distribution of exponential family. Secondly, they are affected by multi-collinearity, outliers and missing values in the data. Besides, it is difficult to use GLM for selecting important predictors and their interactions. Finally, categorical predictors with large numbers of categories can lead to unreliable results due to sparsity-related issues [49].

Data mining is a very popular approach dealing with these problems effectively. Data mining techniques are typically fast, and easily select predictors and their interactions.

Besides, they are minimally affected with missing values, outliers or collinearity. As well, they effectively process high-level categorical predictors [49]. As a data mining technique, *Multiple Adaptive Regression Spline* ($MARS$) is very useful for high dimensional problems and does not impose any specific relationship between the predictor and dependent variables. Instead, it can estimate the contribution of the basis functions so that both the additive and interaction effects of the predictors are allowed to determine the dependent variable.

The use of MARS to enhance GLM building makes the model-building process considerably faster and more efficient [49]. In this study, we will analysis an extended form of GLM, which is known as *Generalized Partial Linear Models* ($GPLMs$). In GPLM, the usual parametric terms are augmented by a single nonparametric component. In other words, GPLM decomposes input variables into two sets and additively combines classical linear models with nonlinear model part.

Generalized partial linear models have a great advantage that consists in some *grouping* which could be done for the input dimensions or features in order to assign appropriate submodels specifically [92]. There are linear, nonlinear ones as well as parametrical and nonparametrical ones. By separating linear models from nonlinear or nonparametrical ones, inverse problem methods such as Tikhonov regularization [3] can be applied for the linear submodels separately, within the entire GPLMs. Such a particular representation of submodels provides both a better accuracy and a better stability (regularity) under noise in the data.

In this thesis, we aim to integrate GPLM with a modified form of MARS. The MARS algorithm has two steps to estimate the model function: these are the forward and backward stepwise algorithms. In the first one, the model is built by adding basis functions until a maximum level of complexity is reached. Whereas, in the backward stepwise algorithm, it starts removing the least significant basis functions from the model. In this study, we propose to use *penalized residual sum of squares* ($PRSS$) to the control complexity and accuracy of the model instead of the backward algorithm and treat it as an optimization problem. This alternative method to the backward stepwise algorithm provides an alternative modeling approach for MARS, named *Conic Multivariate Adaptive Regression Splines* ($CMARS$). Here 'C' represents not

only the word conic but also convex and continuous.

By using penalty terms, we built PRSS changing the form into a *Tikhonov regularization problem* and solve it by using regularization toolbox of MATLAB. As well as studying the regularization of the nonparametric part, we also mention theoretically the regularization of the parametric part. However, for the sake of simplicity, we disregard the parametric part, knowing, however, how to deal with in the presence of linear part. Besides, we provide numerical examples for the regularization of the nonparametric part with two data sets; one has interaction and the other does not have.

Furthermore, we also focused on a classification technique, *Infinite Kernel Learning* (IKL) which is a modern method of Machine Learning (support vector machines). Classification is easier if the data is linear. However, if it is not, then, kernels are very helpful as it is possible to project data into a higher dimensional feature space where usual linear classifiers can be applied to classify the data as if the data is linear. If the data is huge, there is need for many kernels and *multiple kernel learning* is used for heterogeneous and large-scale data. Our method, *Infinite Kernel Learning* (IKL), is based on the motivation of multiple kernel learning. Besides, we analyze three data sets and display the results. Besides, we make a comparison between the two methods; IKL and Tikhonov regularization. For this aim, we use two data sets with the difference consisting in the (non-)homogeneity of the data. After analyzing the data sets, we compare the results of the methods by using some statistical performance measures. We conclude with an outlook to future studies.

CHAPTER 2

LITERATURE SURVEY AND BACKGROUND

2.1 Linear Regression Models

Linear Regression is a statistical technique that correlates the change in the dependent (*response*) variable to the independent (*regressor*) variable(s). In linear regression, the model is not necessarily linear in the independent variables. Instead it depends linearly on the unknown parameters and has a linearly additive relationship. The general form of a Linear Regression Model (LRM) is as follows [66]:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \varepsilon.$$

Here, y represents the response variable, x_i $(i = 1, 2, \ldots, k)$ represent the independent variables and ε is the unobserved random error term. It is assumed that errors are normally distributed and mutually independent zero mean random variables, each with the same variance σ^2. Besides, β_0 is the intercept term, also known as '*bias*' in some fields, and the parameters β_i are unknown regression coefficients measuring the strength of the relationship between independent and dependent variables. In other words, they explain the expected change in y corresponding to the one unit change in x_i assuming ceteris paribus. If it is positive, y increases as x increases.

In general, the goal of linear regression is to find the line that best predicts the dependent variable from a set of data. Numerous procedures have been developed for this purpose but *least-squares estimation* (LSE) is the most popular one by far. However, in some cases it is not useful and it is preferred to use a more general form of it, which is called as *maximum likelihood estimation* (MLE) [39]. Both methods find the line that minimizes the sum of the squares of the vertical distances of the points

from the hyperplane.

2.1.1 Least-Squares Estimation Method

The least-squares method is a simple but powerful prediction method. It can be interpreted as a method of fitting data. For the simple univariate linear model with N observations, the model is

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \quad \text{for} \quad i = 1, 2, \ldots, N,$$

where N is the number of the data with $E(\varepsilon_i) = 0$ and $Var(\varepsilon_i) = \sigma^2$. The least-squares estimates of β_0 and β_1 can be found by minimizing the function of the residual sum of the squares (RSS) between y and its expected value:

$$RSS(\beta_0, \beta_1) = \sum_{i=1}^{N} (y_i - E(y_i))^2 = \sum_{i=1}^{N} (y_i - \beta_0 - \beta_1 x_i)^2.$$

To estimated values of β_0 and β_1 can be found by minimizing the following equations:

$$\frac{\partial RSS}{\partial \beta_0} = 0, \quad \frac{\partial RSS}{\partial \beta_1} = 0.$$

The LS estimators are often referred to as *Best Linear Unbiased Estimators* ($BLUEs$), since the LS estimators have minimum variance among all linear unbiased estimators [5].

There can be more than one regressor variable, let us say k variables, then, we use *Multiple Linear Regression* (MLR) model. In MLR, the data looks like as in Table 2.1 [63]:

Table 2.1: Data for Multiple Linear Regression

y	x_1	x_2	$\ldots$	x_k
y_1	x_{11}	x_{12}	$\ldots$	x_{1k}
y_2	x_{21}	x_{22}	$\ldots$	x_{2k}
$\vdots$	$\vdots$	$\vdots$		$\vdots$
y_N	x_{N1}	x_{N2}	$\ldots$	x_{Nk}

The model can be written as:

$$y_i = \beta_0 + \sum_{j=1}^{k} \beta_j x_{ij} + \varepsilon_i \quad \text{for } i = 1, 2, \ldots, N,$$

where errors are assumed to be uncorrelated random variables with $E(\varepsilon_i) = 0$ and $Var(\varepsilon_i) = \sigma^2$. In MLR, RSS is as follows:

$$RSS(\boldsymbol{\beta}) = \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{k} x_{ij}\beta_j \right)^2 .$$

Since there are N equations with $k+1$ unknown parameters and also it is a quadratic function of the parameters, it is more practical to write in matrix form [39]:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}. \tag{2.1}$$

Here, $\boldsymbol{y}$ is the $N \times 1$ response vector; $\boldsymbol{\beta}$ is the $(k+1) \times 1$ regression coefficients vector including the intercept; $\boldsymbol{\epsilon}$ is the $N \times 1$ random error vector; N is the number of observations in the data set, and $\boldsymbol{X}$ is the $N \times (k+1)$ independent variable (design) matrix, defined as follows by the input data $X_{i,j}$ $(i = 1, 2, \ldots, N; j = 1, 2, \ldots, k)$:

$$\boldsymbol{X} = \begin{bmatrix} 1 & X_{11} & \ldots & X_{1k} \\ 1 & X_{21} & \ldots & X_{2k} \\ \vdots & \vdots & & \vdots \\ 1 & X_{N1} & \ldots & X_{Nk} \end{bmatrix}.$$

Then, RSS can be represented as follows [39]:

$$RSS(\boldsymbol{\beta}) = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^T (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) = \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 . \tag{2.2}$$

Here, $\|.\|_2$ is the Euclidean norm.

Differentiating RSS with respect to $\boldsymbol{\beta}$ results in

$$\nabla RSS(\boldsymbol{\beta}) = -2\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}).$$

After setting the first derivative of RSS to zero, we get the *normal equations* [39]

$$\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{X}^T\boldsymbol{y}.$$

If $\boldsymbol{X}^T\boldsymbol{X}$ is nonsingular, the unique solution can be obtained and the fitted values are defined as

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}} = \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}.$$

However, if $\boldsymbol{X}^T\boldsymbol{X}$ is singular, then the *Singular Value Decomposition (SVD)* method is used to obtain solutions for the normal equations.

2.1.2 Maximum Likelihood Estimation Method

Least-squares estimation is a very convenient method, however, in some cases it does not make much sense. If the distribution of the errors is known, then MLE is an alternative estimation method. In fact, it is a more general approach and has better statistical properties than LSE [39]. For example, while *Least Square(LS)* estimators have minimum variance among only linear estimators, *Maximum Likelihood (ML)* estimators have minimum variance when compared to all other unbiased estimators, so this method is more efficient than LS method.

The likelihood of a set of data is the probability of obtaining that particular set of data, given the chosen probability distribution model. The values of the unknown parameters that maximize the sample likelihood are known as the Maximum Likelihood Estimates or MLE's [73].

In LS method, we do not need any distributional assumption, whereas in MLE we need to know the distribution. By assuming that random errors of data points are uncorrelated and normally distributed with variances σ_i^2 $(i = 1, 2, \ldots, N)$, we can derive the ML estimates of the equation (2.1). The probability density function for y_i $(i = 1, 2, \ldots, N)$ is as follows:

$$f(y_i|\boldsymbol{\beta}, \boldsymbol{\sigma}) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left[-\frac{1}{2\sigma_i^2}(y_i - E(y_i))^2\right], \tag{2.3}$$

where $\boldsymbol{\sigma}$ is a diagonal matrix with diagonal entries σ_1, σ_2,..., σ_N, that is assumed to be equal to a constant term, σ. As the likelihood function consists of the joint multiplications of each density function y_i, when $\sigma_i = \sigma$ $(i = 1, 2, \ldots, N)$, the likelihood function of the equation (2.3) looks like :

$$\begin{aligned} L(\boldsymbol{\beta}, \boldsymbol{\sigma}|\boldsymbol{y}) &= \prod_{i=1}^{N} f(y_i) \\ &= (2\pi\sigma^2)^{-N/2} \exp\left[\sum_{i=1}^{N}\left(\frac{-1}{2\sigma^2}(y_i - (\boldsymbol{X}\boldsymbol{\beta})_i)^2\right)\right]. \end{aligned}$$

It is more practical to take the logarithm of the likelihood function. Thus, the log-likelihood is:

$$\begin{aligned} \ln L &= -\frac{N}{2} \ln (2\pi\sigma^2) - \frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) \\ &= -\frac{N}{2} \ln (2\pi\sigma^2) - \frac{1}{2\sigma^2}\, RSS(\boldsymbol{\beta}), \end{aligned}$$

where $RSS(\boldsymbol{\beta})$ is same as in equation (2.2). Obviously, the first part consists of constant terms like N, π and σ, so we can ignore it. In the second part, on the other hand, RSS is not constant and to maximize the log-likelihood function, we need to minimize RSS with respect to $\boldsymbol{\beta}$. Then, it turns to the same least-square problem mentioned before. This means that the MLE method gives identical estimates with LSE when the errors are random and normally distributed [5, 39, 73].

When there is heteroscedasticity ($\sigma_i \neq \sigma_j$ for all $i \neq j$) among uncorrelated error terms that follow a multivariate normal distribution with a known covariance matrix, then we should also consider standard deviation σ_i in equation (2.4). Our new minimization problem looks like

$$\min_{\boldsymbol{\beta}} \quad \sum_{i=1}^{N} \frac{(y_i - (\boldsymbol{X}\boldsymbol{\beta})_i)^2}{\sigma_i^2}.$$

By using a diagonal weight matrix $\boldsymbol{W} := \text{diag}(1/\sigma_1, 1/\sigma_2, \ldots, 1/\sigma_N)$, the new system of equations is

$$\boldsymbol{y}_w = \boldsymbol{X}_w\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

where $\boldsymbol{X}_w := \boldsymbol{W}\boldsymbol{X}$ and $\boldsymbol{y}_w := \boldsymbol{W}\boldsymbol{y}$. If $\boldsymbol{X}_w^T\boldsymbol{X}_w$ is nonsingular, then the MLE of $\boldsymbol{\beta}$ for a weighted system is obtained by the following equation:

$$\hat{\boldsymbol{\beta}}^* = (\boldsymbol{X}_w^T\boldsymbol{X}_w)^{-1}\boldsymbol{X}_w^T\boldsymbol{y}_w.$$

Although both MLE and LSE methods provide parameter estimators that have many good properties, they are sensitive to the presence of outliers [73].

2.1.3 Nonlinear Regression Models

In real life, it is not always possible to see a linear relationship between variables. Sometimes the true relationship to be modeled may be curved, rather than a straight line or a flat plane. Then, to fit something like this, we need nonlinear regression models.

Nonlinear Estimation is a general fitting procedure that will estimate any kind of relationship between dependent and independent variables. The dependent variables are modeled as a nonlinear function of model parameters and one or more independent

variables. In general, all regression models may be stated as:

$$Y = f(\boldsymbol{x}, \boldsymbol{\theta}) + \epsilon,$$

where $\boldsymbol{\theta}$ is a $(k \times 1)$-vector of unknown parameters $\boldsymbol{\theta}=(\theta_1, \theta_2, \ldots, \theta_k)^T$, ϵ is an uncorrelated random error term with variances σ_i^2 $(i = 1, 2, \ldots, N)$ and a zero of mean, $f(\boldsymbol{x};\boldsymbol{\theta})$ is the expectation function for the nonlinear regression model and $\boldsymbol{x} = (x_1, x_2, \ldots, x_k)^T$ is an input vector [66]. Entire equation can be comprised in vector notation by the following system:

$$\boldsymbol{y} = \boldsymbol{\eta}(\boldsymbol{\theta}) + \boldsymbol{\epsilon},$$

where $\boldsymbol{\eta}(\boldsymbol{\theta}) := (f(\boldsymbol{x}_1, \boldsymbol{\theta}), f(\boldsymbol{x}_2, \boldsymbol{\theta}), \ldots, f(\boldsymbol{x}_N, \boldsymbol{\theta}))^T$ and $\boldsymbol{\epsilon}$ is the vector of residual. There are many methods for nonlinear regression models: Nonlinear Regression methods, Maximum Likelihood Estimation method, the Gauss-Newton method and the Levenberg-Marquardt Method [104].

2.2 Generalized Linear Models

Generalized Linear Models (GLM) are used in many areas of prediction, in regression and classification as well. It makes it possible to flexibly look for linear and nonlinear relationships between a continuous, or binomial, multinomial categorical dependent variable and categorical or continuous predictor variables. This approach is used when the normality and constant variance assumptions are not satisfied [66].

A number of widely used types of analysis can be considered as special applications of generalized linear models, such as binomial and multinomial logit and prohibit regression models. In generalized linear models, the mean value of a dependent variable depends on a linear predictor through a nonlinear link function and allows the response variable Y; its probability distribution to be any member of an exponential family of distributions which has the basic structure

$$\mu_i = h(\eta_i) = h(\boldsymbol{X}_i^T \boldsymbol{\beta}), \qquad \text{where} \quad \mu_i = E(Y_i), \text{ for } i = 1, 2, \ldots, N. \tag{2.4}$$

Here, N is the number of data, h denotes the smooth link function, $\boldsymbol{X}_i^T$ is the ith row of the model matrix $\boldsymbol{X}$ and $\boldsymbol{\beta}$ is the vector of unknown parameters.

A GLM usually makes the distribution assumptions that the response variable is independent and can have any distribution from an exponential density family. It has the following form [103]:

$$f_\theta(y) = \exp\left(\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right), \tag{2.5}$$

where b, a, c are arbitrary functions, ϕ is an arbitrary, so-called *scale parameter* and θ is known as the *canonical parameter* of the distribution.

Many widely used statistical models are belonging to GLMs. For example: classical linear models with normal errors, logistic and prohibit models for binary data, log-linear models for multinomial data, Poisson, Binomial, Gamma and Normal Distribution, etc.. These can be formulated as a GLM by selecting an appropriate link function and a response probability distribution. If the identity function is chosen as the link along with the normal distribution, then ordinary linear models are recovered as a special case.

2.2.1 Properties of the Exponential Family Distributions

Before finding the mean and variance of Y in θ, we give the following to properties:

- $E(\frac{\partial^2}{\partial\theta^2} l(y,\theta,\phi)) = 0, \quad$ where $\quad l(y,\theta,\phi) := \log(f(y,\theta,\phi))$,
- $E(\frac{\partial^2}{\partial\theta^2} l(y,\theta,\phi)) = -E(\frac{\partial}{\partial\theta} l(y,\theta,\phi))^2$.

Both statements follow from the well-known result that the integral of a probability function is always equal to one over the whole range:

$$\int f(y,\theta,\phi)dy = 1.$$

The first property can be derived by taking the derivative with respect to θ

$$\int \frac{\partial f(y,\theta,\phi)}{\partial\theta} dy = 0, \int \left(\frac{\partial \log(f(y,\theta,\phi))}{\partial\theta}\right) f(y,\theta,\phi)dy = 0.$$

The right-hand sight correspond the expectation of $\frac{\partial \log(f(y,\theta,\phi)}{\partial\theta}$. Thus,

$$E\left(\frac{\partial \log(f(y,\theta,\phi)}{\partial\theta}\right) = 0. \tag{2.6}$$

The second property is obtained taking the second derivative with respect to θ:

$$\frac{\partial \int (\frac{\partial \log(f(y,\theta,\phi))}{\partial \theta}) f(y,\theta,\phi) dy}{\partial \theta} = 0,$$

$$\int \frac{\partial^2 l(y,\theta,\phi) f(y,\theta,\phi) dy}{\partial \theta^2} + \int \frac{\partial l(y,\theta,\phi)}{\partial \theta} \frac{\partial f}{\partial \theta} dy = 0,$$

$$E\left(\frac{\partial^2 l(y,\theta,\phi)}{\partial \theta^2}\right) = -E\left((\frac{\partial l(y,\theta,\phi)}{\partial \theta})^2\right). \tag{2.7}$$

It is easy to find the mean and variance of Y by means of θ. From the form of the exponential density (2.5), it follows that

$$l(y,\theta,\phi) = \frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi). \tag{2.8}$$

By taking the derivative and using the expectation of (2.8), we get:

$$E\left(\frac{\partial l(y,\theta,\phi)}{\partial \theta}\right) = \frac{E(y) - b'(\theta)}{a(\phi)}.$$

The left-hand side is zero by (2.6). Hence,

$$E(y) = b'(\theta) = \mu. \tag{2.9}$$

The variance can be found by taking one more derivative:

$$\frac{\partial^2 l(y,\theta,\phi)}{\partial \theta^2} = -\frac{b''(\theta)}{a(\phi)}.$$

From (2.7), we obtain

$$E\left(\frac{\partial^2 l(y,\theta,\phi)}{\partial \theta^2}\right) = -E\left(\frac{\partial (l(y,\theta,\phi))}{\partial \theta}\right)^2 = E\left(-\frac{b''(\theta)}{a(\phi)}\right).$$

By evaluating the derivative of (2.8), we get:

$$-E(\frac{y - b'(\theta)}{a(\phi)})^2 = -\frac{b''(\theta)}{a(\phi)}, \; -\frac{E(y-\mu)^2}{a^2(\phi)} = -\frac{b''(\theta)}{a(\phi)},$$

$$Var(y) = b''(\theta)a(\phi). \tag{2.10}$$

This form covers all the cases of practical interest here. For example, it allows the possibility of unequal variances in models based on the normal distribution.

2.2.2 Estimation

While the maximization of likelihood turns out to need an iterative least-squares approach, the estimation and inference for GLM are based on the theory of MLE. Even though the estimation needs a numerical approximation, each step of the iteration can be given by a weighted least-squares fit. Since the weights are varying during the iteration, the likelihood is optimized by an *iteratively reweighted least squares algorithm IRLS*.

2.2.3 Maximum Likelihood and Deviance Minimization

As stated before, Y is a vector of N response variables denoted by $\boldsymbol{Y} = (Y_1, Y_2, \ldots, Y_N)^T$ and member of exponential family distribution with canonical parameter θ_i, which is determined by μ_i (via equation (2.8)) and, hence, by $\boldsymbol{\beta}$ ultimately. Given a vector $\boldsymbol{Y}$, maximum likelihood estimation of $\boldsymbol{\beta}$ is possible.

The sample log-likelihood of the vector $\boldsymbol{Y}$ is

$$\ell(\boldsymbol{Y}, \boldsymbol{\mu}, \boldsymbol{\phi}) := \sum_{i=1}^{N} \ell(Y_i, \theta_i, \phi), \qquad \text{where} \quad \boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_N)^T, \tag{2.11}$$

with θ_i is a function of $\eta_i = \boldsymbol{X}_i^T \boldsymbol{\beta}$ and $\ell(Y_i, \theta_i, \phi) = \log(Y_i, \theta_i, \phi)$.

Although the estimation method of choice for $\boldsymbol{\beta}$ is maximum-likelihood, there exists an alternative method, the so-called *Minimization of Deviance* [64]. The *scaled deviance* is defined as follows:

$$D(\boldsymbol{Y}, \boldsymbol{\mu}, \boldsymbol{\phi}) := 2\ell(\boldsymbol{Y}, \boldsymbol{\mu}^{max}, \boldsymbol{\phi}) - \ell(\boldsymbol{Y}, \boldsymbol{\mu}, \boldsymbol{\phi}).$$

Here, $\boldsymbol{\mu}^{max}$ is the vector that maximizes the saturated model. Since the term $\ell(\boldsymbol{Y}, \boldsymbol{\mu}^{max}, \boldsymbol{\phi})$ does not depend on $\boldsymbol{\beta}$, the minimization of the scaled deviance is equivalent to the maximization of the sample log-likelihood (2.11).

The *non-scaled deviance* is shown with the following equation [64];

$$D(\boldsymbol{Y}, \boldsymbol{\mu}) = D(\boldsymbol{Y}, \boldsymbol{\mu}, \boldsymbol{\phi}) a(\boldsymbol{\phi}). \tag{2.12}$$

The non-scaled deviance $D(\boldsymbol{Y}, \boldsymbol{\mu})$ can be thought as the GLM equivalent of the residual sum of squares (RSS) in linear regression, since it compares the log-likelihood ℓ for the model $\boldsymbol{\mu}$ with the maximal achievable value of ℓ [64].

The maximum likelihood representation of (2.12) can be found by using (2.8) in equation (2.11). Thus,

$$\ell(Y, \boldsymbol{\mu}, \boldsymbol{\phi}) = \sum_{i=1}^{N} (\frac{Y_i\theta_i - b(\theta_i)}{a(\phi)} - c(Y_i, \phi)). \tag{2.13}$$

As seen before, neither $a(\phi)$ nor $c(Y_i, \phi)$ depends on the unknown parameter vector $\boldsymbol{\beta}$ (through $\boldsymbol{\theta}$). Therefore, it is sufficient to consider

$$\sum_{i=1}^{N} (Y_i\theta_i - b(\theta_i)) \tag{2.14}$$

for the maximization. By taking derivative of (2.14) and denoting it as the gradient

$$\nabla(\boldsymbol{\beta}) = \frac{\partial}{\partial \boldsymbol{\beta}}[-2\sum_{i=1}^{N}(Y_i\theta_i - b(\theta_i))] = -2\sum_{i=1}^{N}(Y_i - b'(\theta_i))\frac{\partial}{\partial \boldsymbol{\beta}}\theta_i,$$

our optimization problem becomes $\nabla(\boldsymbol{\beta})$=0. This is a nonlinear system of equations in $\boldsymbol{\beta}$ and an iterative solution has to be computed.

2.2.4 Iteratively Re-Weighted Least Squares Algorithms

Iteratively Re-weighted Least Squares Algorithm ($IRLS$) is a method to find the maximum likelihood estimates of a generalized linear model.

Two well-known iterative maximum likelihood algorithms are *Fisher-scoring* and *Newton-Raphson*. Both algorithms give the same parameter estimates; however, the estimated covariance matrix of the parameter estimators may differ slightly. This is due to the fact that the Fisher-scoring method is based on the expected information matrix while the Newton-Raphson method is based on the observed information matrix. In the case of a binary logit model, the observed and expected information matrices are identical, resulting in identical estimated covariance matrices for both algorithms.

In our optimization problem, the smoothness of the link function allows us to compute the *Hessian* of $D(Y,\boldsymbol{\mu})$, denoted by $\boldsymbol{H}(\boldsymbol{\beta})$, so that Newton-Raphson algorithm can be applied using the following iteration steps [64]:

$$\hat{\boldsymbol{\beta}}^{new} = \hat{\boldsymbol{\beta}}^{old} - (H(\hat{\boldsymbol{\beta}}^{old}))^{-1}\nabla(\hat{\boldsymbol{\beta}}^{old}).$$

By replacing the Hessian by its expectation, it turns out to be the *Fisher scoring algorithm* [64]:

$$\hat{\boldsymbol{\beta}}^{new} = \hat{\boldsymbol{\beta}}^{old} - (EH(\hat{\boldsymbol{\beta}}^{old}))^{-1}\nabla(\hat{\boldsymbol{\beta}}^{old}).$$

For these iterations, there exists some simple representations. We have the following equation (from (2.4) and (2.9)):

$$\mu_i = h(\eta_i) = h(\boldsymbol{X_i^T}\boldsymbol{\beta}) = b'(\theta_i).$$

By taking the derivative of the right-hand term with respect to $\boldsymbol{\beta}$, we get:

$$\begin{aligned} h'(\boldsymbol{X}_i^T\boldsymbol{\beta})\boldsymbol{X}_i &= b''(\theta_i)\frac{\partial}{\partial\boldsymbol{\beta}}\theta_i, \\ \frac{\partial\theta_i}{\partial\boldsymbol{\beta}} &= \frac{h'(\eta_i)}{V(\mu_i)}\boldsymbol{X}_i, \end{aligned}$$

where $V(\mu_i) = b''(\theta_i)$. Now, one more derivative is taken:

$$\frac{\partial^2}{\partial\boldsymbol{\beta}\boldsymbol{\beta}^T}\theta_i = \frac{h''(\eta_i)V(\mu_i) - h'(\eta_i)^2V'(\mu_i)}{V(\mu_i)^2}\boldsymbol{X}_i\boldsymbol{X}_i^T.$$

Hence, the gradient and the Hessian of the deviance can be expressed by [64]:

$$\begin{aligned} \nabla(\boldsymbol{\beta}) &= -2\sum_{i=1}^{N}(Y_i - \mu_i)\frac{h'(\eta_i)}{V(\mu_i)}\boldsymbol{X}_i, \\ H(\boldsymbol{\beta}) &= 2\sum_{i=1}^{N}(\frac{h'(\eta_i)^2}{V(\mu_i)} - (Y_i - \mu_i)\frac{h''(\eta_i)V(\mu_i) - h'(\eta_i)^2V'(\mu_i)}{V(\mu_i)^2})\boldsymbol{X_i}\boldsymbol{X_i}^T. \end{aligned}$$

The expectation of $H(\boldsymbol{\beta})$ in the Fisher scoring algorithm equals

$$EH(\boldsymbol{\beta}) = 2\sum_{i=1}^{N}(\frac{h'(\eta_i)^2}{V(\mu_i)})\boldsymbol{X}_i\boldsymbol{X}_i^T.$$

Then, the *Fisher Scoring iteration* step for $\boldsymbol{\beta}$ can be expressed with the following formula [64]:

$$\boldsymbol{\beta}^{new} = \boldsymbol{\beta}^{old} + (\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}^T\widetilde{\boldsymbol{Y}} = (\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{Z},$$

with the weight matrix is $\boldsymbol{W} := \mathrm{diag}(\frac{h'(\eta_1)^2}{V(\mu_1)}, \ldots, \frac{h'(\eta_n)^2}{V(\mu_n)})$ and with the vectors $\widetilde{\boldsymbol{Y}} = (\widetilde{Y}_1, \ldots, \widetilde{Y}_n)^T$, $\boldsymbol{Z} = (Z_1, \ldots, Z_n)^T$, by $\widetilde{Y}_i = \frac{Y_i - \mu_i}{h'(\eta_i)}$ and $Z_i = \eta_i + \widetilde{Y}_i = \boldsymbol{X}_i{}^T\boldsymbol{\beta}^{old} + \frac{Y_i - \mu_i}{h'(\eta_i)}$ $(i = 1, 2, \ldots, N)$.

Since the weights are recalculated in each step, it is called as the *iteratively reweighted least squares* $(IRLS)$ algorithm. For the Newton-Raphson algorithm a representation equivalent to above can be found, only the weight matrix $\boldsymbol{W}$ is different in our case of the Fisher scoring iteration.

The iteration will be stopped when the parameter estimate and/or the deviance do not change significantly anymore. Then, $\hat{\boldsymbol{\beta}}$ is the final parameter of the iteration process.

2.3 Generalized Partial Linear Models

A *Generalized Partial Linear Model* (*GPLM*) extends the GLM in that the usual parametric terms are augmented by a single nonparametric component.

2.3.1 Introduction

The GPLM model is given by [92]

$$E(Y|\boldsymbol{X}, \boldsymbol{T}) = G(\boldsymbol{X}^T\boldsymbol{\beta} + \gamma(\boldsymbol{T})), \tag{2.15}$$

where $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_m)^T$ is a finite dimensional parameter and $\gamma(\cdot)$ is a smooth function which we try to estimate by *B-splines*. Here, $\boldsymbol{X}$ denotes an m-variable random vector which typically covers discrete covariables, and $\boldsymbol{T}$ is a q-variate random vector of continuous covariables to be modeled in a nonparametric way.

Straightforward maximization of the log-likelihood function L, which is written in the composite form $L(\theta(\boldsymbol{\beta}, \gamma))$ to emphasize the roles of predictors, parameters, and of the unknown curve is no longer appropriate as a method of estimation. This leads to overfitting in the absence of any constraints on $\boldsymbol{\beta}$. Indeed, it typically renders the parameters $\boldsymbol{\beta}$ unidentifiable. But progress is possible by maximizing instead a penalized version of log-likelihood, if we are willing to place weak constraints on the form of $\boldsymbol{\gamma}$ by assuming that it is smooth. Thus, we maximize the penalized log-likelihood [92]

$$\ell(\eta, y) := L(\theta(\boldsymbol{\beta}, \gamma) - \frac{1}{2}\tau \int_a^b (\gamma''(t))^2 dt,$$

where $H(\boldsymbol{\mu}) := \eta(\boldsymbol{X}, \boldsymbol{T}) = \boldsymbol{X}^T\boldsymbol{\beta} + \gamma(\boldsymbol{T})$, and $G := H^{-1}$ is a *link function* which links the mean of the response variable to the predictors.

Here, ℓ represents the log-likelihood of the linear predictor and the second term is the penalizing part, and τ is a smoothing parameter that controls the trade-off between accuracy of the data fitting and its smoothness (or complexity) [14]. By smoothing, it is desired to guarantee that the estimation is sufficiently robust with respect to noise in data and other forms of perturbation [92].

2.3.2 The Mathematical Tool of Splines

Models that closely fit the data is preferable in any regression procedure. Transformations of the response variable is a method to improve the fit and may help to fix violations of model assumptions such as constant error variance. Also a predictor variable can be divided into logical categories (e.g., weight categories), or additional terms that are functions of the existing predictors such as quadratic or cubic terms can be added. Nonetheless, methods such as spline modeling, taking into consideration the variation in the relationship between the predictor variable and the response variable, may provide a better fit both within and between levels of the predictor variable. Still no one is the best approach, as some modeling methods may produce better results for predicted values (e.g., narrower confidence intervals) than other methods, depending on the data. Greenland (1995) [28], indicating that categorical analysis does not make use of within category information and is based on an unrealistic model for dose-response and trends, propose to use spline regression (and fractional polynomial regression) as an alternative method to categorical analysis for dose response and trend analysis. Spline regression is based on more realistic category-specific models that are especially beneficial when subjected to nonlinearity [45].

Splines either line or curve are usually required to be continuous and smooth. Univariate polynomial splines are piecewise polynomials in one variable of some degree k with function values and the first k-1 derivatives that agree at the points where they join. These points that mark one transition to the next are referred to as break points, interior knots, or simply knots [23, 80]. Knots provide the curve freedom to turn as well as follow the data more closely. Although splines with few knots are generally smoother than splines with many knots, the fit of the spline function to the data increases by allowing more knots [32]. For any given set of knots, the smooth spline is computed by multiple regression on an appropriate set of basis elements, or basis functions representing the particular family of piecewise polynomials. The truncated-power series basis is a simple choice of basis functions for piecewise splines [92]. Although conceptually simple, truncated power series are not attractive numerically, because they can allow big rounding problems. Despite there are many types of splines and estimation procedures [23, 30], in this thesis we will focus on GPLM by using B-splines. B-spline

bases, on the other hand, allow for efficient elegant computations even when there is a huge number of knots [8].

2.3.2.1 B-Splines

The term B-spline was introduced by Isaac Jacob Schoenberg and is the short form of basis spline. B-spline functions have a minimal support regarding over a given degree, smoothness, and domain partition. According to a fundamental theorem, every spline function of a given degree, smoothness, and domain partition, can be outlined as a linear combination of B-splines of that same degree and smoothness, and over that same partition [8].

B-splines consist of polynomial pieces having a special connection among pieces. In a B-spline, each control point is connected with a basis function. The curve is [92]

$$\gamma(t) := \sum_{j=1}^{r} \lambda_j B_{j,k}(t) \qquad (t \in [a, b]),$$

where $\lambda_1, \lambda_2, \ldots, \lambda_r$ are r control parameters, $B_{i,k}(t)$ are basis functions of degree k, $\boldsymbol{t}= (t_1, t_2, \ldots, t_q)^T$ is a knot vector with $a \leq t_j < t_{j+1} \leq b$, and must be specified by $k = q - r - 1$. This determines the values of t at which the pieces of the curve join.

Let us note some important examples:

- *Zero-Degree* B-spline:

$$B_{j,0}(t) = \begin{cases} 1, & t_j \leq t \leq t_{j+1}, \\ 0, & \text{otherwise}, \end{cases} \qquad \text{for} \quad j = 1, 2, \ldots, q;$$

- *k-degree* B-spline [92]:

$$B_{j,k}(t) = \frac{t - t_j}{t_{j+k} - t_j} B_{j,k-1}(t) - \frac{t_{j+k+1} - t}{t_{j+k+1} - t_{j+1}} B_{j+1,k-1}(t) \qquad (k \geq 1);$$

for $k \geq 2$, its derivative is

$$\frac{d}{dx} B_{j,k}(t) = \frac{k}{t_{j+k} - t_j} B_{j,k-1}(t) + \frac{k}{t_{j+k+1} - t_{j+1}} B_{j+1,k-1}(t).$$

B-spline bases overlap with each other. For example, first-degree B-spline bases overlap with two neighbors, second-degree B-spline bases with four-degree ones, and this continues like this.

Some properties of a B-spline are [92]:

- it consists of $k+1$ polynomial pieces, each of degree k;
- the polynomial pieces join at k inner knots;
- at the joining points, derivatives up to order $k-1$ coincide;
- a B-spline basis function is positive on a domain spanned by $k+2$ knots; outside, it is zero;
- except at boundaries, it overlaps with $2k$ polynomial pieces of its neighbours;
- at a given point t, $k+1$ B-splines basis functions are nonzero.

2.3.3 Estimation Methods

Although the maximization of likelihood turns out to require an iterative least-squares approach, estimation and inference for GLMs are based on the theory of maximum likelihood estimation. A particular semiparametric model of interest is the generalized partial linear model (GPLM) which extends the generalized linear models in that the usual parametric terms are augmented by a single nonparametric component. Generally, the estimation methods for GPLM are based on the idea that an estimate of $\hat{\boldsymbol{\beta}}$ can be found for a known $\boldsymbol{\gamma}(\cdot)$ and an estimate of $\hat{\boldsymbol{\gamma}}(\cdot)$ can be found for a known $\boldsymbol{\beta}$. In this thesis, we will focus on different types of estimation of $\boldsymbol{\gamma}(\cdot)$ and $\boldsymbol{\beta}$ based on B-splines.

2.3.3.1 Penalized Maximum Likelihood

Let us consider the GPLM model (2.15) in the introduction part, where it is assumed that $G = H^{-1}$ is a link function. Here, however, the model can be thought as semi-parametric GLM since all terms are linear except one; i.e.,

$$H(\boldsymbol{\mu}) = \eta(\boldsymbol{X}, \boldsymbol{T}) = \boldsymbol{X}^T\boldsymbol{\beta} + \gamma(\boldsymbol{T}) = \sum_{j=1}^{m} X_j\boldsymbol{\beta}_j + \gamma(\boldsymbol{T}) \qquad (i = 1, 2, \ldots, N). \quad (2.16)$$

For simplicity, the observation values t_i of $\boldsymbol{T}$ in GPLM are considered one-dimensional. Then, $\mu_i = G(\eta_i)$ and

$$\eta_i = H(\mu_i) = \boldsymbol{X}_i^T\boldsymbol{\beta} + \gamma(t_i). \quad (2.17)$$

We will use penalized maximum likelihood estimation to avoid overfitting. This method is characterized through a score function $\partial\ell(\eta, y)/\partial\eta$. For this model the penalized maximum criterion is given by [92]:

$$j(\boldsymbol{\beta}, \gamma) = \ell(\eta, y) - \frac{1}{2}\tau \int_a^b (\gamma''(t))^2 dt. \tag{2.18}$$

As we estimate the model by penalized maximum likelihood, we desire to maximize (2.18) and for this we minimize the second part. We will do it by using B-splines through the local scoring algorithm, so we write a k degree B-spline with knots at the value t_i $(i = 1, 2, \ldots, N)$ instead of $\gamma(t)$. There will be $N - 2$ interior points and $N + k - 1$ unknown parameters.

Hence, we reach a representation

$$\gamma(t) := \sum_{j=1}^{v} \lambda_j B_{j,k}(t),$$

where λ_j are coefficients, $B_{j,k} = B_j$ are B-spline basis functions and $\nu{=}N + k - 1$. The vector notation is as follows:

$$\boldsymbol{\gamma(t)} = \boldsymbol{B\lambda},$$

where $\boldsymbol{\gamma(t)}{:=}\ (\gamma(t_1), \ldots, \gamma(t_N))^T$ and $\boldsymbol{B}{=}\ (B_{ij})_{\substack{i=1,2,\ldots,N \\ j=1,2,\ldots,\nu}}$ is a $(N{\times}\nu)$-matrix of $B_{ij} := B_j(t_i)$, and $\boldsymbol{\lambda}{=}\ (\lambda_1, \lambda_2, \ldots, \lambda_\nu)^T$.

If we define a $(\nu{\times}\nu)$-matrix $\boldsymbol{K}{=}\ (K_{kl})_{k,l=1,2,\ldots,N}$ matrix by $K_{kl} := \int_a^b B_k''(t)B_l''(t)dt$, then the penalized maximum criterion (2.18) can be written as

$$j(\boldsymbol{\beta}, \boldsymbol{\gamma}) := l(\boldsymbol{\eta}, \boldsymbol{y}) - \frac{1}{2}\tau \boldsymbol{\lambda}^T \boldsymbol{K} \boldsymbol{\lambda}. \tag{2.19}$$

By assuming $N \geq \nu$ and that $\boldsymbol{B}$ has full rank, we insert the least-squares estimation $\boldsymbol{\lambda} = (\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T\boldsymbol{\gamma(t)}$ into equation (2.19) and write $\boldsymbol{M}{:=}\boldsymbol{B}(\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{K}(\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T$, then we get

$$j(\boldsymbol{\beta}, \boldsymbol{\gamma}) = l(\boldsymbol{\eta}, \boldsymbol{y}) - \frac{1}{2}\tau \boldsymbol{\gamma}^T \boldsymbol{M} \boldsymbol{\gamma}. \tag{2.20}$$

Now, to solve the minimization problem of (2.20), we need to find the optimal estimators $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\gamma}}$. Let us denote $\boldsymbol{g}_1 {:=}\boldsymbol{X}\ \boldsymbol{\beta}$ and $\boldsymbol{g}_2 {:=}\boldsymbol{\gamma(t)}$; then (2.17) will be

$$H(\boldsymbol{\mu}) = \eta(\boldsymbol{X}, \boldsymbol{t}) = \boldsymbol{g_1} + \boldsymbol{g_2},$$

where $\boldsymbol{X}$ is an $(N\times m)$-matrix, and $\boldsymbol{g_1}$ and $\boldsymbol{g_2}$ are N-vectors of entries $\boldsymbol{X}_i^T\boldsymbol{\beta}$ and $\gamma(t_i)$, respectively. The following system of equations should be solved to maximize (2.18) over $\boldsymbol{g_1}$ and $\boldsymbol{g_2}$:

$$\begin{aligned}\frac{\partial j(\boldsymbol{\beta},\boldsymbol{\gamma})}{\partial \boldsymbol{g_1}} &= (\frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{g_1}})^T \frac{\partial \ell(\boldsymbol{\eta},\boldsymbol{y})}{\partial \boldsymbol{\eta}} = 0,\\ \frac{\partial j(\boldsymbol{\beta},\boldsymbol{\gamma})}{\partial \boldsymbol{g_2}} &= (\frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{g_2}})^T \frac{\partial \ell(\boldsymbol{\eta},\boldsymbol{y})}{\partial \boldsymbol{\eta}} - \tau \boldsymbol{M}\boldsymbol{g}_2 = 0.\end{aligned} \tag{2.21}$$

These system equations are nonlinear in $\boldsymbol{\eta}$ and $\boldsymbol{g_2}$. To reach a solution, they are linearized around a current guess $\boldsymbol{\eta^0}$ and obtain a Newton-Raphson type equation:

$$\frac{\partial \ell(\boldsymbol{\eta},\boldsymbol{y})}{\partial \boldsymbol{\eta}} \approx \frac{\partial \ell(\boldsymbol{\eta},\boldsymbol{y})}{\partial \boldsymbol{\eta}}\,|_{\boldsymbol{\eta}^0} + \frac{\partial^2 \ell(\boldsymbol{\eta},\boldsymbol{y})}{\partial \boldsymbol{\eta}\boldsymbol{\eta}^T}\,|_{\boldsymbol{\eta}^0}\,(\boldsymbol{\eta}-\boldsymbol{\eta^0}) = \boldsymbol{0}. \tag{2.22}$$

By using (2.22) in (2.21), and putting $\boldsymbol{r}{:=}\partial \ell(\boldsymbol{\eta},\boldsymbol{y})/\partial \boldsymbol{\eta}$ and $\boldsymbol{C}{:=}-\partial^2 \ell(\boldsymbol{\eta},\boldsymbol{y})/\partial \boldsymbol{\eta}\boldsymbol{\eta}^T$, we reach the following matrix notation:

$$\begin{pmatrix} \boldsymbol{C} & \boldsymbol{C} \\ \boldsymbol{C} & \boldsymbol{C}+\tau \boldsymbol{M} \end{pmatrix} \begin{pmatrix} \boldsymbol{g_1}^1 - \boldsymbol{g_1}^0 \\ \boldsymbol{g_2}^1 - \boldsymbol{g_2}^0 \end{pmatrix} = \begin{pmatrix} \boldsymbol{r} \\ \boldsymbol{r} - \tau \boldsymbol{M}\boldsymbol{g}_2^0 \end{pmatrix}, \tag{2.23}$$

where $(\boldsymbol{g_1^0},\boldsymbol{g_2^0}) \rightarrow (\boldsymbol{g_1^1},\boldsymbol{g_2^1})$ is a Newton-Raphson step, and $\boldsymbol{C}$ and $\boldsymbol{r}$ are evaluated at $\boldsymbol{\eta^0}$. To have a more simple form for the equation (2.23), let us put $\boldsymbol{h}{:=}\boldsymbol{\eta^0}{+}\boldsymbol{C^{-1}r}$, and $\boldsymbol{S}_B{:=}(\boldsymbol{C}{+}\tau\boldsymbol{M})^{-1}\boldsymbol{C}$ that is a weighted B-spline operator. Then, (2.23) takes the form

$$\begin{pmatrix} \boldsymbol{C} & \boldsymbol{C} \\ \boldsymbol{S}_B & \boldsymbol{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{g_1^1} \\ \boldsymbol{g_2^1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{C} \\ \boldsymbol{S}_B \end{pmatrix} \boldsymbol{h}. \tag{2.24}$$

If we multiply the upper row with $\boldsymbol{C^{-1}}$ and the second row with $(\boldsymbol{C}+\tau\boldsymbol{M})^{-1}$, we can transform it to

$$\begin{pmatrix} \boldsymbol{g_1^1} \\ \boldsymbol{g_2^1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{X}\boldsymbol{\beta^1} \\ \boldsymbol{\gamma^1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{h}-\boldsymbol{g_2^1} \\ \boldsymbol{S}_B(\boldsymbol{h}-\boldsymbol{g_1^1}) \end{pmatrix}. \tag{2.25}$$

Here, $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\gamma}}$ can be found explicitly with no iteration (inner loop backfitting); then,

$$\begin{aligned}\hat{\boldsymbol{g_1}} &= \boldsymbol{X}\hat{\boldsymbol{\beta}} = \boldsymbol{X}\boldsymbol{X}^T\boldsymbol{C}(\boldsymbol{I}-\boldsymbol{S}_B)\boldsymbol{X}^{-1}\boldsymbol{X^T}\boldsymbol{C}(\boldsymbol{I}-\boldsymbol{S}_B)\boldsymbol{h},\\ \hat{\boldsymbol{g_2}} &= \hat{\boldsymbol{\gamma}} = \boldsymbol{S}_B(\boldsymbol{h}-\boldsymbol{X}\hat{\boldsymbol{\beta}}),\end{aligned} \tag{2.26}$$

where $\boldsymbol{X}{=}(x_{ij})_{i=1,2,...,N;\ j=1,2,...,m}$ is the regression matrix for the values x_i and $\boldsymbol{h}$ is the adjusted dependent variable. Furthermore, $\boldsymbol{S}_B$ computes a weighted B-spline smoothing on the variable t_i with weights given by $\boldsymbol{C}{=}-\partial^2\ell\ (\boldsymbol{\eta},\boldsymbol{y})/\partial\ \boldsymbol{\eta}\boldsymbol{\eta^T}$.

Newton-Raphson updates solve a weighted and penalized quadratic criterion. This criterion is a local approximation of the penalized log-likelihood. From the updated

$(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\gamma}})$, the outer loop must be iterated to update $\boldsymbol{\eta}$ and, thus, $\boldsymbol{h}$ and $\boldsymbol{C}$. Then, the loop is repeated until convergence is sufficient [27]. As the outer loop is simply a Newton-Raphson step, a step size optimization is performed, and the outer loop will convergence. Let us consider a trial value of the form

$$\boldsymbol{\eta}^{\phi} := \boldsymbol{\phi}\boldsymbol{\eta}^{1} + (1 - \boldsymbol{\phi})\boldsymbol{\eta}^{0}, \tag{2.27}$$

with $\boldsymbol{g_s}$ $(s = 1, 2)$ defined. Thus, (2.27) becomes a Newton-Raphson step of size ϕ and we maximize $j(\boldsymbol{\eta}^{(\phi)})$ over ϕ [92]. Convergence is ensured by the standard results on the Newton-Raphson procedure [74].

The asymptotic properties of these models can be found in [27, 38]. By considering the equations (2.26), we obtain

$$E(\hat{\boldsymbol{\beta}}) = \boldsymbol{\beta} + \boldsymbol{X}^T\boldsymbol{C}(\boldsymbol{I} - \boldsymbol{S}_B)\boldsymbol{X}\}^{-1}\boldsymbol{X}^T\boldsymbol{C}(\boldsymbol{I} - \boldsymbol{S}_B)\boldsymbol{B}\boldsymbol{\lambda},$$
$$Cov(\hat{\boldsymbol{\beta}}) = (\boldsymbol{X}^T\boldsymbol{C}(\boldsymbol{I} - \boldsymbol{S}_B)\boldsymbol{X}\}^{-1}\boldsymbol{X}^T\boldsymbol{C}(\boldsymbol{I} - \boldsymbol{S}_B)^2\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{C}(\boldsymbol{I} - \boldsymbol{S}_B)\boldsymbol{X}\}^{-1},$$

where $\{\boldsymbol{X^T C}(\boldsymbol{I}\text{-}\boldsymbol{S}_B)\boldsymbol{X}\}^{-1}\boldsymbol{X}^T\boldsymbol{C}(\boldsymbol{I}\text{-}\boldsymbol{S}_B)\boldsymbol{B}\boldsymbol{\lambda}$ is the estimated correction term.

Besides, considering the equations (2.25)-(2.26), the functions $\boldsymbol{g_1}$ and $\boldsymbol{g_2}$ are estimated by linear mapping or the smoother applied to the adjusted dependent variable $\boldsymbol{h}$, with weight $\boldsymbol{C}$ given by the information matrix. When $\boldsymbol{R}_B$ is the weighted additive fit operator, then, by convergence,

$$\hat{\boldsymbol{\eta}} = \boldsymbol{R}_B(\hat{\boldsymbol{\eta}} + \boldsymbol{C}^{-1}\hat{\boldsymbol{r}}) = \boldsymbol{R}_B\boldsymbol{h},$$

where $\hat{\boldsymbol{r}}$=$\partial\ell(\boldsymbol{\eta},\boldsymbol{y})/\partial\boldsymbol{\eta}|_{\hat{\boldsymbol{\eta}}}$ [92]. By changing from $\boldsymbol{h}$, $\boldsymbol{R}_B$ and $\boldsymbol{C}$ to their asymptotic versions $\boldsymbol{h_0}$, $\boldsymbol{R}_{B_0}$ and $\boldsymbol{C_0}$, where $\boldsymbol{h}$≈$\boldsymbol{h_0}$ has mean $\boldsymbol{\eta^0}$ and variance $\boldsymbol{C_0^{-1}}\phi \approx \boldsymbol{C^{-1}}\phi$. Then,

$$\begin{aligned} Cov(\hat{\boldsymbol{\eta}}) &\approx \boldsymbol{R}_{B_0}\boldsymbol{C_0^{-1}}\boldsymbol{R}_{B_0}^T\phi \\ &\approx \boldsymbol{R}_B\boldsymbol{C^{-1}}\boldsymbol{R}_B^T\phi, \end{aligned}$$

and

$$Cov(\hat{\boldsymbol{g_s}}) \approx \boldsymbol{R}_{B_s}\boldsymbol{C^{-1}}\boldsymbol{R}_{B_s}^T\phi \qquad (s = 1, 2).$$

Here, $\boldsymbol{R}_{B_j}$ is the matrix producing $\hat{g}_j$ from $\boldsymbol{h}$ based on B-splines. Besides, $\hat{\boldsymbol{\eta}}$ is asymptotically distributed as $N(\boldsymbol{\eta}_0, \boldsymbol{R}_{B_0}\boldsymbol{C}_0^{-1}\ \boldsymbol{R}_{B_0}^T\phi)$ [38].

2.3.3.2 Penalized Iteratively Re-Weighted Least Squares

The penalized likelihood is maximized by the *penalized iteratively reweighted least squares* ($P-IRLS$) method. By denoting $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\gamma}}$ as the estimated parameter vectors of $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, and $\eta_i^{[p]}=\boldsymbol{X}_i^T\hat{\boldsymbol{\beta}}+\hat{T}$, $\mu_i^{[p]}=H^{-1}(\eta_i^{[p]})$, respectively, where $G(\eta_i^{[p]})$ is the inverse function of the link at the pth iteration. Thus, we can express (2.24) as the linear system that finds $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$. Finally, we minimize the following equation to find the $(p+1)$th estimate of the linear predictor $\boldsymbol{\eta}^{[p+1]}$:

$$\|\boldsymbol{C}^{[p]}(\boldsymbol{h}^{[p]}-\boldsymbol{\eta})\|_2+\tau\boldsymbol{\gamma}^T\boldsymbol{M}\boldsymbol{\gamma}, \tag{2.28}$$

where $\|.\|_2$ is the Euclidean norm and $\boldsymbol{h}^{[p]}$ is the iteratively adjusted dependent variable. It is expressed by

$$h_i^{[p]}:=\eta_i^{[p]}+H'(\mu_i^{[p]})(y_i-\mu_i^{[p]}),$$

where H' is the first derivative of H with respect to $\boldsymbol{\beta}$ and $\boldsymbol{C}^{[p]}$ is a diagonal weight matrix with elements $C_{ii}^{[p]}:=1/V(\mu_i^{[p]})H'(\mu_i^{[p]})^2$, where $V(\mu_i^{[p]})$ is proportional to the variance of Y_i according to the current estimate $\mu_i^{[p]}$. By using $\boldsymbol{\gamma}(\boldsymbol{t})=\boldsymbol{B}\boldsymbol{\lambda}$ in (2.28), then it looks as follows:

$$\|\boldsymbol{C}^{[p]}(\boldsymbol{h}^{[p]}-\boldsymbol{X}\boldsymbol{\beta}-\boldsymbol{B}\boldsymbol{\lambda}\|_2+\tau\boldsymbol{\lambda}^T\boldsymbol{K}\boldsymbol{\lambda}. \tag{2.29}$$

Here we assume that $\boldsymbol{K}$ is of rank $z<v$ [27]. It is possible to write $\boldsymbol{J}^T\boldsymbol{K}\boldsymbol{J}=\boldsymbol{I}$, $\boldsymbol{T}^T\boldsymbol{K}\boldsymbol{T}=\boldsymbol{0}$ and $\boldsymbol{J}^T\boldsymbol{T}=\boldsymbol{0}$, where $\boldsymbol{J}$ and $\boldsymbol{T}$ are two matrices with ν rows and with full column ranks z and ν-z, respectively. Rewriting

$$\boldsymbol{\lambda}=\boldsymbol{T}\boldsymbol{\delta}+\boldsymbol{J}\boldsymbol{\varepsilon}, \tag{2.30}$$

with vectors $\boldsymbol{\delta}$ and $\boldsymbol{\varepsilon}$ of dimensions z and ν-z, respectively. Now, the term (2.28) becomes

$$||\boldsymbol{C}^{[p]}(\boldsymbol{h}^{[p]}-[\boldsymbol{X},\boldsymbol{\beta}\boldsymbol{T}]\begin{pmatrix}\boldsymbol{\beta}\\ \boldsymbol{\delta}\end{pmatrix}-\boldsymbol{B}\boldsymbol{J}\boldsymbol{\varepsilon})\|_2+\tau\boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon}.$$

We can split its minimization by separating to solution with respect to $\boldsymbol{\beta}$ and $\boldsymbol{\delta}$ from the one on $\boldsymbol{\varepsilon}$, by using *Householder decomposition* [20]. Then, we can write

$$\boldsymbol{Q}_1^T\boldsymbol{C}^{[p]}[\boldsymbol{X},\boldsymbol{B}\boldsymbol{T}]=\boldsymbol{R},\qquad \boldsymbol{Q}_2^T\boldsymbol{C}^{[p]}[\boldsymbol{X},\boldsymbol{B}\boldsymbol{T}]=\boldsymbol{0},$$

where $\boldsymbol{Q}$=[$\boldsymbol{Q}_1$,$\boldsymbol{Q}_2$] is orthogonal and $\boldsymbol{R}$ is nonsingular, upper triangular and of full rank $m+\nu-z$. Then, our problem turns to minimize the sum of

$$\|\boldsymbol{Q}_1^T\boldsymbol{C}^k\boldsymbol{h}^k - \boldsymbol{R}\begin{pmatrix}\boldsymbol{\beta}\\ \boldsymbol{\delta}\end{pmatrix} - \boldsymbol{Q}_1^T\boldsymbol{C}^k\boldsymbol{B}\boldsymbol{J}\boldsymbol{\varepsilon}\|_2 \tag{2.31}$$

with respect to ($\boldsymbol{\beta}$,$\boldsymbol{\delta}$), given $\boldsymbol{\varepsilon}$ based on minimizing

$$\|\boldsymbol{Q}_2^T\boldsymbol{C}^k\boldsymbol{h}^k - \boldsymbol{Q}_2^T\boldsymbol{C}^k\boldsymbol{B}\boldsymbol{J}\boldsymbol{\varepsilon}\|_2 + \tau\boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon}. \tag{2.32}$$

By an appropriate choice of $\boldsymbol{\beta}$ and $\boldsymbol{\delta}$, given $\boldsymbol{\varepsilon}$, the term (2.31) can be set to zero. If we take $\boldsymbol{H}$:=$\boldsymbol{Q}_2^T\boldsymbol{C}^k\boldsymbol{h}^k$ and $\boldsymbol{V}$:=$\boldsymbol{Q}_2^T\boldsymbol{C}^k\boldsymbol{B}\boldsymbol{J}$, (2.32) becomes the minimization problem

$$\|\boldsymbol{H} - \boldsymbol{V}\boldsymbol{\varepsilon}\|_2 + \tau\boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon},$$

that is a *Tikhonov regularization problem* [3]. The solution is

$$\widetilde{\boldsymbol{\varepsilon}} = (\boldsymbol{V}^T\boldsymbol{V} + \tau\boldsymbol{I})^{-1}\boldsymbol{V}^T\boldsymbol{H}.$$

We can find other parameters as

$$\begin{pmatrix}\widetilde{\boldsymbol{\beta}}\\ \widetilde{\boldsymbol{\delta}}\end{pmatrix} = \boldsymbol{R}^{-1}\boldsymbol{Q}_2^T\boldsymbol{C}^k(\boldsymbol{H} - \boldsymbol{B}\boldsymbol{J}\widetilde{\boldsymbol{\varepsilon}}).$$

The vector $\widetilde{\boldsymbol{\lambda}}$ can be computed from (2.30) and thus, $\boldsymbol{\eta}^{[p+1]}$=$\boldsymbol{X}\widetilde{\boldsymbol{\beta}}$+$\boldsymbol{B}\widetilde{\boldsymbol{\lambda}}$ can be computed. The matrices $\boldsymbol{J}$ and $\boldsymbol{T}$ can be computed via a *Cholesky and Householder transformation* [20].

2.3.3.3 An Alternative to the Choice for Penalty Parameters

Penalized maximum likelihood method and also P-IRLS methods both contain the smoothing parameter τ. To estimate this parameter, there are two commonly used methods; *Generalized Cross Validation* (GCV) and minimization of an *UnBiased Risk Estimator* ($UBRE$) [14]. However, here, we will mention an alternative method, called conic quadratic programming [92].

If we turn back to equation (2.29) and use *Cholesky Decomposition*, where $\boldsymbol{K}$ is a ($\nu\times\nu$)-matrix $\boldsymbol{K}$ such that $\boldsymbol{K}$=$\boldsymbol{U}^T\boldsymbol{U}$, then, the equation is:

$$\|\boldsymbol{W}\boldsymbol{\varphi} - \boldsymbol{v}\|_2 + \tau\|\boldsymbol{U}\boldsymbol{\lambda}\|_2^2. \tag{2.33}$$

Here, $\boldsymbol{\varphi}$:=$(\boldsymbol{\beta}^T,\boldsymbol{\lambda}^T)^T$, $\boldsymbol{W}$:=$\boldsymbol{C}^{[p]}(\boldsymbol{X},\boldsymbol{B})$ and $\boldsymbol{v}$:=$\boldsymbol{C}^{[p]}\boldsymbol{h}^{[p]}$.

Then, the problem (2.33) turns into an optimization problem with constraints:

$$\min \quad G(\boldsymbol{\varphi}) \quad \text{subject to} \quad g(\boldsymbol{\lambda}) \leq 0, \tag{2.34}$$

where $G(\boldsymbol{\varphi}) := \|\boldsymbol{W}\boldsymbol{\varphi} - \boldsymbol{v}\|_2$ and g($\boldsymbol{\lambda}$):= $\|\boldsymbol{U}\boldsymbol{\lambda}\|_2 - M$, and $M \geq 0$ which is chosen with some tolerance before or adapted in a learning process. Then, the optimization problem (2.34) can be equivalently written in the following form:

$$\begin{aligned} & \min \quad t, \\ \text{subject to} \quad & \|\boldsymbol{W}\boldsymbol{\varphi} - \boldsymbol{v}\|_2^2 \leq t^2, \\ & \|\boldsymbol{U}\boldsymbol{\lambda}\|_2 \leq M, \quad t \geq 0, \end{aligned}$$

where $\boldsymbol{W}$ and $\boldsymbol{V}$ are $(N\times(m+v))$- and $(v\times v)$-matrices, while $\boldsymbol{\varphi}$ and $\boldsymbol{v}$ are $(m+v)$- and n-vectors. Then, our optimization problem becomes:

$$\begin{aligned} & \min \quad t, \\ \text{subject to} \quad & \|\boldsymbol{W}\boldsymbol{\varphi} - \boldsymbol{v}\|_2 \leq t, \\ & \|\boldsymbol{U}\boldsymbol{\lambda}\|_2 \leq \sqrt{M}. \end{aligned} \tag{2.35}$$

By use of continuous optimization techniques, from conic quadratic optimization programming [68]:

$$\begin{aligned} & \min \quad \boldsymbol{c}^T\boldsymbol{x}, \\ \text{subject to} \quad & \|\boldsymbol{D}_i\boldsymbol{x} - \boldsymbol{d}_i\|_2 \leq \boldsymbol{p}_i^T\boldsymbol{x} - q_i \quad (i = 1, 2, \ldots, k). \end{aligned}$$

it can be seen that the minimization problem is a conic quadratic programming problem with

$$\boldsymbol{c} = (1, \boldsymbol{0}_{m+v}^T)^T, \quad \boldsymbol{x} = (t, \boldsymbol{\varphi}^T)^T = (t, \boldsymbol{\beta}^T, \boldsymbol{\lambda}^T)^T, \quad \boldsymbol{D}_1 = (\boldsymbol{0}_N, \boldsymbol{W}), \quad \boldsymbol{d}_1 = \boldsymbol{v},$$

$$\boldsymbol{p}_1 = (1, 0, \ldots, 0)^T, \quad q_1 = 0, \quad \boldsymbol{D}_2 = (\boldsymbol{0}_v, \boldsymbol{0}_{v\times m}, \boldsymbol{U}), \quad \boldsymbol{d}_2 = \boldsymbol{0}_v, \quad \boldsymbol{p}_2 = \boldsymbol{0}_{m+v+1}$$

and $q_2 = -\sqrt{M}$.

Equation (2.35) is reformulated for writing the dual problem to this problem and it

looks as follows:

$$\begin{aligned} & \min \quad t, \\ \text{subject to} \qquad & \boldsymbol{\psi} := \begin{pmatrix} \mathbf{0}_N & \boldsymbol{W} \\ 1 & \mathbf{0}_{m+v}^T \end{pmatrix} \begin{pmatrix} t \\ \boldsymbol{\varphi} \end{pmatrix} + \begin{pmatrix} -\boldsymbol{v} \\ 0 \end{pmatrix}, \\ & \boldsymbol{\rho} := \begin{pmatrix} \mathbf{0}_v & \mathbf{0}_{v\times m} & \boldsymbol{U} \\ 0 & \mathbf{0}_m^T & \mathbf{0}_v^T \end{pmatrix} \begin{pmatrix} t \\ \boldsymbol{\varphi} \end{pmatrix} + \begin{pmatrix} \mathbf{0}_v \\ \sqrt{M} \end{pmatrix}, \\ & \boldsymbol{\psi} \in L^{N+1}, \boldsymbol{\rho} \in L^{v+1}, \end{aligned}$$

where L^{N+1}, L^{v+1} are the $(N+1)$- and $(v+1)$-dimensional *ice-cream* (or *second-order*, or *Lorentz*) *cones*, defined by:

$$\begin{aligned} L^{l+1} := \{ \quad & \boldsymbol{x} = (x_1, x_2, \ldots, x_{l+1})^T \in \mathbb{R}^{l+1} \quad | \\ & x_{l+1} \geq \sqrt{x_1^2 + \ldots + x_l^2} \quad \} \qquad (l \geq 1). \end{aligned}$$

The dual problem to the latter problem is given by

$$\begin{aligned} & \max \quad (\boldsymbol{v}^T, 0)\boldsymbol{K}_1 + (\mathbf{0}_v^T, -\sqrt{M})\boldsymbol{K}_2 \\ \text{such that} \quad & \begin{pmatrix} \mathbf{0}_N^T & 1 \\ \boldsymbol{W}^T & \mathbf{0}_{m+v} \end{pmatrix} \boldsymbol{K}_1 + \begin{pmatrix} \mathbf{0}_v^T & 0 \\ \mathbf{0}_{m\times v} & \mathbf{0}_m \\ \boldsymbol{U}^T & \mathbf{0}_v \end{pmatrix} \boldsymbol{K}_2 = \begin{pmatrix} 1 \\ \mathbf{0}_{m+v} \end{pmatrix}, \\ & \boldsymbol{K}_1 \in L^{N+1}, \boldsymbol{K}_2 \in L^{v+1}. \end{aligned}$$

Classical polynomial time algorithms can be used to solve convex optimization problems such as semi-definite programming, geometric programming and, in particular, Conic Quadratic Problems. However, these algorithms use only local information on the objective function and have constraints. To solve "well-structured" convex problems like conic quadratic problems, *Interior Point Methods* [81, 70] firstly introduced by Karmarkar in 1984, are used. These methods (also called Barrier Methods) are based on both the given (primal) and the dual problem. They allow better complexity bounds and performs better practical performance. As well, they guarantee feasibility throughout the entire iteration procedures, while penalty methods and Tikhonov regularization can be regarded as *Exterior Point Methods* with possible infeasibility [92].

Until now, it is explained that a spline regression problem can be presented either as a Tikhonov regularization problem or as a conic quadratic problem. In the following

chapters, we will explain about both Tikhonov regularization and Conic quadratic problems which we connect with *multiple adaptive regression splines* ($MARS$) for our nonlinear arbitrary function $\gamma(t)$. This is called as adaptive because the selection of basis functions is data-based and specific to the problem at hand. By this connection, *conic multivariate adaptive regression splines* ($CMARS$) will be introduced.

2.3.4 Motivations and Applications

Generalized partial linear models (GPLMs) has a great advantage that consists in some *grouping* which could be done for the input dimensions or features in order to assign appropriate submodels specifically [92]. There are linear, nonlinear ones as well as parametrical and nonparametrical ones. By separating linear models from nonlinear or nonparametrical ones, inverse problem methods such as Tikhonov regularization [3] can be applied for the linear submodels separately, within the entire GPLMs. Such a particular representation of submodels provides both a better accuracy and a better stability (regularity) under noise in the data.

Among the real-word motivations which lead to GPLMs, there are the following ones, all of them related with important modern applications [92]:

(i) General empirical knowledge and data bases (contributing to a linear submodel) and expert knowledge, e.g., in the financial or actuarial sectors, contributing to a nonlinear model; in the field of understanding the role of expert knowledge, still too little is understood yet.

(ii) Remaining in the area of financial markets and representing various processes by stochastic differential equations and Lévy processes, the deterministic drift term could be stated by a linear submodel whereas the (possibly simulated) stochastic diffusion term and the compound Poisson processes on jump behaviour could be represented by a nonlinear model.

(iii) While a linear submodel may easily represent given (open) information, a nonlinear submodel could collect hidden information such as, e.g., Hidden Markov Models. This model distinction between non-hidden and hidden can be used in speech processing, image processing, in the financial sector of, e.g., loan banking and credit risk,

and in physics.

The grouping of input dimensions or features mentioned above is in reality done by the help of *data mining*, especially, by *clustering* and *classification* [100]. In fact, firstly, Taylan, Weber and Beck (2007) [89] clustered time points of the change of prices at some stock exchange. Secondly, Weber et al. (2007) [92] regressed credit default to the features of the credit takers. Thirdly, in the modeling and estimation work of Kropat, Weber and Pedamallu (2009) [53] on regulatory networks, a distinction is made between target variables (e.g., from biology, medicine or emissions) and environmental variables (e.g., of toxic substances or from finance). In both categories, items (variables, dimensions of features, or actors) are clustered according to whether they are considered to be related with each other - stochastically dependent or correlated. This is practically done by means of clustering via the geometrical positions of all the given data points, and ellipsoids are raised on the clusters to represent these mutual relationships. Let us underline that this idea also led to the introduction of ellipsoid games by Alparslan Gök and Weber (2009) [1, 98, 99].

2.4 Tikhonov Regularization

Ill-posed problems are frequently encountered in many fields of science. The term itself has its origins in the early 20th century and was introduced by Hadamard who wrongly believed that ill-posed problems did not model real world problems, but later it appeared that it was possible. According to Hadamard, a linear problem is called as *well posed* if it satisfies the following three conditions: (i) existence, (ii) uniqueness, and (iii) stability. However, if at least one or more of these conditions are not satisfied, then the problem is said to be *ill-posed* [50]. Inverse problems, where the values of some model parameters are obtained from the observed data, are often ill-posed.

There are some methods to turn these ill-conditioned problems into well-posed. These methods are established on the so-called *regularization techniques*. The principal goal of regularization is to incorporate more information about the desired solution in order to stabilize the problem and find a useful and stable solution. One of the most commonly used methods is *Tikhonov regularization* named by Andrey Tychonoff in 1984 [29]. In statistics, it is also known as *ridge regression*. The most basic version of

this method is as follows:

$$\min_{\boldsymbol{\beta}} \quad \|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 + \varphi^2 \|\boldsymbol{\beta}\|_2^2, \tag{2.36}$$

where $\varphi^2 = \lambda \in \mathbb{R}_+$ is the regularization or tradeoff parameter.

In Tikhonov regularization, the regularized solution is thought as a minimizer of a weighted combination of the residual norm and a side constraint. As the weight given to the minimization of the side constraint is controlled by the regularization parameter, the quality of the solution is determined by that parameter. A parameter that can fairly balance between the residual error and the regularization error, i.e., in stability of the approximate solution, is considered as an optimal regularization parameter [50]. When the norm of the error in data or the norm of the solution of the error-free problem is available, it is possible to consider and compute a suitable value for the regularization parameter [25].

By the application of Tikhonov regularization to ill-posed equations, the regularization parameter brings the optimal rate of convergence for the approximations. However, when the rates of convergence are derived, assumptions about the nature of the stabilization (i.e., the choice of the semi-norm in the Tikhonov regularization) and the regularity imposed on the solution should be made [67]. Actually, there is a trade-off between stabilization and regularity in terms of convergence rate.

2.4.1 Choosing the Regularization Parameters in Tikhonov Regularization

A method incorporating information about the solution size as well as using information about the residual size is a desired method for choosing the regularization parameter for discrete ill-posed problems. In fact, it is desired to reach a fair balance to keep both of these values small.

Although there are several possible methods to find a suitable choice of the regularization parameter, it is possible to divide these methods into two main categories. The first method is based on a posteriori strategy for choosing the regularization parameter, i.e., knowledge or a good estimate of error norm is needed, while the second one includes the methods that do not require any knowledge about error norm. In fact, it

is based on a priori knowledge of a structure of the input error, meaning that the error terms on the right-hand side can be considered as white noise, uncorrelated zero-mean random variables with a common variance [50]. While the *discrepancy principle* is an example of the first category, the *Cross-Validation* and *L-curve* are examples of the second [33].

The L-curve criterion is a useful method for determining the regularization parameter especially when data includes noise. It is first introduced by Lawson and Hanson in 1974 [55]. This method is established on plotting the norm of the regularized solution versus the corresponding residual norm, and to select a regularization parameter related to the characteristic L-shaped 'corner' of the graph. The transition between under- and over-regularization regions is taking place this corner. There are two meanings of the 'corner': according to first meaning, it is the point, where the curve is closest to the origin and to the second, it is the point, where the curvature is maximum [37]. Specifically, the L-curve has two characteristic parts: "flat" part and an almost "vertical" part [50]. In the more horizontal part, as the regularization parameter is too large, the solution is dominated by the regularization errors and thus solutions are oversmoothed. However, in the vertical part, the regularization parameter is too small and the solution is dominated by the right-hand side errors and thus solutions are undersmoothed. In other words, solutions are affected by the regularization parameter, not by any other additional properties of the problem, e.g., a statistical distribution of the errors [50]. Hence, an appropriate choice of this parameter is very crucial for ill-posed problems.

In linear scale, it is difficult to view the features of the L-curve because of the large range of values for the two norms. However, when drawn in double logarithmic scale, it is possible to see the features of the curve. The corner of the L-curve is clearly seen. As well, particular scalings of the right-hand side and the solution simply shift the L-curve horizontally and vertically [50]. Thus, it is better to analyze the L-curve in the double logarithmic scale.

As it shows how the regularized solution varies by the change in the regularization parameter, L-curve is important for Tikhonov regularization in the analysis of discrete ill-posed problems. The corner of the L-curve corresponds to a good balance between

the minimization of the sizes because, at this corner, the solution changes from being dominated by the regularization errors to being dominated by the errors on right-hand side, and also the corresponding regularization parameter is a good parameter [50]. In fact, the value at this corner corresponds to the optimal value of the regularization parameter [33].

2.4.2 Choosing a Good Solution in Tikhonov Regularization

Tikhonov solution can be expressed easily in terms of the *singular value decomposition* (SVD) of the coefficient matrix $\boldsymbol{X}$:

$$\boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{y},$$

where $\boldsymbol{X}$ is an ill-conditioned matrix. There can be numerous least-squares solutions for a general linear least-squares problem. When the data contain noise and noise is not fitted exactly in any point, then, as long as the norm of the residual $\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2$ is minimized enough, there can be many solutions that fit the data well.

In Tikhonov regularization, we consider all solutions with $\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2$ under the discrepancy principle [3], and select the one that minimizes the norm of $\boldsymbol{\beta}$. Since the norm (length) $\|\boldsymbol{\beta}\|_2$ represents the complexity of the possible solution, it is usually preferred to obtain a solution minimizing the norm of $\boldsymbol{\beta}$. Besides, by minimizing, any unnecessary features can be removed from the regularized solution and the model can show a better fit to data.

Different kinds of Tikhonov regularization are represented as minimization problems. Under the discrepancy principle, all solutions with $\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2$ are considered, and we select the one that minimizes the norm of $\boldsymbol{\beta}$,

$$\min_{\boldsymbol{\beta}} \quad \|\boldsymbol{\beta}\|_2 \qquad \text{such that} \quad \|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2 \leq \delta. \tag{2.37}$$

In the first optimization problem (2.37), as δ increases, the set of feasible models expands, and the minimum value of $\|\boldsymbol{\beta}\|_2$ decreases.

Next, we introduce the problem

$$\min_{\boldsymbol{\beta}} \quad \|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2 \qquad \text{such that} \quad \|\boldsymbol{\beta}\|_2 \leq \epsilon. \tag{2.38}$$

In this second optimization problem, as ϵ decreases, the set of feasible solutions becomes smaller, and the minimum value of $\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2$ increases.

There is also a third option to consider: a *dampened LS problem.* This form is obtained by applying a Lagrange multiplier to the problem (2.38). Then, we get

$$\min_{\boldsymbol{\beta}} \quad \|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 + \varphi^2\|\boldsymbol{\beta}\|_2^2, \tag{2.39}$$

where $\lambda = \varphi^2$ is the Lagrange multiplier and φ is the **regularization parameter** between the two parts.

These three problems can reach the same solution for appropriate choices of δ, ϵ and φ [34]. We will deal with the third option, solving the damped least-squares form of the problem (2.39).

As $\|\boldsymbol{\beta}\|_2$ is a a strictly decreasing function of φ and $\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2$ increasing function of φ, the curve of optimal values of these norms often looks like an L-curve on log-log scale.

It is possible to compute an appropriate value for the parameter of Tikhonov regularization when the norm of the solution of the error-free problem is known or when the norm of the error is known. However, in many important applications, the norm of the error is not explicitly known. In this case, the L-curve is a popular approach for choosing a suitable regularization parameter [34]. Actually, *L-curve* is used to control the trade-off so that the regularization parameter could properly balance the two parts. Besides, λ also controls the sensitivity of the regularized solution (coefficients of basis functions) to perturbations in $\boldsymbol{y}$ and $\boldsymbol{\beta}$, and the perturbation bound is proportional to λ^{-1} [35]. Hence, this regularization parameter is an important quantity controlling the properties of the regularized solution, λ should therefore be chosen with care.

If we plot the optimal values of $\|\boldsymbol{\beta}\|_2^2$ versus $\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2$ on a log-log scale, as $\|\boldsymbol{\beta}\|_2^2$ is a strictly decreasing function of φ and $\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2$ is a strictly increasing function of φ, we can see that the curve has a characteristic L shape.

2.4.3 Solution of Zeroth-Order Tikhonov Regularization Problems

In the previous part, different kinds of Tikhonov regularization represented by minimization problems are mentioned and stated that for some appropriate choice of the values δ, ϵ and φ, these problems can have the same solution. These problems may be solved by using **singular value decomposition**, or **SVD** [3].

In the SVD [54], an $(N \times m)$-matrix $\boldsymbol{X}$ is defined as

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T,$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal matrices and $\boldsymbol{S}$ is an $(N \times m)$-matrix where the nonnegative diagonal elements are called *singular values*. The SVD matrices can be computed in MATLAB by the *svd* command.

The problem (2.39) is a damped least-squares problem with a penalization term φ and it can be solved by the method of normal equations. The set of constraint equations for a 0th-order Tikhonov regularization solution of $\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}$:

$$(\boldsymbol{X}^T\boldsymbol{X} + \varphi^2\boldsymbol{I})\boldsymbol{\beta} = \boldsymbol{X}^T\boldsymbol{y}. \tag{2.40}$$

Applying SVD of $\boldsymbol{X}$, the equation (2.40) can be written as

$$(\boldsymbol{V}\boldsymbol{S}^T\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T + \varphi^2\boldsymbol{I})\boldsymbol{\beta} = \boldsymbol{V}\boldsymbol{S}^T\boldsymbol{U}^T\boldsymbol{y}.$$

As $(\boldsymbol{V}\boldsymbol{S}^T\boldsymbol{S}\boldsymbol{V}^T + \varphi^2\boldsymbol{I})\boldsymbol{\beta} = \boldsymbol{V}\boldsymbol{S}^T\boldsymbol{U}^T\boldsymbol{y}$ is nonsingular for $\varphi \neq 0$, this problem has a unique solution:

$$\boldsymbol{X}_\varphi = \sum_{i=1}^{k} \frac{{s_i}^2}{{s_i}^2 + \varphi^2} \frac{(\boldsymbol{U}_{.,i})^T\boldsymbol{y}}{s_i} \boldsymbol{V}_{.,i} \qquad \text{where} \quad k = \min\{N, m\}.$$

Here, the quantities

$$f_i := \frac{{s_i}^2}{{s_i}^2 + \varphi^2}$$

are called *filter factors*. The filter factors control the contribution of the singular values (and their corresponding singular vectors) to the solution. If $s_i << \varphi$, then $f_i \approx 0$ and if $s_i >> \varphi$, then $f_i \approx 1$. For more details of this application we refer to [3].

In many cases, however, instead of using SVD solution, a solution that minimizes the norm of *first-* or *second-order derivative* of $\boldsymbol{\beta}$ is preferred. Here, the matrix $\boldsymbol{L}$ will

be used to differentiate $\boldsymbol{\beta}$. Matrices that are used to discriminate $\boldsymbol{\beta}$ for the aim of regularization are referred to as *roughening matrices* [3].

These first- or second-order derivatives are approximated from the first- or second-order difference quotients of $\boldsymbol{\beta}$, regarded as a function evaluated at the "points" j and $j+1$. All of them are composed of products $\boldsymbol{L\beta}$ of $\boldsymbol{\beta}$ with matrices $\boldsymbol{L}$ representing the discrete differential operators of first and second order, respectively. They are band structure matrices with values -1, 1 and 1, -2, 1 on the band, respectively [3].

If the unit matrix ($\boldsymbol{L} = \boldsymbol{I}$) is used, then the optimization problem in (2.39) can be considered as a special case of (2.41). This type of problem is called as ***0th-order*** Tikhonov regularization problem and it can be solved by the method of SVD. However, in general, the matrix L is different from the identity matrix, and this type of problems is known as *higher-order regularization problems.*

In *first-order Tikhonov regularization*, the damped least-squared problem

$$\min_{\boldsymbol{\beta}} \ \|\boldsymbol{y} - \boldsymbol{X\beta}\|_2^2 + \varphi^2 \|\boldsymbol{L\beta}\|_2^2 \tag{2.41}$$

is solved by using the matrix $\boldsymbol{L}$:

$$\boldsymbol{L} = \begin{bmatrix} -1 & 1 & & & & \\ & -1 & 1 & & \mathbf{0} & \\ & & \ldots & & & \\ & \mathbf{0} & & -1 & 1 & \\ & & & & -1 & 1 \end{bmatrix}. \tag{2.42}$$

In the *second-order Tikhonov regularization*, the matrix $\boldsymbol{L}$ is as follows:

$$\boldsymbol{L} = \begin{bmatrix} -1 & -2 & 1 & & & & & \\ & 1 & -2 & 1 & & & \mathbf{0} & \\ & & & \ldots & & & & \\ & \mathbf{0} & & & 1 & -2 & 1 & \\ & & & & & 1 & -2 & 1 \end{bmatrix}. \tag{2.43}$$

In (2.42), $\|\boldsymbol{L\beta}\|_2$ is a finite-difference approximation proportional to the first derivative of $\boldsymbol{\beta}$, while it is proportional to the second derivative of $\boldsymbol{\beta}$ in (2.43). As $\|\boldsymbol{L\beta}\|_2$ is zero for any constant model, not just for $\boldsymbol{\beta} = \boldsymbol{0}$, it is a semi-norm. In (2.43), the

minimization of the seminorm $\|\boldsymbol{L}\boldsymbol{\beta}\|_2$ penalizes solutions that are rough in a second order derivative sense.

To solve higher-order problems, the *generalized singular value decomposition*, or *Higher-Order Tikhonov Regularization* ($GSVD$) is used [34, 36]. The GSVD enables the solution to the damped least-squares equation (2.39) to be expressed as a sum of filter factors times generalized singular vectors.

2.5 Regularization Toolbox

In this thesis, MATLAB Regularization toolbox is used [35]. It is a Matlab package for the analysis and solution of discrete ill-posed problems.

Ill-posed problems and regularization methods for computing stabilized solutions to the ill-posed problems occur frequently enough in science and engineering to make it worth-while to present a general framework for their numerical treatment. The purpose of this package of MATLAB routines is to provide the user with easy-to-use routines, based on numerically robust and efficient algorithms, for doing experiments with analysis and solution of discrete ill-posed problems by means of regularization methods.

This toolbox contains a number of useful functions such as *gsvd, cgsvd, discrep, dsvd, lsqi, tgsvd*, and *Tikhonov* for under-determined problems. Singular value decomposition (SVD) is a commonly used numerical tool for analysis of discrete ill-posed problems when there is only one matrix. However, when there is a matrix-pair, the generalized singular value decomposition (GSVD) is used. The SVD reveals all the difficulties associated with the ill-conditioning of a matrix while the GSVD of the matrix-pair yields important insight into the regularization problem involving both the coefficient matrix (basis function matrix) and the regularization matrix $\boldsymbol{L}$ [35].

Specifically, the useful commands for performing Tikhonov regularization are *l_curve* for plotting the L-curve, *l_corner* for estimating the corner using a smoothed spline interpolation method, and *Tikhonov* for computing the solution for a particular value of λ, where $\lambda = \varphi^2$ is the regularization parameter that controls the weight given to minimization of the side constraint relative to minimization of the residual norm as

in the equation (2.39). As this parameter controls the sensitivity of the regularized solution (coefficients of basis functions) to perturbations in $\boldsymbol{y}$ and $\boldsymbol{\beta}$, it is an important quantity and should therefore be chosen with care. The L-curve criterion can be used to decide about this parameter. The corner of this curve, the point with maximum curvature, corresponds to the place this parameter should be chosen. Thus, *l_curve* and *l_corner* commands are helpful here.

2.6 Infinite Kernel Learning

2.6.1 Introduction to Support Vector Machines

Classifying data is a common task in *machine learning*. A major focus of machine learning research is to automatically learn to recognize complex patterns and make intelligent decisions based on data; the difficulty lies in the fact that the set of all possible behaviors given all possible inputs is too complex to describe. Kernel-based techniques such as *support vector machines*, Bayes point machines, kernel principal component analysis, and Gaussian processes represent a major development in machine learning algorithms [46].

SVMs are a set of related supervised learning methods used for classification and regression. A SVM constructs a hyperplane or set of hyperplanes in a high or infinite dimensional space, which can be used for classification, regression or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training datapoints of any class (the so-called *functional margin*), since in general the larger the margin the lower the generalization error of the classifier [44].

In the case of support vector machines, a data point is viewed as a p-dimensional vector (a list of p numbers), and we want to know whether we can separate such points with a $(p-1)$-dimensional hyperplane. This is called a *linear classifier*. There are many hyperplanes that might classify the data. One reasonable choice as the best hyperplane is the one that represents the largest separation, or margin, between the two classes. So we choose the hyperplane so that the distance from it to the nearest data point on each side is maximized. If such a hyperplane exists, it is known as the *maximum-margin hyperplane* and the linear classifier it defines is known as a

maximum margin classifier [44].

Multiclass SVM aims to assign labels to instances by using support vector machines, where the labels are drawn from a finite set of several elements. The dominating approach for doing so is to reduce the single multiclass problem into multiple binary classification problems. Each of the problems yields a binary classifier, which is assumed to produce an output function that gives relatively large values for examples from the positive class and relatively small values for examples belonging to the negative class. Two common methods to build such binary classifiers are where each classifier distinguishes between (i) one of the labels to the rest (one-versus-all) or (ii) between every pair of classes (one-versus-one). Classification of new instances for one-versus-all case is done by a winner-takes-all strategy, in which the classifier with the highest output function assigns the class (it is important that the output functions be calibrated to produce comparable scores). For the one-versus-one approach, classification is done by a max-wins voting strategy, in which every classifier assigns the instance to one of the two classes, then the vote for the assigned class is increased by one vote, and finally the class with most votes determines the instance classification.

2.6.2 Kernel Learning

Machine learning algorithms can also be used to classify nonlinear data. Especially, when the data is large-scale and heterogeneous, multiple kernel methods are helpful. Note that a single kernel is used to map the input space to a higher dimensional feature space. The logic of *multiple kernel learning* is to use finitely many pre-chosen kernels together in a convex combination [83]

$$k_\beta(\boldsymbol{x}_i, \boldsymbol{x}_j) := \sum_{\kappa=1}^{K} \beta_\kappa k_\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) \ , \ \text{where } i, j = 1, 2, \ldots, N. \tag{2.44}$$

The sum in (2.44) is refined by an integral in the present study. A multiple kernel reformulation is modeled via semi-definite programming for choosing the optimum weights of corresponding kernels in [4]. However, this has some negative effects regarding computation time due to semi-definite programming. This reformulation is

enhanced in [83] via *semi-infinite linear programming* by using optimization model

$$\begin{aligned} &\max_{\theta,\boldsymbol{\beta}} \quad \theta \quad (\theta \in \mathbb{R}, \boldsymbol{\beta} \in \mathbb{R}^K) \\ &\text{such that} \quad \boldsymbol{\beta} \geqslant \boldsymbol{0},\ \sum_{\kappa=1}^{K} \beta_\kappa = 1, \\ &\sum_{\kappa=1}^{K} \beta_\kappa S_\kappa(\boldsymbol{\alpha}) \geqslant \theta \ \ \forall \boldsymbol{\alpha} \in \mathbb{R}^N \text{ with } \boldsymbol{0} \leqslant \boldsymbol{\alpha} \leqslant C\mathbf{1} \text{ and } \sum_{i=1}^{N} y_i \alpha_i = 0, \end{aligned} \tag{2.45}$$

where $\mathbf{1} = (1, 1, 1, \ldots, 1)^T \in \mathbb{R}^N$.

However, there is a limitation on the finite combinations of kernels such that they are restricted up to a finite choice. This restriction does not permit always to display the similarity or dissimilarity of data points, especially for large-scaled and highly nonlinearly distributed ones. A finite combination may not work here. Thus, a new combination of *infinitely* many kernels in Riemann-Stieltjes integral form is suggested in [76, 78] by using infinite and semi-infinite programming considering all elements in kernel space which is named *infinitely kernel learning* (IKL) [76, 77, 78]. Then, the problem becomes infinite in both its number of constraints and its dimension; which is known as *infinite programming* (IP). An infinite combination has the following form:

$$k_\beta(x_i, x_j) := \int_\Omega k(x_i, x_j, \omega) d\beta(\omega), \tag{2.46}$$

with $\omega \in \Omega$ being a kernel parameter and β being a monotonically increasing function of integral 1, or just a probability measure on Ω. Moreover, the function $k(x_i, x_j, \omega)$ is supposed to be a twice continuously differentiable function over ω, e.g., $k(x_i, x_j, \cdot) \in C^2$. As infinitely many kernels is suggested to deal with the restriction of the kernel combination composed by finitely many pre-chosen kernels, then, the questions on *which* combinations of kernels and on the *structure* of the mixture of kernels appear, and it may be solved, i.e., by *homotopies* [76, 77, 78].

This new formulation gives the chance to record ("scanning") all possible choices of kernels from the kernel space and, thus, it is possible to keep the uniformity. Infinitely many kernels result in infinitely numerous coefficients and these coefficients are described by an *increasing monotonic function* through *positive measures* [76, 77]. The

formulation of IKL in [76, 77, 78] is as follows:

$$\begin{aligned} &\max_{\theta,\beta} \ \theta \ \ (\theta \in \mathbb{R},\ \beta : \ \text{ a positive measure on } \Omega) \\ &\text{such that } \ \theta - \textstyle\int_\Omega T(\omega, \boldsymbol{\alpha}) d\beta(\omega) \leqslant 0 \ \ (\boldsymbol{\alpha} \in A), \\ &\qquad\qquad\qquad\qquad \textstyle\int_\Omega d\beta(\omega) = 1, \end{aligned} \tag{2.47}$$

where $T(\omega, \boldsymbol{\alpha}) := S(\omega, \boldsymbol{\alpha}) - \sum_{i=1}^{N} \alpha_i,\ S(\omega, \boldsymbol{\alpha}) := \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j, \omega)$ and $\Omega := [0, 1]$ and $A := \{\boldsymbol{\alpha} \in \mathbb{R}^N \mid \boldsymbol{0} \leqslant \boldsymbol{\alpha} \leqslant C\boldsymbol{1} \text{ and } \sum_{i=1}^{N} \alpha_i y_i = 0\}$ are our index sets.

There exist infinitely numerous inequality constraints due to the inequality constraint, uniform in $\boldsymbol{\alpha} \in A$, and the state variable β comes from an infinite dimensional space. Hence, our problem becomes one of *infinite programming (IP)* [2]. The *dual* of (2.47) can be represented as

$$\begin{aligned} &\min_{\sigma,\rho} \ \sigma \ \ (\sigma \in \mathbb{R},\ \rho : \text{ a positive measure on } A) \\ &\text{such that } \ \sigma - \textstyle\int_A T(\omega, \boldsymbol{\alpha}) d\rho(\boldsymbol{\alpha}) \geqslant 0 \ (\omega \in \Omega), \\ &\qquad\qquad\qquad\qquad \textstyle\int_A d\rho(\boldsymbol{\alpha}) = 1. \end{aligned} \tag{2.48}$$

Due to the conditions $\int_\Omega d\beta(\omega) = 1$ and $\int_A d\rho(\boldsymbol{\alpha}) = 1$, positive measures β (or ρ) are probability measures and these measures are parameterized in the present study by the probability density functions as in [76, 77].

Here, we observe that the primal IKL formulation (2.47) and the dual one (2.48) looks like each other except that minimization is replaced with maximization and the direction of inequalities in the constraints are reversed in (2.48). Besides, the index set A and the variable $\boldsymbol{\alpha}$ becomes Ω and ω, respectively. Both index sets are compact and the objective functions of both the dual and the primal, θ and σ, are continuous. Although there exists similarity, the primal and the dual problem are different on the way how the sets of inequality constraints are described [79].

It is explained in [79] that after a parametrization, the primal (or dual) problem has variables in finite dimension as instead of optimizing over the measure β, in an infinite dimensional space, it is minimized over the pdf parameter vector $\wp^{\boldsymbol{\mathcal{P}}}$. This permits to express the infinite programming problem by *semi-infinite programming (SIP)* as the variables are in a finite dimension and there are infinitely numerous inequality constraints. Hence, the primal problem becomes the following SIP with additional constraint functions $\mathrm{U}_i^{\mathcal{P}}(\wp^{\boldsymbol{\mathcal{P}}})$ and $\mathrm{V}_j^{\mathcal{P}}(\wp^{\boldsymbol{\mathcal{P}}})$, coming from the definition of

the parameter sets related to the specific pdf function of the primal problem [76, 77]:

$$
\begin{aligned}
(Primal\ SIP) \qquad & \min_{\theta, \wp^{\mathcal{P}}} \quad -\theta \\
& \text{such that} \quad \int_{\Omega} T(\omega, \boldsymbol{\alpha}) f^{\mathcal{P}}(\omega; \wp^{\mathcal{P}}) d(\omega) - \theta \geq 0 \quad (\boldsymbol{\alpha} \in A), \\
& \qquad \mathrm{U}_i^{\mathcal{P}}(\wp^{\mathcal{P}}) = 0 \quad (i \in I^{\mathcal{P}}), \\
& \qquad \mathrm{V}_j^{\mathcal{P}}(\wp^{\mathcal{P}}) \geq 0 \quad (j \in J^{\mathcal{P}}).
\end{aligned}
\tag{2.49}
$$

2.6.2.1 Exchange and Conceptual Reduction Methods

To solve SIP problems, *discretization* [41] can be employed. It is based on a selection of finitely many points from the infinite index set of inequality constraints. Here, these infinite index sets, respectively, are A and Ω for the primal and the dual problems.

The discretized primal SIP problem of (2.49) can be expressed as follows:

$$
\begin{aligned}
P(A_k) \qquad & \min_{\theta, \wp^{\mathcal{P}}} \quad -\theta \\
& \text{subject to} \quad g^{\mathcal{P}}((\theta, \wp^{\mathcal{P}}), \boldsymbol{\alpha}) := \int_{\Omega} T(\omega, \boldsymbol{\alpha}) f^{\mathcal{P}}(\omega; \wp^{\mathcal{P}}) d\omega - \theta \geqslant 0 \quad (\boldsymbol{\alpha} \in A_k), \\
& \qquad \mathrm{U}_i^{\mathcal{P}}(\wp^{\mathcal{P}}) = 0 \quad (i \in I^{\mathcal{P}}), \\
& \qquad \mathrm{V}_j^{\mathcal{P}}(\wp^{\mathcal{P}}) \geq 0 \quad (j \in J^{\mathcal{P}}).
\end{aligned}
\tag{2.50}
$$

Here, $P(\cdot)$ represents the primal, k shows the iteration step (not a kernel function), $A_k \subseteq \mathbb{R}^N$ is the discretized set. As well, Ω_k can be expressed by a one-dimensional uniform grid, which means discretfization of a chosen set where all elements $\boldsymbol{x} = (x_1, x_2, \ldots, x_l)^T$ have same spacing over their ith coordinate ($i = 1, 2, \ldots, l$). For instance, all columns have the same spacing and all of the rows have the same spacing, but not necessarily the same as the column spacing, in $\mathbb{R}^2$.

An alternative, also more powerful method, to discretization is *exchange method* (PEM) [40, 41, 86, 96]. It is a method between discretization and the reduction ansatz [41] in the sense of refinement and complexity of the algorithm. The discretized upper level problem $P(A_k)$ (2.50) is approximately solved when a discretization A_k is given, whereas the solution of the lower level problem

$$
\begin{aligned}
& \min_{\boldsymbol{\alpha}} \quad g((\theta, \beta), \boldsymbol{\alpha}) \\
& \text{subject to} \quad \boldsymbol{\alpha} \in A
\end{aligned}
\tag{2.51}
$$

is achieved, firstly. The discretization points of A_k are updated in a next iteration. The iteration stops when the algorithm terminates with regard to some stopping criterion. The adaptive exchange algorithm to the algorithm primal problem is given in [79].

Another alternative method is the *conceptual reduction method* ($PCRM$) that is based on local reduction starting with an arbitrary point $\boldsymbol{x}^*$ (not necessarily feasible) for the SIP problem. It solves the lower level problem at that point, e.g., it solves $Q(\boldsymbol{x}^*)$ to get all the local minima $\boldsymbol{y}^1, \boldsymbol{y}^2, \ldots, \boldsymbol{y}^r$ of $Q(\boldsymbol{x}^*)$ (finiteness of local minnima is assumed):

$$\begin{array}{ll} Q(\bar{\boldsymbol{x}}) & \min\limits_{\boldsymbol{y}} \; g(\bar{\boldsymbol{x}}, \boldsymbol{y}) \\ \text{such that} \;\; u_k(\boldsymbol{y}) = 0 \;\; (k \in K) \text{ and } \;\; v_\ell(\boldsymbol{y}) \geq 0 \;\; (\ell \in L). & \end{array} \tag{2.52}$$

As the infinite index sets are compact, and the differentiability, nondegeneracy and continuity assumptions hold, then, by Theorem of Heine-Borel there are finitely many local minima of the lower level problem $Q(\boldsymbol{x})$ indeed (cf. [101]). The adaptive algorithm is given in [79].

In Chapter 3, we will analyze two data sets, homogenous and heterogenous, respectively by IKL and by CMARS techniques. Then, we compare them over the two data sets and observe which technique is good for which data.

CHAPTER 3

METHODS

3.1 Multivariate Adaptive Regression Splines

Multivariate Adaptive Regression Splines ($MARS$) is developed by Friedman in 1991 [25]. It is an important tool in statistics as well as in classification and regression. As an adaptive regression procedure, it is useful for solving high dimensional problems (many explanatory variables). Besides, it shows a great promise for fitting nonlinear multivariate functions. MARS builds flexible models through piecewise linear regressions, and nonlinearity of the models is approximated by having different regression slopes in the corresponding intervals of each predictor. As the intervals underlying those pieces, except of their boundaries, are closed and non-overlapping, the slope of each regression line can change from one interval to another one if there is a "*knot*" defined in between.

The search for finding the predictor variables in the final model and their respective knots is a fast but intensive procedure. MARS searches variables one by one as well as looking for interactions between variables in any degree [19]. The procedure of MARS is simply a generalization of stepwise linear regression. It uses a stepwise procedure to introduce and delete explanatory variables, but also it considers transformations and interactions between the variables. In the algorithm of MARS, each of the explanatory variables is partitioned into regions that each region has its own regression equation. Besides, as MARS has an advantage to estimate the contributions of the basis functions, both the additive and the interactive effects of the predictors are allowed to determine the response variable [90].

The algorithm of MARS includes a two-stage process to generate a model: forward and backward. In the first stage, an overfitted model is produced including an extra large number of basis functions (BFs). However, an overfitted model is not generalized well to new data, even it has a good fit to the data used to build the model. Thus, the backward step is used to prune the model and achieve a model that has a better generalization ability.

The BFs represent distinct intervals of every predictor divided by knots, and every possible knot location is tested. In fact, a MARS model is a linear summation of certain BFs in each dimension, and interactions among them, if existing. The BFs contributing least to the overall performance are removed from the model as initially the model includes many incorrect terms in the forward step. Thus, this removing in the backward step provides to reduce the "*complexity*" of the model without decreasing the fit to the data. Besides, by allowing arbitrary shapes of BFs and their interactions, MARS is capable of reliably tracking very complex data structures that often hide in high dimensions [19].

3.1.1 Word by Word Definition of MARS

The first word, "*multivariate*", means that it is able to deal with multidimensional data, examine individual features and possible interactions among them. The second word "*adaptive*" means selective since MARS automatically deletes certain number of predictors when their contribution to the final model is trivial. The word "*regression*" indicates the commonly used statistical term, often represented as a general prediction function (linear case):

$$Y = \beta_0 + \sum_{j=1}^{k} \beta_j x_j + \epsilon,$$

where Y is the response variable, β_0 is the constant term, β_j are the coefficients and x_j are the predictor variables.

Finally, the last word "*splines*" means a wide class of piecewise defined functions used in applications requiring data interpolation or smoothing. A spline can be developed by dividing the region into a conventional number of regions and a knot is the boundary between regions. By obtaining a sufficient number of knots, any shape can be well

approximated [104].

3.1.2 The Procedure of MARS

Parametric modeling methods such as linear regression are relatively easy to improve and interpret when compared to nonparametric ones. However, they have a limited flexibility and work well only if the underlying assumptions are satisfied. Thus, to overcome the drawbacks of the usual parametric approaches, nonparametric models are developed locally over specific subregions of the data. MARS is one of the nonparametric modeling approaches. The data are searched for an optimum number of subregions and a simple function is optimally fit to the realizations in each subregion [105]. The nonlinearity of a model is approximated by using separate linear regression slopes in separate intervals of the independent variable space.

The general model can be stated as follows:

$$\begin{aligned} Y &= f(x_1, x_2, \ldots, x_p) + \epsilon \\ &= f(\boldsymbol{x}) + \epsilon, \end{aligned}$$

where f is an unknown function, Y is a continuous or binary response variable, $\boldsymbol{x} = (x_1, x_2, \ldots, x_p)^T$ is a vector of predictor variables and the error term ϵ is white noise ($\epsilon \sim N(0, \sigma^2)$).

It is possible to express MARS in an expanded form of the piecewise linear basis functions, $(x-t)_+$ and $(t-x)_+$ with a knotting value at t. The following two functions are truncated linear ones, where $x \in \mathbb{R}$ [39]:

$$(x-t)_+ := \begin{cases} x-t, & \text{if} \quad x > t, \\ 0, & \text{otherwise,} \end{cases} \qquad (t-x)_+ := \begin{cases} t-x, & \text{if} \quad x < t, \\ 0, & \text{otherwise.} \end{cases} \tag{3.1}$$

In equation (3.1), $(\cdot)_+$ indicates the use of only the positive parts. These two truncated functions are piecewise linear nonsmooth splines and they are called as a *reflected pair*. Here, the aim is to form reflected pairs for each input x_j with knots at each observed value x_{ij} of that input. Then, the collection of the BFs can be written as [11]

$$C := \{(x_j - t)_+, (t - x_j)_+ \mid t \in \{x_{1,j}, x_{2,j}, ..., x_{N,j}\}\,,\ j \in \{1, 2, ..., p\}\}\,.$$

There will be $2Np$ BFs in total when all input values are different from each other. As well, even each BF depends only on a single x_j, it is considered as a function over the whole input space $\mathbb{R}^p$ [39].

BFs that are the tensor products of univariate spline functions are used to generalize spline fitting in higher dimensions. Thus, multivariate spline BFs are as follows:

$$B_m(\boldsymbol{x}) := \prod_{k=1}^{K_m} \left(s_{km} \cdot (x_{v(km)} - t_{km})\right)_+ ,$$

where K_m is the total number of truncated linear functions in the mth BF, $x_{v(km)}$ is the input variable corresponding to the kth truncated linear function in the mth basis function, t_{km} is the corresponding knot value and $s_{km} \in \{\pm 1\}$ [104].

The model-building strategy looks like a forward stepwise linear regression. However, here the functions from the set C and their products are used instead of the original inputs. Thus, we reach the following model:

$$Y = \hat{f}(\boldsymbol{x}) + \epsilon = c_0 + \sum_{m=1}^{M} c_m B_m(\boldsymbol{x}) + \epsilon, \tag{3.2}$$

where c_0 is the intercept term and M is the number of BFs in the current model [19].

The coefficients c_m are estimated by least-squares method given some choices for the B_m as in linear regression. Thus, the most important concept to generate the model is the construction of the functions B_m. The model construction starts with only the constant function $B_0(\boldsymbol{x}) = 1$, and all functions in the set C are candidate functions. The possible function forms of BFs $B_m(\boldsymbol{x})$ are as follows [52]:

- 1,
- x_j,
- $(x_j - t_k)_+$,
- $x_l x_j$,
- $(x_j - t_k)_+ x_l$, and
- $(x_j - t_k)_+ (x_l - t_h)_+$.

Here, the point is that each BF must have different input variables in the MARS algorithm. Therefore, the BFs above which are obtained from two multiplied BFs use different input variables such as x_j, x_l, and t_k, t_h are their corresponding knots. At each stage, we consider as a new basis function pair all products of a function B_m in the model set $\mathbf{M}$ with one of the reflected pairs in C. Then, the model set $\mathbf{M}$ is extended with the terms of the form

$$\hat{C}_{M+1}B_l(\boldsymbol{x})(x_j - t)_+ + \hat{C}_{M+2}B_l(\boldsymbol{x})(t - x_j)_+;$$

that provides the largest decrease in training error [39]. Here, the coefficients $\hat{C}_{M+1}$, $\hat{C}_{M+2}$ and also all the other $M+1$ coefficients in the model are estimated by least-squares method. The process finishes when the model set $\mathbf{M}$ has some preset maximum number of terms. Thus, it is clear that the model set $\mathbf{M}$ has an *iterative* built up procedure.

Some possible basis function candidates are as follows [52]:

- $x_j \qquad (j = 1, 2, \ldots, p)$,
- $(x_j - t_k)_+$, if x_j is already in the model,
- $x_l x_j$, if x_l and x_j are already in the model,
- $(x_j - t_k)_+ x_l$ if $x_l x_j$ and $(x_j - t_k)_+$ are already basis functions,
- $(x_j - t_k)_+(x_l - t_h)_+$, if $(x_j - t_k)_+ x_l$ and $(x_l - t_h)_+ x_j$ are already in the model.

Thus, linear terms are involved in the final model providing a better interpretability of the model.

3.1.3 Lack-of-Fit Criterion

A large model equation (3.2) including some unnecessary variables and typically overfitting the data is obtained at the end of the procedure above. In order to detect and remove these variables, a backward deletion procedure is necessary. In this procedure, the term whose removal leads the smallest increase in RSS is deleted at each stage. At the end of this process, an estimated best model $\hat{f}_M$ of each size (number

of terms) M is obtained. Here, cross-validation can be used to estimate the optimal value of M. However, for computational savings, the MARS procedure uses *generalized cross-validation*. This criterion, also known as *lack-of-fit* criterion, is defined as [25]

$$LOF\hat{f}_M = GCV_{Friedman} := \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{f}_M(\boldsymbol{x}_i))^2/(1 - C(M)/N)^2,$$

$$C(M) = \text{trace}(\boldsymbol{B}(\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T) + 1.$$

Here, N is the number of data samples, $C(M)$ is the cost penalty measures of a model containing M basis functions, and B is an $(M \times N)$-matrix. Indeed, $C(M)$ is the number of fitted parameters. The numerator is the usual RSS and it is penalized by the denominator. This denominator helps to balance the increasing variance in the case where the model complexity increases.

Besides, when there are r linearly independent BFs in the model and K knots were selected in the forward stage, then, the cost penalty measure is $C(M) = r + cK$. Here, the quantity c represents a cost for each BF optimization and it is generally equal to 3 [39]. However, if the model is additive, then a penalty of $c = 2$ is used. Moreover, a smaller $C(M)$ produces a larger model with more BFs while a larger $C(M)$ creates a smaller model with less BFs. By the help of lack-of-fit criteria, the best model is obtained along the backward sequence minimizing generalized cross-validation [19, 39].

The use of piecewise linear BFs and the particular model strategy it has, make MARS a special procedure. The piecewise linear BFs are important because they can operate locally; they are zero over a part of their range. If they are multiplied each other, the result is nonzero only over the small part of the factor space where both component functions are nonzero. Hence, the regression surface is built up by using nonzero components locally - just where they are needed. Besides, other basis functions such as polynomials can be used. However, this would produce a nonzero product everywhere, and would not work well.

The limitation put on the formation of model terms that each input can appear at most once in a product helps to avoid the formation of higher-order powers of an input, which increases or decreases too sharply near the boundaries of the factor space. It is

possible to approximate such higher-order powers in a more stable way by the help of piecewise linear functions.

Moreover, the possibility to set an upper limit on the order of interaction is a useful option in the MARS procedure. For instance, if we choose two as a limit, then a three-fold or any higher way of products are not allowed. Instead, this limit just allows pairwise products of piecewise linear functions which can be helpful to interpret the final model. One as an upper limit brings about an additive model [39].

3.1.4 MARS Software Package

The MARS software used in this study is *MARS Version 2, Salford Systems, San Diego, Calif., USA* [104]. MARS helps to find the "best" model by allowing the user to set control parameters to explore different models. Thus, the maximum number of knots is determined by trial and error. Besides, there is no restriction on the maximum number of interactions, it can be more than the degree of two (2-way interaction). Moreover, MARS is a well designed software that implements MARS technique with user-friendly graphical interface. Developed by Salford Systems, the MARS package is available at [13].

Thus far, MARS is introduced and explained with details. In the following part, however, we will mention about $CMARS$, which is a modified form of MARS and an integrated model-based approach. In this method, continuous optimization will be used, in the form of a penalized optimization problem and then, optimization techniques will be applied to solve the problem. This newly introduced method is known as $CMARS$ and will be explained in the following section.

3.2 Conic Multivariate Adaptive Regression Splines

In this section, we introduce a modified version of MARS known as *Conic Multivariate Adaptive Regression Splines* ($CMARS$). Here, "C" represents the word *conic* as well as *convex* and *continuous*.

Being a useful and flexible nonparametric regression technique, MARS has two algo-

rithms to estimate the model function: the forward and the backward stepwise algorithms. In CMARS, however, we can construct a penalized residual sum of squares instead of the backward stepwise algorithm, and treat this function as an optimization problem.

The notation for the piecewise linear BFs in CMARS is as follows:

$$c^{+}(x,\tau) = (+(x-\tau))_{+}, \qquad c^{-}(x,\tau) = (-(x-\tau))_{+}, \tag{3.3}$$

where $[q]_{+} := \max\{0, q\}$ and τ is an univariate knot. As well, the notation to represent the relationship between input and response variables has the following form:

$$Y = f(\boldsymbol{X}) + \epsilon, \tag{3.4}$$

where Y is a response variable, $\boldsymbol{X} = (X_1, X_2, \ldots, X_p)^T$ is a vector of predictor variables and the additive random variable ϵ is white noise.

Reflected pairs for each input X_j $(j = 1, 2, \ldots, k)$ with k-dimensional knots $\boldsymbol{\tau}_i = (\tau_{i,1}, \tau_{i,2}, \ldots, \tau_{i,k})^T$ at or just nearby each input data vectors $\tilde{\boldsymbol{x}}_i = (\tilde{x}_{i,1}, \tilde{x}_{i,2}, \ldots, \tilde{x}_{i,p})^T$ of that input $(i = 1, 2, \ldots, N)$ are constructed. As in the previous section, such a nearby placement indicates a slight modification so that knots' values are not equal to the input values. In fact, to prevent from nondifferentiability in our optimization problem, it may be assumed that without loss of generality $\tau_{i,j} \neq \tilde{x}_{i,j}$ for all i and j. Even, the knots $\tau_{i,j}$ far away from the input values $\tilde{x}_{i,j}$ but providing a better data fit can be selected.

The formulation for the set of BFs is as follows:

$$\wp := \{(x_j - \tau)_{+}, (\tau - x_j)_{+} | \tau \in \{x_{1,j}, x_{2,j}, \ldots, x_{N,j}\}, j \in \{1, 2, \ldots, k\}\}. \tag{3.5}$$

When all the input values are different from each other, the number of total BFs will be $2Np$. Hence, it is possible to write $f(\boldsymbol{X})$ as a linear combination of successively built up basis functions and the intercept θ_0. Then, (2.16) has the following form:

$$Y = \boldsymbol{\theta}_0 + \sum_{m=1}^{M} \boldsymbol{\theta}_m \boldsymbol{\psi}_m(\boldsymbol{X}) + \epsilon. \tag{3.6}$$

where θ_m is the unknown coefficient for the mth basis function $(m = 1, 2, \ldots, M)$, θ_0 is the constant term, ψ_m $(m = 1, \ldots, M)$ represents a basis function from $\wp$ or product

of two or more such functions, ψ_m is taken from a set of M linearly independent basis elements. A set of eligible knots $\tau_{i,j}$ is assigned separately for each input variable dimension and is chosen to approximately coincide with the input levels represented in the data. Interaction basis functions are created by multiplying an existing basis function with a truncated linear function involving a new variable.

The form of the mth basis function provided the observations represented by the data $\boldsymbol{x_i}$ $(i = 1, \ldots, N)$ is as follows [90]:

$$\psi_m(\boldsymbol{x}) := \prod_{j=1}^{K_m} \left(s_{\kappa_j^m} \cdot (x_{\kappa_j^m} - \tau_{\kappa_j^m}) \right)_+, \tag{3.7}$$

where K_m is the number of truncated linear functions multiplied in the mth basis function, $x_{\kappa_j^m}$ is the input variable corresponding to the jth truncated linear function in the mth basis function, $\tau_{\kappa_j^m}$ is the knot value corresponding to the variable $x_{\kappa_j^m}$, and $s_{\kappa_j^m}$ is the selected sign +1 or -1.

As in the previous section, a *lack-of-fit* criterion can be used to compare the possible basis functions. As well, we can restrict the search for new basis functions to a maximum order of interactions. For instance, if it is allowed up to two-factor interactions, then, $K_m \leq 2$ is a proper limitation.

To decrease the complexity of the model together with not reducing the fit to the data, the backward stepwise algorithm is used in MARS. Basis functions that contributes to the smallest increase in the residual squared error are removed from the model at each stage, producing an optimally estimated model $\hat{f}_\alpha$ with respect to each number of terms, called α which expresses some *complexity* of our estimation. To estimate the optimal value of α, generalized cross-validation can be used. This criterion is defined as follows [14]:

$$GCV := \frac{\sum_{i=1}^{N} (y_i - \hat{f}_\alpha(\boldsymbol{x}_i))^2}{N(\boldsymbol{1} - \boldsymbol{M}(\alpha)/N)^2}, \tag{3.8}$$

where $\boldsymbol{M}(\alpha) := u + dK$. Here, N is the number of sample observations, u is the number of linearly independent basis functions, K is the number of knots selected in the forward process, and d is a cost for basis-function optimization and also a smoothing parameter for the procedure.

In this thesis, to estimate the function $f(\boldsymbol{X})$, we propose to employ penalty terms, instead of the backward stepwise algorithm, in addition to the least-squares estimation

in order to control the lack-of-fit from the viewpoint of the *complexity* of the estimation.

3.2.1 The Penalized Residual Sum of Squares

For the MARS model with M_{max} BFs having been collected in the forward stepwise algorithm, let us use penalized residual sum of squares (PRSS) instead of backward elimination. The PRSS form for the MARS model is as follows:

$$PRSS = \sum_{i=1}^{N} (y_i - f(\tilde{\boldsymbol{x}}_i))^2 + \sum_{m=1}^{M_{max}} \lambda_m \sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s \in V(m)}} \int \theta_m^2 \left[D_{r,s}^{\boldsymbol{\alpha}} \psi_m(\boldsymbol{t}^m) \right]^2 d\boldsymbol{t}^m, \tag{3.9}$$

where $V(m) = \{K_j^M | j = 1, 2, \ldots, K_m\}$ is the variable set associated with the mth basis function, ψ_m, $\boldsymbol{t}^m = (t_{m_1}, t_{m_2}, \ldots, t_{m_{K_m}})^T$ represents the vector of variables which contribute to the mth basis function ψ_m. The penalty parameter λ_m is nonnegative ($\lambda_m \geq 0$) for any value of m. This parameter establishes the ***tradeoff*** between both *accuracy*, i.e., a small sum of error squares, and *not too high a complexity*.

The flatness of the model functions is measured with the integrals of the first-order derivatives, and unstability and complexity inscribed into the model (via the model functions) are measured with while the integrals of the second-order derivatives [39, 88]. Besides, we refer to

$$D_{r,s}^{\boldsymbol{\alpha}} \psi_m(\boldsymbol{t}^m) := \frac{\partial^{\,\boldsymbol{\alpha}} \psi_m}{\partial^{\alpha_1} \boldsymbol{t}_r^m \, \partial^{\alpha_2} \boldsymbol{t}_s^m}(\boldsymbol{t}^m) \tag{3.10}$$

for $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)^T$, $|\boldsymbol{\alpha}| = \alpha_1 + \alpha_2$, where $\alpha_1, \alpha_2 \in \{0, 1\}$.

Note that, in any case where $\alpha_i = 2$, the derivative $D_{r,s}^{\alpha} \psi_m(\boldsymbol{t}^m)$ vanishes, and by addressing indices $r < s$, we have applied Schwarz's Theorem. Finally, since all the regarded derivatives of any function ψ_m exist except on a set of measure zero, the integrals and entire optimization problems are well-defined [91].

By using the representations (3.6) and (3.7) in (3.9), the objective function (3.9) has

the following form [91]:

$$PRSS = \sum_{i=1}^{N} \left(\tilde{y}_i - \theta_0 - \sum_{m=1}^{M} \theta_m \psi_m(\tilde{\boldsymbol{x}}_i^m) - \sum_{m=M+1}^{M_{max}} \theta_m \psi_m(\tilde{\boldsymbol{x}}_i^m) \right)^2 + \sum_{m=1}^{M_{max}} \lambda_m \sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s \in V(m)}} \int \theta_m^2 \left[D_{r,s}^{\boldsymbol{\alpha}} \psi_m(\boldsymbol{t}^m) \right]^2 d\boldsymbol{t}^m, \tag{3.11}$$

where the vector $\tilde{\boldsymbol{x}}_i = (\tilde{x}_{i,1}, \tilde{x}_{i,2}, \ldots, \tilde{x}_{i,q})^T$ denotes any of the input vectors while $\tilde{\boldsymbol{x}}_i^m = \left(\tilde{x}_{i,\kappa_1}, \tilde{x}_{i,\kappa_2}, \ldots, \tilde{x}_{i,\kappa_{K_m}} \right)^T$ shows the corresponding projection vectors of $\tilde{\boldsymbol{x}}_i$ onto those coordinates that contribute to the mth basis function, ψ_m, which are related with the ith link function y_i. Here, we recall that those coordinates are collected in the set $V(m)$.

As the second-order derivatives of the piecewise linear functions ψ_m $(m = 1, 2, ..., M)$ and, thus, the penalty terms related are vanishing, we can rearrange the representation of *PRSS* as follows:

$$PRSS := \sum_{i=1}^{N} (y_i - \psi(\tilde{\boldsymbol{d}}_i)^T \boldsymbol{\theta})^2 + \sum_{m=1}^{M_{max}} \lambda_m \sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s \in V(m)}} \int \theta_m^2 \left[D_{r,s}^{\boldsymbol{\alpha}} \psi_m(\boldsymbol{t}^m) \right]^2 d\boldsymbol{t}^m, \tag{3.12}$$

where $\psi(\tilde{\boldsymbol{d}}_i) = (1, \psi_1(\tilde{\boldsymbol{x}}_i^1), \psi_2(\tilde{\boldsymbol{x}}_i^2), \ldots, \psi_M(\tilde{\boldsymbol{x}}_i^M), \psi_{M+1}(\tilde{\boldsymbol{x}}_i^{M+1}), \ldots, \psi_{M_{max}}(\tilde{\boldsymbol{x}}_i^{M_{max}}))^T$, $\boldsymbol{\theta} := (\theta_0, \theta_1, \ldots, \theta_{M_{max}})^T$ with the point $\tilde{\boldsymbol{d}}_i := (\tilde{\boldsymbol{x}}_i^1, \tilde{\boldsymbol{x}}_i^2, \ldots, \tilde{\boldsymbol{x}}_i^M, \tilde{\boldsymbol{x}}_i^{M+1}, \ldots, \tilde{\boldsymbol{x}}_i^{M_{max}})^T$ in the argument. The multi-dimensional integrals

$$\int \theta_m^2 \left[D_{r,s}^{\boldsymbol{\alpha}} \psi_m(\boldsymbol{t}^m) \right]^2 d\boldsymbol{t}^m,$$

are approximated by using discretized forms of them instead [91]. Then, the form of the integrals is as follows:

$$\int_{Q^m} \theta_m^2 \left[D_{r,s}^{\boldsymbol{\alpha}} \psi_m(\boldsymbol{t}^m) \right]^2 d\boldsymbol{t}^m \approx \sum_{(\sigma^j)_{j \in (1,2,\ldots,K_m)} \in \{0,1,2,\ldots,N+1\}^{K_m}} \theta_m^2 \cdot$$

$$\left[D_{r,s}^{\alpha} \psi_m(t_{\substack{l^{\kappa_j^m}_{\sigma^{\kappa_j^m}}, \kappa_j^m}}, ..., t_{\substack{l^{\kappa_{K_m}^m}_{\sigma^{\kappa_{K_m}^m}}, \kappa_{K_m}^m}}) \right]^2 \cdot \prod_{j=1}^{K_m} \left(t_{\substack{l^{\kappa_j^m}_{\sigma^{\kappa_j^{m+1}}}, \kappa_j^m}} - t_{\substack{l^{\kappa_j^m}_{\sigma^{\kappa_j^m}}, \kappa_j^m}} \right).$$

We can rearrange PRSS in the following form:

$$\begin{aligned} PRSS &\approx \sum_{i=1}^{N}\Big(y_i-\boldsymbol{\theta}^T\boldsymbol{\psi}(\tilde{\boldsymbol{d}}_i)\Big)^2 \\ &+ \sum_{m=1}^{M_{max}}\lambda_m \sum_{\substack{|\boldsymbol{\alpha}|=1\\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2}\ \sum_{r<s\atop r,s\in V(m)}\ \sum_{(\sigma^{\kappa_j})}\theta_m^2\cdot\left[D_{r,s}^{\boldsymbol{\alpha}}\psi_m(\tilde{x}_{\substack{l^{\kappa_m^m}_{\sigma^{\kappa_m^m}_j},\kappa_j^m}},\ldots,\tilde{x}_{\substack{l^{\kappa_{K_m}^m}_{\sigma^{\kappa_{K_m}^m}_{K_m}},\kappa_{K_m}^m}})\right]^2 \\ &\cdot \prod_{j=1}^{K_m}\left(\tilde{x}_{\substack{l^{\kappa_j^m}_{\sigma^{\kappa_{j+1}},\kappa_j^m}}}-\tilde{x}_{\substack{l^{\kappa_j^m}_{\sigma^{\kappa_j},\kappa_j^m}}}\right), \end{aligned} \tag{3.13}$$

where $(\sigma^{\kappa_j})_{j\in\{1,2,\ldots,p\}}\in\{0,1,2,...,N+1\}^{K_m}$. There are some more notation related with the sequence (σ^{κ_j}) [91]:

$$\hat{\boldsymbol{x}}_i^m=\left(\tilde{x}_{\substack{l^{\kappa_j^m}_{\sigma^{\kappa_j^m}_j},\kappa_j^m}},\ldots,\tilde{x}_{\substack{l^{\kappa_{K_m}^m}_{\sigma^{\kappa_{K_m}^m}_{K_m}},\kappa_{K_m}^m}}\right),\quad \Delta\hat{\boldsymbol{x}}_i^m:=\prod_{j=1}^{K_m}\left(\tilde{x}_{\substack{l^{\kappa_j^m}_{\sigma^{\kappa_{j+1}},\kappa_j^m}}}-\tilde{x}_{\substack{l^{\kappa_j^m}_{\sigma^{\kappa_j},\kappa_j^m}}}\right). \tag{3.14}$$

$PRSS$ can be approximated by using (3.14) as follows:

$$\begin{aligned} PRSS &\approx \sum_{i=1}^{N}\Big(y_i-\boldsymbol{\theta}^T\boldsymbol{\psi}(\tilde{\boldsymbol{d}}_i)\Big)^2 \\ &+ \sum_{m=1}^{M_{max}}\lambda_m\theta_m^2\sum_{i=1}^{(N+1)^{K_m}}\left(\sum_{\substack{|\boldsymbol{\alpha}|=1\\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2}\ \sum_{r<s\atop r,s\in V(m)}\left[D_{r,s}^{\boldsymbol{\alpha}}\psi_m(\hat{\boldsymbol{x}}_i^m)\right]^2\right)\Delta\hat{\boldsymbol{x}}_i^m. \end{aligned} \tag{3.15}$$

The approximate relation (3.13) can be written in a short form as follows:

$$PRSS\approx\Big\|\boldsymbol{y}-\boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta}\Big\|_2^2+\sum_{m=1}^{M_{max}}\lambda_m\sum_{i=1}^{(N+1)^{K_m}}L_{im}^2\theta_m^2, \tag{3.16}$$

where $\boldsymbol{\psi}(\tilde{\boldsymbol{d}})=\Big(\boldsymbol{\psi}(\tilde{\boldsymbol{d}}_1),\ldots,\boldsymbol{\psi}(\tilde{\boldsymbol{d}}_N)\Big)^T$ is an $(N\times(M_{max}+1))$-matrix and the numbers L_{im}^2 are defined by their roots

$$L_{im}:=\left[\left(\sum_{\substack{|\boldsymbol{\alpha}|=1\\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2}\ \sum_{r<s\atop r,s\in V(m)}\left[D_{r,s}^{\boldsymbol{\alpha}}\psi_m(\hat{\boldsymbol{x}}_i^m)\right]^2\right)\Delta\hat{\boldsymbol{x}}_i^m\right]^{1/2}.$$

3.2.2 Application of Tikhonov Regularization

Here, $PRSS$ is considered as a Tikhonov regularization problem [3] by using the equation (3.16) and it can be written as follows [91]:

$$\begin{aligned} PRSS &\approx \left\| \boldsymbol{y} - \boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta} \right\|_2^2 + \sum_{m=1}^{M_{max}} \lambda_m \sum_{i=1}^{(N+1)^{K_m}} L_{im}^2 \theta_m^2 \\ &= \left\| \boldsymbol{y} - \boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta} \right\|_2^2 + \sum_{m=1}^{M_{max}} \lambda_m \left(\begin{bmatrix} L_{1m}\theta_m, \ldots, L_{(N+1)^{K_m}m}\theta_m \end{bmatrix} \begin{bmatrix} L_{1m}\theta_m \\ \vdots \\ L_{(N+1)^{K_m}m}\theta_m \end{bmatrix} \right) \\ &= \left\| \boldsymbol{y} - \boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta} \right\|_2^2 + \sum_{m=1}^{M_{max}} \lambda_m \|\boldsymbol{L}_m \theta_m\|_2^2 \\ &= \left\| \boldsymbol{y} - \boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta} \right\|_2^2 + \lambda_1 \|\boldsymbol{L}_1 \theta_1\|_2^2 + \ldots + \lambda_{M_{max}} \|\boldsymbol{L}_{M_{max}} \theta_{M_{max}}\|_2^2, \end{aligned}$$

where $\boldsymbol{L}_m := (L_{1m}, \ldots, L_{(N+1)^{K_m},m})^T$ $(m = 1, 2, \ldots, M_{max})$. By making a uniform penalization by taking the same λ for each derivative term, e.g., $\lambda_1 = \lambda_2 = \ldots = \lambda_{M_{max}} =: \lambda$, where $\lambda_m \geq 0$ $(m = 1, 2, \ldots, M_{max})$, $PRSS$ turns into a *Tikhonov regularization problem* with a single tradeoff parameter. Then, the approximation becomes

$$PRSS \approx \left\| \boldsymbol{y} - \boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta} \right\|_2^2 + \lambda \|\boldsymbol{L}\boldsymbol{\theta}\|_2^2 \tag{3.17}$$

with $\boldsymbol{\theta}$ being an $((M_{max}+1)\times 1)$-parameter vector to be estimated through the data points and $\boldsymbol{L}$ is a diagonal $(M_{max}+1)\times(M_{max}+1)$-matrix as follows:

$$\boldsymbol{L} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & L_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & L_{M_{max}} \end{bmatrix}.$$

Hence, the $PRSS$ becomes a *Tikhonov regularization problem* (2.41), where $\lambda = \varphi^2$ for some $\varphi \in \mathbb{R}$ [3]. Our Tikhonov regularization problem has multiple objective functions through a linear combination of $\left\| \boldsymbol{y} - \boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta} \right\|_2^2$ and $\|\boldsymbol{L}\boldsymbol{\theta}\|_2^2$. The solution that minimizes both first objective function $\left\| \boldsymbol{y} - \boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta} \right\|_2^2$ and second objective $(\|\boldsymbol{L}\boldsymbol{\theta}\|_2^2)$ is a desired solution in the sense of a compromise (tradeoff) solution. We refer to [64] for a new contribution to the dependence of locally linear embedding on regularization parameter(s).

In this section, we investigated CMARS model which is based on the regularization of the nonparametric part in a GPLM. In the following section, however, we also mention the regularization of linear part which is investigated in [87].

3.3 The Generalized Partial Linear Model with CMARS

Until now, we focused on the regularization of the nonlinear part of for a generalized partial linear model (GPLM) by Tikhonov regularization. We did not deal with the linear model part for the sake of simplicity, knowing, however, how we have to argue and proceed in the presence of the linear part. In this section, however, we will make an introduction to the regularization of the linear part by CMARS approach.

Previously, in Subsection 2.3.1, the GPLM model is given by the following formula [92]:

$$E(Y|\boldsymbol{X}, \boldsymbol{T}) = G(\boldsymbol{X}^T\boldsymbol{\beta} + \gamma(\boldsymbol{T})),$$

where $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_m)^T$ is a finite dimensional parameter, $\gamma(\cdot)$ is a smooth function, $\boldsymbol{X}$ is an m-variable random vector typically covering discrete covariables, and $\boldsymbol{T}$ is a q-variate random vector of continuous covariables to be modeled in a nonparametric way.

There are many different methods to estimate the unknown parameters in a GPLM. In this study, however, we focus on special types of estimation $\gamma(\cdot)$ by CMARS and $\boldsymbol{\beta}$ by least-square estimation with Tikhonov regularization [87].

3.3.1 Least-Squares Estimation with Tikhonov Regularization

Assuming that $G = H^{-1}$ is a known *link function* connecting the mean of the dependent variable, $\mu = E(Y|\mathbf{X}, \boldsymbol{T})$, to the predictors, the equation (2.15) can be written as follows:

$$H(\mu) = \eta(\boldsymbol{X}, \boldsymbol{T}) = \boldsymbol{X}^T\boldsymbol{\beta} + \gamma(\boldsymbol{T}) = \sum_{j=1}^{m} X_j\beta_j + \gamma(\boldsymbol{T}). \tag{3.18}$$

This can be considered as a semiparametric generalized linear model as all terms are linear except one. In this equation, $\mu_i = G(\eta_i)$ and η_i=$H(\mu_i)$=$\boldsymbol{x}_i^T\boldsymbol{\beta}$+$\gamma(\boldsymbol{t}_i)$, where $\gamma(\cdot)$ is a smooth function.

The unknown parameter $\boldsymbol{\beta}$ of the parametric part is pre-estimated by linear least-squares estimators with Tikhonov regularization. By the help of this method, we minimize the residual sum of squares (RSS):

$$\boldsymbol{y}^{preproc} = \boldsymbol{X}^T\boldsymbol{\beta}^{preproc} + \epsilon = \beta_0 + \sum_{j=1}^{m} X_j\beta_j + \epsilon, \tag{3.19}$$

where $\boldsymbol{\beta}^{preproc} = (\beta_0, \beta_1, \beta_2, \ldots, \beta_m)^T$ and $\boldsymbol{y}^{preproc}$ are our given response data vector $\boldsymbol{y}$. Tikhonov regularization proposed an approximate solution to (3.19) by minimizing the quadratic functional:

$$\min_{\boldsymbol{\beta}^{preproc}} \|\boldsymbol{y}^{preproc} - \boldsymbol{X}\boldsymbol{\beta}^{preproc}\|_2^2 + \lambda\|\boldsymbol{L}\boldsymbol{\beta}^{preproc}\|_2^2, \tag{3.20}$$

where λ is a tradeoff parameter between accuracy and complexity. By solving Tikhonov regularization problem (3.20), the response vector $\boldsymbol{y}^{preproc}$ and the unknown coefficients $\boldsymbol{\beta}^{preproc}$ are found. Then, as the regression coefficients are obtained, the linear least-squares model is subtracted (without intercept) from corresponding responses

$$\boldsymbol{y} - \bar{\boldsymbol{X}}\boldsymbol{\beta}^{preproc} = \hat{\boldsymbol{y}} = \boldsymbol{\eta}. \tag{3.21}$$

Here, $\bar{\boldsymbol{X}}$ is the design matrix $\boldsymbol{X}$, except its first column, and $\hat{\boldsymbol{y}}$ is the resulting vector of residuals, redefined as our new observations. We use $\hat{y}$ to determine the knots (via MARS) for our CMARS application.

3.3.2 CMARS Technique for the Nonparametric Part

Now, to estimate the parameter $\gamma(\cdot)$ of the nonparametric part, we use Conic Multivariate Adaptive Regression Splines (CMARS) approach.

The equation (3.18) can be rewritten by using basis functions as in the equation (3.6). That's, $\gamma(\boldsymbol{t}_i)$ can be written as a linear combination of successively built up by basis functions and the intercept θ_0 as follows:

$$\eta_i = H(\mu) = \boldsymbol{x}_i^T\boldsymbol{\beta} + \theta_0 + \sum_{m=1}^{M} \theta_m\psi_m(\boldsymbol{t_i}), \tag{3.22}$$

where $\boldsymbol{\psi_m}$ $(m = 1, \ldots, M)$ represents a basis function from $\wp$ or a product of two or more such functions, $\boldsymbol{\psi_m}$ is taken from a set of M linearly independent basis elements, and θ_m are the unknown coefficients for the mth basis function $(m = 1, \ldots, M)$, θ_0

is the constant term. For each input variable dimension, a set of eligible knots $\tau_{i,j}$, selected by MARS with reference to the residual vector, is assigned separately for each dimension, and this set is chosen to approximately coincide with the input levels represented in the data.

Here, the form of the mth basis function is same with the equation (3.6) in CMARS except the variable $\boldsymbol{t}$ instead of $\boldsymbol{x}$. The form of BF is as follows:

$$\psi_m(\boldsymbol{t}) := \prod_{j=1}^{K_m} (s_{\kappa_j^m} \cdot (t_{\kappa_j^m} - \tau_{\kappa_j^m}))_+, \tag{3.23}$$

where $\boldsymbol{t_i}$ $(i = 1, \ldots, N)$ are the observations provided and $t_{\kappa_j^m}$ is the input variable corresponding to the jth truncated linear function in the mth basis function, $\tau_{\kappa_j^m}$ is the knot value corresponding to the variable $t_{\kappa_j^m}$, and $s_{\kappa_j^m}$ is the selected sign +1 or -1.

As in CMARS, a *lack-of-fit* criterion can be used to compare the possible basis functions and the search for new basis functions can be restricted to a maximum order of interactions. For example, if only up to two-factor interactions are permitted, then $K_m \leq 2$ would be our restriction. As explained in Section 3.2, this algorithm does this by removing from the model basis functions that contribute to the smallest increase in the residual squared error at each stage, producing an optimally estimated model $\hat{\gamma}_\alpha$ with respect to each number of terms, called α, which expresses some *complexity* of our estimation. Again, to estimate the lack-of-fit, GCV criterion can be used to estimate the optimal value of α. In this GPLM with CMARS model, GCV is similar to the equation (3.8) and can be defined as follows:

$$GCV := \frac{\sum_{i=1}^{N} (\eta_i - \boldsymbol{x}_i^T \hat{\boldsymbol{\beta}} - \hat{\gamma}_\alpha(\boldsymbol{t}_i))^2}{(1 - \boldsymbol{M}(\alpha)/N)^2}, \tag{3.24}$$

where $\boldsymbol{M}(\alpha) := u + dK$ [87]. Here, N is the number of sample observations, u is the number of linearly independent basis functions, K is the number of knots selected in the forward process, and d is a cost for basis-function optimization as well as a smoothing parameter for the procedure.

To estimate the function $\gamma(\boldsymbol{t})$, we propose to employ the penalty terms in addition to the least-squares estimation instead of the backward stepwise algorithm. Thus, it is possible to control the lack-of-fit from the viewpoint of the *complexity* of the estimation.

3.3.3 The Penalized Residual Sum of Squares Problem for GPLM with CMARS

The equation (2.16) can be written as follows:

$$\eta = H(\mu_i) = \boldsymbol{x}_i^T\boldsymbol{\beta} + \boldsymbol{\psi}^T(\boldsymbol{t}_i)\boldsymbol{\theta}, \tag{3.25}$$

where $\boldsymbol{\theta}$=$(\theta_0, \theta_1, \ldots, \theta_M)^T$ and $\boldsymbol{\psi(\tilde{d}_i)}$=$(\psi_1(t_i), \psi_2(t_i), \ldots, \psi_M(t_i))$. The form of the penalized residual sum of squares (PRSS) for the GPLM with CMARS is as follows:

$$\begin{aligned} PRSS &= \sum_{i=1}^{N} \left(\eta_i - \boldsymbol{x}_i^T\boldsymbol{\beta} - \boldsymbol{\psi}^T(\boldsymbol{t}_i)\boldsymbol{\theta}\right)^2 \\ &+ \sum_{m=1}^{M_{max}} \lambda_m \sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V(m)}} \int \theta_m^2 \left[D_{r,s}^{\boldsymbol{\alpha}}\psi_m(\boldsymbol{t}^m)\right]^2 d\boldsymbol{t}^m, \end{aligned} \tag{3.26}$$

where $V(m) = \{K_j^M | j = 1, 2, \ldots, K_m\}$ is the variable set associated with the mth basis function ψ_m, $\boldsymbol{t}^m = (t_{m_1}, t_{m_2}, \ldots, t_{m_{K_m}})^T$ represents the vector of variables which contribute to the mth basis function ψ_m. This equation is similar to the the PRSS of CMARS, however, here, the parameter of the parametric part, $\boldsymbol{x}_i^T\boldsymbol{\beta}$, is included in the PRSS. Moreover, we refer to

$$D_{r,s}^{\boldsymbol{\alpha}}\psi_m(\boldsymbol{t}^m) := \frac{\partial^{\,\boldsymbol{\alpha}}\psi_m}{\partial^{\alpha_1}\boldsymbol{t}_r^m\,\partial^{\alpha_2}\boldsymbol{t}_s^m}(\boldsymbol{t}^m) \tag{3.27}$$

for $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)^T$, $|\boldsymbol{\alpha}| = \alpha_1 + \alpha_2$, where $\alpha_1, \alpha_2 \in \{0, 1\}$.

The objective function PRSS has the following form when the representations (3.22) and (3.23) are used in (3.26):

$$\begin{aligned} PRSS = \sum_{i=1}^{N} &\left(\eta_i - \boldsymbol{x}_i^T\boldsymbol{\beta} - \theta_0 - \sum_{m=1}^{M}\theta_m\psi_m(\boldsymbol{t}_i^m) - \sum_{m=M+1}^{M_{max}}\theta_m\psi_m(\boldsymbol{t}_i^m)\right)^2 \\ &+ \sum_{m=1}^{M_{max}} \lambda_m \sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V(m)}} \int \theta_m^2 \left[D_{r,s}^{\boldsymbol{\alpha}}\psi_m(\boldsymbol{t}^m)\right]^2 d\boldsymbol{t}^m, \end{aligned} \tag{3.28}$$

where the vector $\boldsymbol{t}_i = (t_{i,1}, t_{i,2}, \ldots, t_{i,q})^T$ denotes any of the input vectors while $\boldsymbol{t}_i^m = \left(t_{i,\kappa_1}, t_{i,\kappa_2}, \ldots, t_{i,\kappa_{K_m}}\right)^T$ shows the corresponding projection vectors of $\boldsymbol{t}_i$ onto those coordinates that contribute to the mth basis function ψ_m, which are related with the ith link function η_i [91].

As the second-order derivatives of the piecewise linear functions ψ_m $(m = 1, 2, ..., M)$ and, hence, the related penalty terms are vanishing, we can rearrange the representation of *PRSS* as follows:

$$\begin{aligned} PRSS &:= \sum_{i=1}^{N}(y_i - \boldsymbol{x}_i^T\boldsymbol{\beta} - \psi(\tilde{d}_i)^T\theta)^2 \\ &+ \sum_{m=1}^{M_{max}} \lambda_m \sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V(m)}} \int \theta_m^2 \left[D_{r,s}^{\boldsymbol{\alpha}}\psi_m(\boldsymbol{t}^m)\right]^2 d\boldsymbol{t}^m, \end{aligned} \tag{3.29}$$

where $\psi(\tilde{\boldsymbol{d}}_i) = (1, \psi_1(\boldsymbol{t}_i^1), \psi_2(\boldsymbol{t}_i^2), \ldots, \psi_M(\boldsymbol{t}_i^M), \psi_{M+1}(\boldsymbol{t}_i^{M+1}), \ldots, \psi_{M_{max}}(\boldsymbol{t}_i^{M_{max}}))^T$, $\boldsymbol{\theta}:= (\theta_0, \theta_1, \ldots, \theta_{M_{max}})^T$ with the point $\tilde{\boldsymbol{d}}_i := (\boldsymbol{t}_i^1, \boldsymbol{t}_i^2, \ldots, \boldsymbol{t}_i^M, \boldsymbol{t}_i^{M+1}, \ldots, \boldsymbol{t}_i^{M_{max}})^T$ in the argument. To approximate the multi-dimensional integrals

$$\int \theta_m^2 \left[D_{r,s}^{\boldsymbol{\alpha}}\psi_m(\boldsymbol{t}^m)\right]^2 d\boldsymbol{t}^m,$$

the discretizations and model approximations are used. Then, the discretized form of the integrals can be written as:

$$\int_{Q^m} \theta_m^2 \left[D_{r,s}^{\boldsymbol{\alpha}}\psi_m(\boldsymbol{t}^m)\right]^2 d\boldsymbol{t}^m \approx \sum_{(\sigma^j)_{j\in\{1,2,\ldots,K_m\}}\in\{0,1,\ldots,N+1\}^{K_m}} \theta_m^2 \cdot$$

$$\left[D_{r,s}^{\alpha}\psi_m(\tilde{x}_{\substack{l^{\kappa_j^m},\kappa_j^m \\ \sigma^{\kappa_j^m}}}, \ldots, \tilde{x}_{\substack{l^{\kappa_{K_m}^m},\kappa_{K_m}^m \\ \sigma^{\kappa_{K_m}^m}}})\right]^2 \cdot \prod_{j=1}^{K_m}\left(\tilde{x}_{\substack{l^{\kappa_j^m},\kappa_j^m \\ \sigma^{\kappa_j^m+1}}} - \tilde{x}_{\substack{l^{\kappa_j^m},\kappa_j^m \\ \sigma^{\kappa_j^m}}}\right).$$

The PRSS can be represented in the following form:

$$\begin{aligned} PRSS &\approx \sum_{i=1}^{N}\left(\eta_i - \boldsymbol{x}_i^T\boldsymbol{\beta} - \boldsymbol{\psi}(\tilde{\boldsymbol{d}}_i)^T\boldsymbol{\theta}\right)^2 \\ &+ \sum_{m=1}^{M_{max}} \lambda_m \sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V(m)}} \sum_{(\sigma^{\kappa_j})} \theta_m^2 \cdot \left[D_{r,s}^{\boldsymbol{\alpha}}\psi_m(t_{\substack{l^{\kappa_j^m},\kappa_j^m \\ \sigma^{\kappa_j^m}}}, \ldots, t_{\substack{l^{\kappa_{K_m}^m},\kappa_{K_m}^m \\ \sigma^{\kappa_{K_m}^m}}})\right]^2 \\ &\cdot \prod_{j=1}^{K_m}\left(t_{\substack{l^{\kappa_j^m},\kappa_j^m \\ \sigma^{\kappa_{j+1}}}} - t_{\substack{l^{\kappa_j^m},\kappa_j^m \\ \sigma^{\kappa_j}}}\right), \end{aligned} \tag{3.30}$$

where $(\sigma^{\kappa_j})_{j\in\{1,2,\ldots,p\}} \in \{0, 1, 2, ..., N+1\}^{K_m}$. There are some more notation related with the sequence (σ^{κ_j}) [91]:

$$\hat{\boldsymbol{t}}_i^m = \left(t_{\substack{l^{\kappa_j^m},\kappa_j^m \\ \sigma^{\kappa_j^m}}}, \ldots, t_{\substack{l^{\kappa_{K_m}^m},\kappa_{K_m}^m \\ \sigma^{\kappa_{K_m}^m}}}\right), \quad \Delta\hat{\boldsymbol{t}}_i^m := \prod_{j=1}^{K_m}\left(t_{\substack{l^{\kappa_j^m},\kappa_j^m \\ \sigma^{\kappa_{j+1}}}} - t_{\substack{l^{\kappa_j^m},\kappa_j^m \\ \sigma^{\kappa_j}}}\right). \tag{3.31}$$

It is possible to approximate $PRSS$ by using (3.31) as follows:

$$\begin{aligned} PRSS &\approx \sum_{i=1}^{N}\Big(\eta_i - \boldsymbol{x}_i^T\boldsymbol{\beta} - \boldsymbol{\psi}(\tilde{\boldsymbol{d}}_i)^T\boldsymbol{\theta}\Big)^2 \\ &+ \sum_{m=1}^{M_{max}}\lambda_m\theta_m^2\sum_{i=1}^{(N+1)^{K_m}}\left(\sum_{\substack{|\boldsymbol{\alpha}|=1\\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2}\sum_{\substack{r<s\\ r,s\in V(m)}}\Big[D_{r,s}^{\boldsymbol{\alpha}}\psi_m(\hat{\boldsymbol{t}}_i^m)\Big]^2\right)\Delta\hat{\boldsymbol{t}}_i^m. \end{aligned} \tag{3.32}$$

The approximate relation in (3.30) can be written with a shorter representation as follows:

$$PRSS \approx \Big\|\boldsymbol{\eta} - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta}\Big\|_2^2 + \sum_{m=1}^{M_{max}}\lambda_m\sum_{i=1}^{(N+1)^{K_m}}L_{im}^2\theta_m^2, \tag{3.33}$$

where $\boldsymbol{\psi}(\tilde{\boldsymbol{d}}) = \Big(\boldsymbol{\psi}(\tilde{\boldsymbol{d}}_1),\ldots,\boldsymbol{\psi}(\tilde{\boldsymbol{d}}_N)\Big)^T$ is an $(N\times(M_{max}+1))$-matrix and the numbers L_{im}^2 are defined by their roots

$$L_{im} := \left[\left(\sum_{\substack{|\boldsymbol{\alpha}|=1\\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2}\sum_{\substack{r<s\\ r,s\in V(m)}}\Big[D_{r,s}^{\boldsymbol{\alpha}}\psi_m(\hat{\boldsymbol{t}}_i^m)\Big]^2\right)\Delta\hat{\boldsymbol{t}}_i^m\right]^{1/2}.$$

3.3.4 Application of Tikhonov Regularization in GPLM with CMARS

The equation (3.33) can also be written as [91]:

$$PRSS \approx \|\boldsymbol{\eta} - \boldsymbol{X}^*\boldsymbol{\beta}^*\|_2^2 + \sum_{m=1}^{M_{max}}\lambda_m\sum_{i=1}^{(N+1)^{K_m}}L_{im}^2\theta_m^2, \tag{3.34}$$

where $\boldsymbol{X}^* = (\boldsymbol{X}\ \ \boldsymbol{\psi}(\tilde{\boldsymbol{d}}))$ is a block matrix constructed by $(N\times p)$-matrix $\boldsymbol{X}$ and $(N\times(M_{max}+1))$ matrix $\boldsymbol{\psi}(\tilde{\boldsymbol{d}})$, $\boldsymbol{\beta}^* = (\boldsymbol{\beta}^T,\ \boldsymbol{\theta}^T)^T$ is a vector constructed $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ vectors.

Then, we deal with the linear systems equations of $\boldsymbol{\eta} = \boldsymbol{X}^*\boldsymbol{\beta}^*$, approximately. As this problem may be ill-posed (irregular or unstable), we approach $PRSS$ function as a *Tikhonov regularization problem* [68]. A Tikhonov solution can be expressed quite easily in terms of *singular value decomposition* (SVD) of the coefficient matrix $\boldsymbol{X}^*$ of a regarded linear system of equations $\boldsymbol{\eta} = \boldsymbol{X}^*\boldsymbol{\beta}^*$.

Hence, we consider formula (3.34) and arrange it as follows:

$$\begin{aligned} PRSS &\approx \|\boldsymbol{\eta} - \boldsymbol{X}^*\boldsymbol{\beta}^*\|_2^2 + \sum_{m=1}^{M_{max}} \lambda_m \sum_{i=1}^{(N+1)^{K_m}} L_{im}^2 \theta_m^2 \\ &= \sum_{m=1}^{M_{max}} \lambda_m \left[(\boldsymbol{L}_{1m}\theta_m)^2 + (\boldsymbol{L}_{2m}\theta_m)^2 + \ldots + (\boldsymbol{L}_{(N+1)^{K_m}m}\theta_m)^2 \right] \\ &= \|\boldsymbol{\eta} - \boldsymbol{X}^*\boldsymbol{\beta}^*\|_2^2 + \sum_{m=1}^{M_{max}} \lambda_m \|\boldsymbol{L}_m\theta_m\|_2^2, \end{aligned} \tag{3.35}$$

where $\boldsymbol{L}_m := (L_{1m}, L_{2m}, \ldots, L_{(N+1)^{K_m},m})^T$ $(m = 1, 2, \ldots, M_{max})$. By making a uniform penalization by taking the same λ for each derivative term, i.e., $\lambda_1 = \lambda_2 = \ldots = \lambda_{M_{max}} =: \lambda$, where $\lambda_m \geq 0$ $(m = 1, 2, \ldots, M_{max})$, the equation 3.35 can be turned into a *Tikhonov Regularization Problem* with a single tradeoff parameter. Then, it looks as follows:

$$PRSS \approx \|\boldsymbol{\eta} - \boldsymbol{X}^*\boldsymbol{\beta}^*\|_2^2 + \lambda\|\boldsymbol{L}\boldsymbol{\theta}\|_2^2, \tag{3.36}$$

where $\boldsymbol{\theta}$ is an $((M_{max} + 1) \times 1)$-parameter vector and $\boldsymbol{L}$ is a diagonal $(M_{max} + 1)\times(M_{max} + 1)$-matrix with first column $\boldsymbol{L}_0 = \boldsymbol{0}_{N+1^{K_m}}$ and the other columns being the vectors $\boldsymbol{L}_m$ defined above. Let us consider the high-dimensional matrix $\boldsymbol{L}^* = (\boldsymbol{R}^*, \boldsymbol{L})$, where $\boldsymbol{R}^*$ is an $((M_{max} + 1) \times p)$-matrix with entries being first or second derivative of $\boldsymbol{\beta}$. These derivatives are obtained from first- or second-order difference quotients of $\boldsymbol{\beta}$, considered as a function which is calculated at the points i and $i+1$. These difference quotients approximate first- and second-order derivatives; altogether, they are composed of products $\boldsymbol{R}^*\boldsymbol{\beta}$ of $\boldsymbol{\beta}$ with matrices $\boldsymbol{R}^*$ that show, respectively, the discrete differential operators of first- and second order. Hence, our *PRSS* problem becomes a classical *Tikhonov regularization problem* [68] with $\varphi > 0$, e.g., $\lambda = \varphi^2$ for some $\varphi \in \mathbb{R}$. Our Tikhonov regularization problem has multiple objective functions through a linear combination of $\|\boldsymbol{\eta} - \boldsymbol{X}^*\boldsymbol{\beta}^*\|_2^2$ and $\|\boldsymbol{X}^*\boldsymbol{\beta}^*\|_2^2$. We choose the solution which minimizes the objective function $\|\boldsymbol{\eta} - \boldsymbol{X}^*\boldsymbol{\beta}^*\|_2^2$ and also the regularization objective $\|\boldsymbol{L}^*\boldsymbol{\beta}^*\|_2^2$ in the sense of a compromise (tradeoff) solution. We refer to [64] for a new contribution to the dependence of locally linear embedding on regularization parameter(s).

Thus far, we explained the theory of the regularizing both the parametric and nonparametric parts of a GPLM by CMARS approach. Herewith, for the sake of simplicity, we disregard the linear model part, knowing, however, how we have to argue and

proceed in the presence of the linear part. And thus, we restricted ourselves on the regularization of nonparametric part in the numerical example of this thesis which is explained with details in the following chapter.

3.4 Numeric Example

In this part, we provide two numeric examples to the regularization of nonparametric part by Tikhonov regularization. For this aim, we use two different types of continuous data; one has interaction between variables and the other does not have. As MARS [25] is able to deal with complex data structures in high dimensions reliably by allowing arbitrary shapes of BFs and their interactions [19], we are expecting better results for data including interaction.

Basis functions are found by MARS version 2 developed by Salford Systems and regression coefficients are estimated by using Tikhonov regularization in MATLAB.

3.4.1 Data with No Interaction

For this study, we used a data set with three predictor variables (x_1, x_2, x_3) and 25 observations (taken from Mendenhal and Sincich (1994) [57], p. 678). As the MARS model is constructed by trial and error, we set the maximum number of BFs to four, i.e., $M_{max} = 4$. The basis functions constructed by MARS are as follows:

$$\psi_1(\boldsymbol{x}) = \max\{0, x_1 - 14.11\},$$

$$\psi_2(\boldsymbol{x}) = \max\{0, 14.11 - x_1\},$$

$$\psi_3(\boldsymbol{x}) = \max\{0, x_1 - 12.01\},$$

$$\psi_4(\boldsymbol{x}) = \max\{0, 12.01 - x_1\},$$

ψ_1 and ψ_2 are standard and reflected BFs for the predictor variable x_1, respectively. The knot point for ψ_1 and for ψ_2 is 14.11. As well, ψ_3 and ψ_4 are the standard and mirror image BFs for the predictor variable x_1. The knot point for ψ_3 and for ψ_4 is 12.01.

The knot values are selected to be very close to the input values of the data point, not exactly same, so that it is possible to differentiate the optimization problem. The selected knot values for corresponding BFs are stated below.

For ψ_1: $\tau_{1,1} = 14.11$, $\widetilde{\tau}_{1,1} = 14.10$, which is not equal to $\tau_{1,1} = 14.11$ but very close to it.

For ψ_2: $\widetilde{\tau}_{1,1} = 14.10$, where $\tau_{1,1} = 14.11$.

For ψ_3: $\widetilde{\tau}_{25,1} = 12.00$, where $\tau_{25,1} = 12.01$.

For ψ_4: $\widetilde{\tau}_{25,1} = 12.00$, where $\tau_{25,1} = 12.01$.

Then, the BFs of the form can be written as follows:

$$\begin{aligned}
\psi_1 : K_1 &= 1, \\
x_{\kappa_1^1} &= x_1, \\
\tau_{\kappa_1^1} &= 14.11, \\
s_{\kappa_1^1} &= +1, \\
\psi_1(\boldsymbol{t}^1) &= \prod_{j=1}^{K_1} \left(s_{\kappa_1^1} \cdot (x_{\kappa_1^1} - \tau_{\kappa_1^1}) \right)_+ \\
&= \left(s_{\kappa_1^1} \cdot (x_{\kappa_1^1} - \tau_{\kappa_1^1}) \right)_+ ,
\end{aligned}$$

and

$$\begin{aligned}
\psi_2 : K_2 &= 1, \\
x_{\kappa_1^2} &= x_1, \\
\tau_{\kappa_1^2} &= 14.11, \\
s_{\kappa_1^2} &= -1, \\
\psi_2(\boldsymbol{t}^2) &= \prod_{j=1}^{K_2} \left(s_{\kappa_1^2} \cdot (x_{\kappa_1^2} - \tau_{\kappa_1^2}) \right)_+ \\
&= \left(s_{\kappa_1^2} \cdot (x_{\kappa_1^2} - \tau_{\kappa_1^2}) \right)_+ ,
\end{aligned}$$

and

$$\begin{aligned}
\psi_3 : K_3 &= 1, \\
x_{\kappa_1^3} &= x_1, \\
\tau_{\kappa_1^3} &= 12.01, \\
s_{\kappa_1^3} &= +1, \\
\psi_3(\boldsymbol{t}^3) &= \prod_{j=1}^{K_3} \left(s_{\kappa_1^3} \cdot (x_{\kappa_1^3} - \tau_{\kappa_1^3}) \right)_+ \\
&= \left(s_{\kappa_1^3} \cdot (x_{\kappa_1^3} - \tau_{\kappa_1^3}) \right)_+ ,
\end{aligned}$$

and

$$\begin{aligned}
\psi_4 : K_4 &= 1, \\
x_{\kappa_1^4} &= x_1, \\
\tau_{\kappa_1^4} &= 12.01, \\
s_{\kappa_1^4} &= -1, \\
\psi_4(\boldsymbol{t}^4) &= \prod_{j=1}^{K_4} \left(s_{\kappa_1^4} \cdot (x_{\kappa_1^4} - \tau_{\kappa_1^4}) \right)_+ \\
&= \left(s_{\kappa_1^4} \cdot (x_{\kappa_1^4} - \tau_{\kappa_1^4}) \right)_+ ,
\end{aligned}$$

Hence, the large model can be written as:

$$\begin{aligned}
y &= \theta_0 + \sum_{m=1}^{M} \boldsymbol{\theta}_m \boldsymbol{\psi}_m(\boldsymbol{x}) + \epsilon, \\
&= \theta_0 + \theta_1 \max\{0, x_1 - 14.11\} + \theta_2 \max\{0, 14.11 - x_1\} + \theta_3 \max\{0, x_1 - 12.01\} \\
&\quad + \theta_4 \max\{0, 12.01 - x_1\} + \epsilon.
\end{aligned}$$

The objective function PRSS in (3.9) can be written as:

$$PRSS = \sum_{i=1}^{25} (y_i - f(\tilde{\boldsymbol{x}}_i))^2 + \sum_{m=1}^{4} \lambda_m \sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_m}} \int \theta_m^2 \left[D_{r,s}^{\boldsymbol{\alpha}} \psi_m(\boldsymbol{t}^m)\right]^2 d\boldsymbol{t}^m$$

$$= \sum_{i=1}^{25} \left(y_i - \boldsymbol{\theta}^T \boldsymbol{\psi}(\tilde{\boldsymbol{d}}_i)\right)^2 + \lambda_1 \left(\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_1}} \int \theta_1^2 \left[D_{r,s}^{\boldsymbol{\alpha}} \psi_1(\boldsymbol{t}^1)\right]^2 d\boldsymbol{t}^1 \right)$$

$$+\lambda_2 \left(\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_2}} \int \theta_2^2 \left[D_{r,s}^{\boldsymbol{\alpha}} \psi_2(\boldsymbol{t}^2)\right]^2 d\boldsymbol{t}^2 \right)$$

$$+\lambda_3 \left(\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_3}} \int \theta_3^2 \left[D_{r,s}^{\boldsymbol{\alpha}} \psi_3(\boldsymbol{t}^3)\right]^2 d\boldsymbol{t}^3 \right)$$

$$+\lambda_4 \left(\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_4}} \int \theta_4^2 \left[D_{r,s}^{\boldsymbol{\alpha}} \psi_4(\boldsymbol{t}^4)\right]^2 d\boldsymbol{t}^4. \right)$$

All evaluations for the notations V_m and $\boldsymbol{t}^m$ (for $m = 1, \ldots, 4$) in the above equation are given below:

$$\begin{aligned}
V_1 &= \left\{\kappa_j^1 | j = 1\right\} = \{1\}, \quad \boldsymbol{t}^1 = (t_1^1)^T = (x_1)^T, \\
V_2 &= \left\{\kappa_j^2 | j = 1\right\} = \{1\}, \quad \boldsymbol{t}^2 = (t_1^2)^T = (x_1)^T, \\
V_3 &= \left\{\kappa_j^3 | j = 1\right\} = \{1\}, \quad \boldsymbol{t}^3 = (t_1^3)^T = (x_1)^T, \\
V_4 &= \left\{\kappa_j^4 | j = 1\right\} = \{1\}, \quad \boldsymbol{t}^4 = (t_1^4)^T = (x_1)^T.
\end{aligned}$$

Moreover, the corresponding derivatives for the BFs $D_{r,s}^{\boldsymbol{\alpha}} \psi_m(\boldsymbol{t}^m)$ (for $m = 1, \ldots, 4$) are written below.

As there is no interaction for ψ_1, $r = s = 1$. The sum of indicated first- and second-order derivatives for ψ_1 is:

$$\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_1}} \left[D_{r,s}^{\boldsymbol{\alpha}} \psi_1(\boldsymbol{t}^1)\right]^2 d\boldsymbol{t}^1,$$

where

$$|\,\boldsymbol{\alpha}|=1: \qquad D_1^1\psi_1(\boldsymbol{t}^1) := \frac{\partial \psi_1}{\partial t_1^1}(\boldsymbol{t}^1) = \frac{\partial \psi_1}{\partial x_1}(x_1) = \begin{cases} 1, & \text{if } x_1 > 14.11 \\ 0, & \text{otherwise.} \end{cases},$$

$$|\,\boldsymbol{\alpha}|=2: \qquad D_1^2\psi_1(\boldsymbol{t}^1) := \frac{\partial^2 \psi_1}{\partial t_1^1 \partial t_1^1}(\boldsymbol{t}^1) = \frac{\partial^2 \psi_1}{\partial x_1 \partial x_1}(x_1) = 0 \;\text{ for all }\; x_1.$$

As there is no interaction for ψ_2, $r = s = 1$. The sum of indicated first- and second-order derivatives for ψ_2 is:

$$\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_2}} \left[D_{r,s}^{\boldsymbol{\alpha}}\psi_2(\boldsymbol{t}^2)\right]^2 d\boldsymbol{t}^2,$$

where

$$|\,\boldsymbol{\alpha}|=1: \qquad D_1^1\psi_2(\boldsymbol{t}^2) := \frac{\partial \psi_2}{\partial t_1^2}(\boldsymbol{t}^2) = \frac{\partial \psi_2}{\partial x_1}(x_1) = \begin{cases} -1, & \text{if } x_1 < 14.11 \\ 0, & \text{otherwise.} \end{cases}$$

$$|\,\boldsymbol{\alpha}|=2: \qquad D_1^2\psi_2(\boldsymbol{t}^2) := \frac{\partial^2 \psi_2}{\partial t_1^2 \partial t_1^2}(\boldsymbol{t}^2) = \frac{\partial^2 \psi_2}{\partial x_1 \partial x_1}(x_1) = 0 \;\text{ for all }\; x_1.$$

Again there is no interaction for the BF $\psi_3(\boldsymbol{t}^3)$, so: $r = s = 1$. The sum of indicated first- and second-order derivatives for ψ_3 is:

$$\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_3}} \left[D_{r,s}^{\boldsymbol{\alpha}}\psi_3(\boldsymbol{t}^3)\right]^2 d\boldsymbol{t}^3,$$

where

$$|\,\boldsymbol{\alpha}|=1: \qquad D_1^1\psi_3(\boldsymbol{t}^3) := \frac{\partial \psi_3}{\partial t_1^3}(\boldsymbol{t}^3) = \frac{\partial \psi_3}{\partial x_1}(x_1) = \begin{cases} 1, & \text{if } x_1 > 12.01 \\ 0, & \text{otherwise.} \end{cases},$$

$$|\,\boldsymbol{\alpha}|=2: \qquad D_1^2\psi_3(\boldsymbol{t}^3) := \frac{\partial^2 \psi_3}{\partial t_1^3 \partial t_1^3}(\boldsymbol{t}^3) = \frac{\partial^2 \psi_3}{\partial x_1 \partial x_1}(x_1) = 0 \;\text{ for all }\; x_1.$$

For the BF $\psi_4(\boldsymbol{t}^4)$, there is no interaction; so: $r = s = 1$. The sum of indicated first- and second-order derivatives for ψ_4 is:

$$\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_4}} \left[D_{r,s}^{\boldsymbol{\alpha}}\psi_4(\boldsymbol{t}^4)\right]^2 d\boldsymbol{t}^4,$$

where

$$|\,\boldsymbol{\alpha}|=1: \qquad D_1^1\psi_4(\boldsymbol{t}^4) := \frac{\partial \psi_4}{\partial t_1^4}(\boldsymbol{t}^4) = \frac{\partial \psi_4}{\partial x_1}(x_1) = \begin{cases} -1, & \text{if } x_1 < 12.01 \\ 0, & \text{otherwise.} \end{cases},$$

$$|\,\boldsymbol{\alpha}|=2: \qquad D_1^2\psi_4(\boldsymbol{t}^4) := \frac{\partial^2 \psi_4}{\partial t_1^4 \partial t_1^4}(\boldsymbol{t}^4) = \frac{\partial^2 \psi_4}{\partial x_1 \partial x_1}(x_1) = 0 \;\text{ for all }\; x_1.$$

Hence, the PRSS objective function in (3.9) becomes:

$$\begin{aligned} PRSS &= \sum_{i=1}^{25}\left(y_i - \boldsymbol{\theta}^T\boldsymbol{\psi}(\tilde{\boldsymbol{d}}_i)\right)^2 \\ &+ \sum_{m=1}^{4}\lambda_m \sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_m}} \int \theta_m^2 \left[D_{r,s}^{\boldsymbol{\alpha}}\psi_m(\boldsymbol{t}^m)\right]^2 d\boldsymbol{t}^m. \end{aligned}$$

Assuming that $\lambda_1 = \lambda_2 = ... = \lambda_{M_{max}} =: \lambda$, then PRSS turns into the Tikhonov regularization form:

$$PRSS \approx \left\|\boldsymbol{y} - \boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta}\right\|_2^2 + \lambda \left\|\boldsymbol{L}\boldsymbol{\theta}\right\|_2^2 \ .$$

Here, the first parts of the PRSS objective function and the Tikhonov regularization problem are equal to each other while the second parts are approximately equal.

$$\sum_{i=1}^{25}\left(y_i - \boldsymbol{\theta}^T\boldsymbol{\psi}(\tilde{\boldsymbol{d}}_i)\right)^2 = \left\|\boldsymbol{y} - \boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta}\right\|_2^2,$$

$$\sum_{m=1}^{4}\lambda_m \sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_m}} \int \theta_m^2 \left[D_{r,s}^{\boldsymbol{\alpha}}\psi_m(\boldsymbol{t}^m)\right]^2 d\boldsymbol{t}^m \approx \lambda \left\|\boldsymbol{L}\boldsymbol{\theta}\right\|_2^2.$$

The RSS values are presented below (the complete form of RSS can be seen in Appendix A):

$$\begin{aligned} \sum_{i=1}^{25}\left(y_i - \boldsymbol{\theta}^T\boldsymbol{\psi}(\tilde{\boldsymbol{d}}_i)\right)^2 = (13.6 - \theta_0 - (\max\{0, 14.1 - 14.11\})\theta_1 - \\ (\max\{0, 14.11 - 14.1\})\theta_2 - \\ (\max\{0, 14.1 - 12.01\})\theta_3 - \\ (\max\{0, 12.01 - 14.1\})\theta_4)^2 + \\ (16.6 - \theta_0 - (\max\{0, 16 - 14.11\})\theta_1 - \\ (\max\{0, 14.11 - 16\})\theta_2 - \end{aligned}$$

$$
\begin{aligned}
&(\max\{0, 16 - 12.01\})\theta_3 - \\
&(\max\{0, 12.01 - 16\})\theta_4)^2 + \\
&\vdots \\
(14.9 - \theta_0 - (\max\{0, 12 - 14.11\})\theta_1 - \\
&(\max\{0, 14.11 - 12\})\theta_2 - \\
&(\max\{0, 12 - 12.01\})\theta_3 - \\
&(\max\{0, 12.01 - 12\})\theta_4)^2.
\end{aligned}
$$

By computing the maximum functions, the form of the RSS is as follows:

$$
\begin{aligned}
\sum_{i=1}^{25} \left(y_i - \boldsymbol{\theta}^T \boldsymbol{\psi}(\tilde{\boldsymbol{d}}_i)\right)^2 &= (13.6 - \theta_0 - 0.01\theta_2 - 2.9\theta_3)^2 + \\
&(16.6 - \theta_0 - 1.89\theta_1 - 3.99\theta_3)^2 + \\
&\vdots \\
&(14.9 - \theta_0 - 2.11\theta_2 - 0.01\theta_4)^2
\end{aligned}
$$

$$
\begin{aligned}
&= (13.6 - \theta_0 - 0.01\theta_2 - 2.9\theta_3)^T (13.6 - \theta_0 - 0.01\theta_2 - 2.9\theta_3) + \\
&(16.6 - \theta_0 - 1.89\theta_1 - 3.99\theta_3)^T (16.6 - \theta_0 - 1.89\theta_1 - 3.99\theta_3) + \\
&(23.5 - \theta_0 - 15.77\theta_1 - 17.87\theta_3)^T (23.5 - \theta_0 - 15.77\theta_1 - 17.87\theta_3) + \\
&\vdots \\
&(13.9 - \theta_0 - 1.09\theta_1 - 3.19\theta_3)^T (13.9 - \theta_0 - 1.09\theta_1 - 3.19\theta_3) + \\
&(14.9 - \theta_0 - 2.11\theta_2 - 0.01\theta_4)^T (14.9 - \theta_0 - 2.11\theta_2 - 0.01\theta_4).
\end{aligned}
$$

We can change the form of the above summation into vector notation and get the representation below. Thus, RSS which is the first part of PRSS is as follows:

$$
\begin{aligned}
\sum_{i=1}^{25} \left(y_i - \boldsymbol{\theta}^T \boldsymbol{\psi}(\tilde{\boldsymbol{d}}_i)\right)^2 &= \left(\boldsymbol{y} - \boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta}\right)^T \left(\boldsymbol{y} - \boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta}\right) \\
&= \|\boldsymbol{y} - \boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta}\|_2^2.
\end{aligned}
$$

Thus, it can be seen that the first parts of the PRSS function and Tikhonov regularization form are equal.

Now, we focus on the second parts which are approximately equal. the multi-dimensional integral in the second part of the equation (3.12) takes the form of (3.15) after discretization and this discretized form is denoted by $\boldsymbol{L}$. Then, we reach the formulation in equation (2.41).

The L_m $(m = 1, \ldots, 4)$ values corresponding to BFs, ψ_1, ψ_2, ψ_3, ψ_4, are calculated as follows:

$$\begin{aligned} L_1 &= \sum_{i=1}^{(N+1)^{K_1}} \left[\left(\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s \in V_1}} \left[D_{r,s}^{\boldsymbol{\alpha}} \left(\max\{0, x_1 - 14.11\} \right) \right]^2 \right) \left(\tilde{x}_{l^{\kappa_1^1}_{\sigma^{\kappa_1^1}}+1,\kappa_1^1} - \tilde{x}_{l^{\kappa_1^1}_{\sigma^{\kappa_1^1}},\kappa_1^1} \right) \right] \\ &= 3.9243, \end{aligned}$$

$$\begin{aligned} L_2 &= \sum_{i=1}^{(N+1)^{K_2}} \left[\left(\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s \in V_2}} \left[D_{r,s}^{\boldsymbol{\alpha}} \left(\max\{0, 14.11 - x_1\} \right) \right]^2 \right) \left(\tilde{x}_{l^{\kappa_1^2}_{\sigma^{\kappa_1^2}}+1,\kappa_1^2} - \tilde{x}_{l^{\kappa_1^2}_{\sigma^{\kappa_1^2}},\kappa_1^2} \right) \right] \\ &= 3.6878, \end{aligned}$$

$$\begin{aligned} L_3 &= \sum_{i=1}^{(N+1)^{K_3}} \left[\left(\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s \in V_3}} \left[D_{r,s}^{\boldsymbol{\alpha}} \left(\max\{0, x_1 - 12.01\} \right) \right]^2 \right) \left(\tilde{x}_{l^{\kappa_1^3}_{\sigma^{\kappa_1^3}}+1,\kappa_1^3} - \tilde{x}_{l^{\kappa_1^3}_{\sigma^{\kappa_1^3}},\kappa_1^3} \right) \right] \\ &= 4.1833, \end{aligned}$$

and

$$\begin{aligned} L_4 &= \sum_{i=1}^{(N+1)^{K_4}} \left[\left(\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s \in V_4}} \left[D_{r,s}^{\boldsymbol{\alpha}} \left(\max\{0, 12.01 - x_1\} \right) \right]^2 \right) \left(\tilde{x}_{l^{\kappa_1^4}_{\sigma^{\kappa_1^4}}+1,\kappa_1^4} - \tilde{x}_{l^{\kappa_1^4}_{\sigma^{\kappa_1^4}},\kappa_1^4} \right) \right] \\ &= 3.3912, \end{aligned}$$

$$\boldsymbol{L} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 3.9243 & 0 & 0 & 0 \\ 0 & 0 & 3.6878 & 0 & 0 \\ 0 & 0 & 0 & 4.1833 & 0 \\ 0 & 0 & 0 & 0 & 3.3912 \end{bmatrix}.$$

In the $\boldsymbol{L}$ matrix, the first column elements of $\boldsymbol{L}$ are all zero and the diagonal elements of this matrix $\boldsymbol{L}_m$ $(m = 1, 2, ..., 5)$ are as given above. Then, $\boldsymbol{L\theta}$ is as follows:

$$\|\boldsymbol{L\theta}\|_2^2 = (\theta_1 \cdot (3.9243))^2 + (\theta_2 \cdot (3.6878))^2 + (\theta_3 \cdot (4.1833))^2 + (\theta_4 \cdot (3.3912))^2 .$$

3.4.2 Data with Interaction

For this study, we used a data set with five predictor variables $(x_1, x_2, x_3, x_4, x_5)$ and 32 observations (taken from Myers and Montgomery (2002) [66] p. 71). The MARS model is built by using the Salford MARS v.2 [58], and to construct the model the maximum number of BFs (M_{max}) and the highest degree of interactions are determined by trial and error. In this data set, M_{max} and the highest degree of interactions are five and two, respectively. Hence, the largest model built in the forward MARS algorithm by the software contains the following BFs:

$$\psi_1(\boldsymbol{x}) = \max\{0, x_2 - 2.21\},$$

$$\psi_2(\boldsymbol{x}) = \max\{0, 2.21 - x_2\},$$

$$\psi_3(\boldsymbol{x}) = \max\{0, x_4 - 0.26\},$$

$$\psi_4(\boldsymbol{x}) = \max\{0, x_1 - 1601\} \cdot \max\{0, x_4 - 0.26\},$$

$$\psi_5(\boldsymbol{x}) = \max\{0, x_5 - 0.71\} \cdot \max\{0, x_4 - 0.26\}.$$

Here, ψ_1 and ψ_2 are the standard and reflected BFs for the predictor variable x_2. The knot point for ψ_1 and for ψ_2 is 2.21. Besides, BF ψ_4 uses the function ψ_3 to express the interaction between the predictor variables x_1 and x_4. As well, ψ_5 represents the interaction between the predictor variables x_4 and x_5.

As in the previous data set, we choose the knot values very close to the input values of the data point so that it is possible to differentiate the optimization problem. The selected knot values for corresponding BFs are written below.

For ψ_1: $\tau_{18,2} = 2.21$, $\widetilde{\tau}_{18,2} = 2.2$, which is not equal to $\tau_{18,2} = 2.21$ but very close to it.

For ψ_2: $\widetilde{\tau}_{18,2} = 2.20$ where, $\tau_{18,2} = 2.21$.

For ψ_3: $\widetilde{\tau}_{1,4} = 0.25$ where, $\tau_{1,4} = 0.26$.

For ψ_4: $\widetilde{\tau}_{6,1} = 1600$ where, $\tau_{6,1} = 1601$, and $\widetilde{\tau}_{1,4} = 0.25$ where, $\tau_{1,4} = 0.26$.

For ψ_5: $\widetilde{\tau}_{25,5} = 0.70$ where, $\tau_{25,5} = 0.71$, and $\widetilde{\tau}_{6,4} = 0.25$ where, $\tau_{6,4} = 0.26$.

Then, the BFs of the form can be written as follows:

$$\begin{aligned}
\psi_1 : K_1 &= 1, \\
x_{\kappa_1^1} &= x_2, \\
\tau_{\kappa_1^1} &= 2.21, \\
s_{\kappa_1^1} &= +1, \\
\psi_1(\boldsymbol{t}^1) &= \prod_{j=1}^{K_1} \left(s_{\kappa_1^1} \cdot (x_{\kappa_1^1} - \tau_{\kappa_1^1}) \right)_+ \\
&= \left(s_{\kappa_1^1} \cdot (x_{\kappa_1^1} - \tau_{\kappa_1^1}) \right)_+ ,
\end{aligned}$$

and

$$\begin{aligned}
\psi_2 : K_2 &= 1, \\
x_{\kappa_1^2} &= x_2, \\
\tau_{\kappa_1^2} &= 2.21, \\
s_{\kappa_1^2} &= -1, \\
\psi_2(\boldsymbol{t}^2) &= \prod_{j=1}^{K_2} \left(s_{\kappa_1^2} \cdot (x_{\kappa_1^2} - \tau_{\kappa_1^2}) \right)_+ \\
&= \left(s_{\kappa_1^2} \cdot (x_{\kappa_1^2} - \tau_{\kappa_1^2}) \right)_+ ,
\end{aligned}$$

and

$$\begin{aligned}
\psi_3 : K_3 &= 1, \\
x_{\kappa_1^3} &= x_4, \\
\tau_{\kappa_1^3} &= 0.26, \\
s_{\kappa_1^3} &= +1, \\
\psi_3(\boldsymbol{t}^3) &= \prod_{j=1}^{K_3} \left(s_{\kappa_1^3} \cdot (x_{\kappa_1^3} - \tau_{\kappa_1^3}) \right)_+ \\
&= \left(s_{\kappa_1^3} \cdot (x_{\kappa_1^3} - \tau_{\kappa_1^3}) \right)_+ ,
\end{aligned}$$

and

$$\begin{aligned}
\psi_4 : K_4 &= 2, \\
x_{\kappa_1^4} &= x_1, \quad x_{\kappa_2^4} = x_4, \\
\tau_{\kappa_1^4} &= 0.26, \quad \tau_{\kappa_2^4} = 2.21, \\
s_{\kappa_1^4} &= +1, \quad s_{\kappa_2^4} = +1, \\
\psi_4(\boldsymbol{t}^4) &= \prod_{j=1}^{K_4} \left[s_{\kappa_j^4} \cdot (x_{\kappa_j^4} - \tau_{\kappa_j^4}) \right]_+ \\
&= \left(s_{\kappa_1^4} \cdot (x_{\kappa_1^4} - \tau_{\kappa_1^4}) \right)_+ \cdot \left(s_{\kappa_2^4} . (x_{\kappa_2^4} - \tau_{\kappa_2^4}) \right)_+ ,
\end{aligned}$$

and

$$\begin{aligned}
\psi_5 : K_5 &= 2, \\
x_{\kappa_1^5} &= x_5, \quad x_{\kappa_2^5} = x_5, \\
\tau_{\kappa_1^5} &= 0.71, \quad \tau_{\kappa_2^5} = 2.21, \\
s_{\kappa_1^5} &= +1, \quad s_{\kappa_2^5} = +1, \\
\psi_5(\boldsymbol{t}^5) &= \prod_{j=1}^{K_5} \left(s_{\kappa_j^5} \cdot (x_{\kappa_j^5} - \tau_{\kappa_j^5}) \right)_+ \\
&= \left(s_{\kappa_1^5} \cdot (x_{\kappa_1^5} - \tau_{\kappa_1^5}) \right)_+ \cdot \left[s_{\kappa_2^5} . (x_{\kappa_2^5} - \tau_{\kappa_2^5}) \right]_+ .
\end{aligned}$$

Then, the large model can be written as follows:

$$\begin{aligned}
y &= \theta_0 + \sum_{m=1}^{M} \theta_m \psi_m(\boldsymbol{x}) + \epsilon, \\
&= \theta_0 + \theta_1 \max\{0, x_2 - 2.21\} + \theta_2 \max\{0, 2.21 - x_2\} + \theta_3 \max\{0, x_4 - 0.26\} \\
&\quad + \theta_4 \max\{0, x_1 - 1601\} \cdot \max\{0, x_4 - 0.26\} \\
&\quad + \theta_5 \max\{0, x_5 - 0.71\} \cdot \max\{0, x_4 - 0.26\} + \epsilon.
\end{aligned}$$

The PRSS objective function in (3.9) can be written as follows:

$$
\begin{aligned}
PRSS &= \sum_{i=1}^{32}\left(y_i - \boldsymbol{\theta}^T\boldsymbol{\psi}(\tilde{\boldsymbol{d}}_i)\right)^2 + \sum_{m=1}^{5}\lambda_m \sum_{\substack{|\boldsymbol{\alpha}|=1\\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2}\sum_{\substack{r<s\\ r,s\in V_m}}\int \theta_m^2\left[D_{r,s}^{\boldsymbol{\alpha}}\psi_m(\boldsymbol{t}^m)\right]^2 d\boldsymbol{t}^m \\
&= \sum_{i=1}^{32}\left(y_i - \boldsymbol{\theta}^T\boldsymbol{\psi}(\tilde{\boldsymbol{d}}_i)\right)^2 + \lambda_1\left(\sum_{\substack{|\boldsymbol{\alpha}|=1\\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2}\sum_{\substack{r<s\\ r,s\in V_1}}\int \theta_1^2\left[D_{r,s}^{\boldsymbol{\alpha}}\psi_1(\boldsymbol{t}^1)\right]^2 d\boldsymbol{t}^1\right) \\
&\quad +\lambda_2\left(\sum_{\substack{|\boldsymbol{\alpha}|=1\\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2}\sum_{\substack{r<s\\ r,s\in V_2}}\int \theta_2^2\left[D_{r,s}^{\boldsymbol{\alpha}}\psi_2(\boldsymbol{t}^2)\right]^2 d\boldsymbol{t}^2\right) \\
&\quad +\lambda_3\left(\sum_{\substack{|\boldsymbol{\alpha}|=1\\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2}\sum_{\substack{r<s\\ r,s\in V_3}}\int \theta_3^2\left[D_{r,s}^{\boldsymbol{\alpha}}\psi_3(\boldsymbol{t}^3)\right]^2 d\boldsymbol{t}^3\right) \\
&\quad +\lambda_4\left(\sum_{\substack{|\boldsymbol{\alpha}|=1\\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2}\sum_{\substack{r<s\\ r,s\in V_4}}\int \theta_4^2\left[D_{r,s}^{\boldsymbol{\alpha}}\psi_4(\boldsymbol{t}^4)\right]^2 d\boldsymbol{t}^4\right) \\
&\quad +\lambda_5\left(\sum_{\substack{|\boldsymbol{\alpha}|=1\\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2}\sum_{\substack{r<s\\ r,s\in V_5}}\int \theta_5^2\left[D_{r,s}^{\boldsymbol{\alpha}}\psi_5(\boldsymbol{t}^5)\right]^2 d\boldsymbol{t}^5\right).
\end{aligned}
$$

All evaluations for the notations V_m and $\boldsymbol{t}^m$ (for $m = 1, \ldots, 5$) in the above equation are given below:

$$
\begin{aligned}
V_1 &= \left\{\kappa_j^1 | j = 1\right\} = \{2\}, \quad \boldsymbol{t}^1 = (t_1^1)^T = (x_2)^T, \\
V_2 &= \left\{\kappa_j^2 | j = 1\right\} = \{2\}, \quad \boldsymbol{t}^2 = (t_1^2)^T = (x_2)^T, \\
V_3 &= \left\{\kappa_j^3 | j = 1\right\} = \{4\}, \quad \boldsymbol{t}^3 = (t_1^3)^T = (x_4)^T, \\
V_4 &= \left\{\kappa_j^4 | j = 1, 2\right\} = \{1, 4\}, \quad \boldsymbol{t}^4 = (t_1^4, t_2^4)^T = (x_1, x_4)^T, \\
V_5 &= \left\{\kappa_j^5 | j = 1, 2\right\} = \{4, 5\}, \quad \boldsymbol{t}^5 = (t_1^5, t_2^5)^T = (x_4, x_5)^T.
\end{aligned}
$$

As well, the corresponding derivatives for the BFs $D_{r,s}^{\boldsymbol{\alpha}}\psi_m(\boldsymbol{t}^m)$ (for $m = 1, \ldots, 5$) are stated below.

For the BF $\psi_1(\boldsymbol{t}^1)$, there is no interaction; so: $r = s = 2$. The sum of indicated first- and second-order derivatives for ψ_1 is:

$$\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_1}} \left[D_{r,s}^{\boldsymbol{\alpha}}\psi_1(\boldsymbol{t}^1)\right]^2 d\boldsymbol{t}^1,$$

where

$$|\,\boldsymbol{\alpha}|=1: \qquad D_2^1\psi_1(\boldsymbol{t}^1) := \frac{\partial\psi_1}{\partial t_1^1}(\boldsymbol{t}^1) = \frac{\partial\psi_1}{\partial x_2}(x_2) = \begin{cases} -1, & \text{if } x_2 > 2.21 \\ 0, & \text{otherwise,} \end{cases},$$

$$|\,\boldsymbol{\alpha}|=2: \qquad D_2^2\psi_1(\boldsymbol{t}^1) := \frac{\partial^2\psi_1}{\partial t_1^1\partial t_1^1}(\boldsymbol{t}^1) = \frac{\partial^2\psi_1}{\partial x_2\partial x_2}(x_2) = 0 \;\text{ for all }\; x_2.$$

For the BF $\psi_2(\boldsymbol{t}^2)$, there is no interaction; so: $r = s = 2$. The sum of indicated first- and second-order derivatives for ψ_2 is:

$$\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_2}} \left[D_{r,s}^{\boldsymbol{\alpha}}\psi_2(\boldsymbol{t}^2)\right]^2 d\boldsymbol{t}^2,$$

where

$$|\,\boldsymbol{\alpha}|=1: \qquad D_2^1\psi_2(\boldsymbol{t}^2) := \frac{\partial\psi_2}{\partial t_1^2}(\boldsymbol{t}^2) = \frac{\partial\psi_2}{\partial x_2}(x_2) = \begin{cases} 1, & \text{if } x_2 < 2.21 \\ 0, & \text{otherwise,} \end{cases},$$

$$|\,\boldsymbol{\alpha}|=2: \qquad D_2^2\psi_2(\boldsymbol{t}^2) := \frac{\partial^2\psi_2}{\partial t_1^2\partial t_1^2}(\boldsymbol{t}^2) = \frac{\partial^2\psi_2}{\partial x_2\partial x_2}(x_2) = 0 \;\text{ for all }\; x_2.$$

For the BF $\psi_3(\boldsymbol{t}^3)$, there is no interaction; so: $r = s = 4$. The sum of indicated first- and second-order derivatives for ψ_3 is:

$$\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_3}} \left[D_{r,s}^{\boldsymbol{\alpha}}\psi_3(\boldsymbol{t}^3)\right]^2 d\boldsymbol{t}^3,$$

where

$$|\,\boldsymbol{\alpha}|=1: \qquad D_4^1\psi_3(\boldsymbol{t}^3) := \frac{\partial\psi_3}{\partial t_1^3}(\boldsymbol{t}^3) = \frac{\partial\psi_3}{\partial x_4}(x_4) = \begin{cases} 1, & \text{if } x_4 > 0.26 \\ 0, & \text{otherwise} \end{cases},$$

$$|\,\boldsymbol{\alpha}|=2: \qquad D_4^2\psi_3(\boldsymbol{t}^3) := \frac{\partial^2\psi_3}{\partial t_1^3\partial t_1^3}(\boldsymbol{t}^3) = \frac{\partial^2\psi_3}{\partial x_4\partial x_4}(x_4) = 0 \;\text{ for all }\; x_4.$$

For the BF $\psi_4(\boldsymbol{t}^4)$, interaction exists between the predictors x_1 and x_4. Here, $r = 1$

and $s = 4$ so: $r < s$. Then, the sum of indicated first- and second-order derivatives of ψ_4 can be written as:

$$\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_4}} \left[D_{r,s}^{\boldsymbol{\alpha}}\psi_4(\boldsymbol{t}^4)\right]^2 d\boldsymbol{t}^4,$$

where

$$|\,\boldsymbol{\alpha}| = 1 : D_{1,4}^1\psi_4(\boldsymbol{t}^4) := \frac{\partial\psi_4}{\partial t_1^4}(\boldsymbol{t}^4) = \frac{\partial\psi_4}{\partial x_1}(x_1, x_4) = \begin{cases} \max\{0, x_4 - 0.26\}, & \text{if } x_1 > 1601, \\ 0, & \text{otherwise,} \end{cases}$$

$$D_{1,4}^1\psi_4(\boldsymbol{t}^4) := \frac{\partial\psi_4}{\partial t_2^4}(\boldsymbol{t}^4) = \frac{\partial\psi_4}{\partial x_4}(x_1, x_4) = \begin{cases} \max\{0, x_1 - 1601\}, & \text{if } x_4 > 0.26, \\ 0, & \text{otherwise;} \end{cases}$$

$$|\,\boldsymbol{\alpha}| = 2 : D_{1,4}^2\psi_4(\boldsymbol{t}^4) := \frac{\partial^2\psi_4}{\partial t_1^4 \partial t_2^4}(\boldsymbol{t}^4) = \frac{\partial^2\psi_4}{\partial x_1 \partial x_4}(x_1, x_4) = \begin{cases} 1, & \text{if } x_4 > 0.26, \\ 0, & \text{otherwise.} \end{cases}$$

For the BF $\psi_5(\boldsymbol{t}^5)$, there is also an interaction between the predictors x_4 and x_5, and $r = 4$ and $s = 5$ so: $r < s$. Then, the sum of indicated first- and second-order derivatives of ψ_5 can be written as:

$$\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_5}} \left[D_{r,s}^{\boldsymbol{\alpha}}\psi_5(\boldsymbol{t}^5)\right]^2 d\boldsymbol{t}^5,$$

where

$$|\,\boldsymbol{\alpha}| = 1 : D_{4,5}^1\psi_5(\boldsymbol{t}^5) := \frac{\partial\psi_5}{\partial t_1^5}(\boldsymbol{t}^5) = \frac{\partial\psi_5}{\partial x_4}(x_4, x_5) = \begin{cases} \max\{0, x_5 - 0.71\}, & \text{if } x_4 > 0.26, \\ 0, & \text{otherwise,} \end{cases}$$

$$D_{4,5}^1\psi_5(\boldsymbol{t}^5) := \frac{\partial\psi_5}{\partial t_2^5}(\boldsymbol{t}^5) = \frac{\partial\psi_5}{\partial x_5}(x_4, x_5) = \begin{cases} \max\{0, x_4 - 0.26\}, & \text{if } x_5 > 0.71, \\ 0, & \text{otherwise;} \end{cases}$$

$$|\,\boldsymbol{\alpha}| = 2 : D_{4,5}^2\psi_5(\boldsymbol{t}^5) := \frac{\partial^2\psi_5}{\partial t_1^5 \partial t_2^5}(\boldsymbol{t}^5) = \frac{\partial^2\psi_5}{\partial x_4 \partial x_5}(x_4, x_5) = \begin{cases} 1, & \text{if } x_5 > 0.71, \\ 0, & \text{otherwise.} \end{cases}$$

The PRSS objective function in (3.9) has the following form:

$$PRSS = \sum_{i=1}^{32} \left(y_i - \boldsymbol{\theta}^T \boldsymbol{\psi}(\tilde{\boldsymbol{d}}_i) \right)^2 \tag{3.37}$$

$$+ \sum_{m=1}^{5} \lambda_m \sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s \in V_m}} \int \theta_m^2 \left[D_{r,s}^{\boldsymbol{\alpha}} \psi_m(\boldsymbol{t}^m) \right]^2 d\boldsymbol{t}^m. \tag{3.38}$$

If $\lambda_1 = \lambda_2 = ... = \lambda_{M_{max}} =: \lambda$, then PRSS looks like a Tikhonov regularization problem as follows:

$$PRSS \approx \left\| \boldsymbol{y} - \boldsymbol{\psi}(\tilde{\mathbf{d}})\boldsymbol{\theta} \right\|_2^2 + \lambda \left\| \boldsymbol{L}\boldsymbol{\theta} \right\|_2^2 \ . \tag{3.39}$$

Again, as in the previous example, we observe that the first parts of the PRSS objective function (3.37) and the Tikhonov regularization problem (3.39) are equal to each other. However, second parts are only approximately equal.

$$\sum_{i=1}^{32} \left(y_i - \boldsymbol{\theta}^T \boldsymbol{\psi}(\tilde{\boldsymbol{d}}_i) \right)^2 = \left\| \boldsymbol{y} - \boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta} \right\|_2^2,$$

$$\sum_{m=1}^{5} \lambda_m \sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s \in V_m}} \int \theta_m^2 \left[D_{r,s}^{\boldsymbol{\alpha}} \psi_m(\boldsymbol{t}^m) \right]^2 d\boldsymbol{t}^m \approx \lambda \left\| \boldsymbol{L}\boldsymbol{\theta} \right\|_2^2 .$$

The RSS values are presented below (the complete form of RSS can be seen in Appendix A):

$$\begin{aligned} \sum_{i=1}^{32} \left(y_i - \boldsymbol{\theta}^T \boldsymbol{\psi}(\tilde{\boldsymbol{d}}_i) \right)^2 = (0.013 - \theta_0 - (\max\{0, 0.58 - 2.21\})\theta_1 - \\ (\max\{0, 2.21 - 0.58\})\theta_2 - \\ (\max\{0, 0.25 - 0.26\})\theta_3 - \\ (\max\{0, 1650 - 1601\})(\max\{0, 0.25 - 0.26\})\theta_4 - \end{aligned}$$

$$
\begin{aligned}
&(\max\{0, 0.9 - 0.71\})(\max\{0, 0.25 - 0.26\})\theta_5)^2 + \\
&(0.016 - \theta_0 - (\max\{0, 0.66 - 2.21\})\theta_1 - \\
&(\max\{0, 2.21 - 0.66\})\theta_2 - \\
&(\max\{0, 0.33 - 0.26\})\theta_3 - \\
&(\max\{0, 1650 - 1601\})(\max\{0, 0.33 - 0.26\})\theta_4 - \\
&(\max\{0, 0.9 - 0.71\})(\max\{0, 0.33 - 0.26\})\theta_5)^2 + \\
&(0.015 - \theta_0 - (\max\{0, 0.66 - 2.21\})\theta_1 - \\
&(\max\{0, 2.21 - 0.66\})\theta_2 - \\
&(\max\{0, 0.33 - 0.26\})\theta_3 - \\
&(\max\{0, 1650 - 1601\})(\max\{0, 0.33 - 0.26\})\theta_4 - \\
&(\max\{0, 0.9 - 0.71\})(\max\{0, 0.33 - 0.26\})\theta_5)^2 + \\
&\vdots \\
&(0.068 - \theta_0 - (\max\{0, 18.5 - 2.21\})\theta_1 - \\
&(\max\{0, 2.21 - 18.5\})\theta_2 - \\
&(\max\{0, 1.5 - 0.26\})\theta_3 - \\
&(\max\{0, 1700 - 1601\})(\max\{0, 1.5 - 0.26\})\theta_4 - \\
&(\max\{0, 0.7 - 0.71\})(\max\{0, 1.5 - 0.26\})\theta_5)^2.
\end{aligned}
$$

By computing the maximum functions, the RSS terms has the following form:

$$
\begin{aligned}
\sum_{i=1}^{32} \left(y_i - \boldsymbol{\theta}^T \boldsymbol{\psi}(\tilde{\boldsymbol{d}}_i)\right)^2 &= (0.013 - \theta_0 - 1.63\theta_2)^2 + \\
(0.016 - \theta_0 - 1.55\theta_2 - 0.07\theta_3 - 3.43\theta_4 - 0.0133\theta_5)^2 &+ \\
&\vdots \\
(0.068 - \theta_0 - 16.29\theta_1 - 1.24\theta_3 - 122,76\theta_4)^2 &
\end{aligned}
$$

and writing into vector notation is as follows:

$$
\begin{aligned}
&= (0.013 - \theta_0 - 1.63\theta_2)^T (0.013 - \theta_0 - 1.63\theta_2) + \\
&(0.016 - \theta_0 - 1.55\theta_2 - 0.07\theta_3 - 3.43\theta_4 - 0.0133\theta_5)^T
\end{aligned}
$$

$$(0.016 - \theta_0 - 1.55\theta_2 - 0.07\theta_3 - 3.43\theta_4 - 0.0133\theta_5) +$$
$$(0.015 - \theta_0 - 1.55\theta_2 - 0.07\theta_3 - 3.43\theta_4 - 0.0133\theta_5)^T$$
$$(0.015 - \theta_0 - 1.55\theta_2 - 0.07\theta_3 - 3.43\theta_4 - 0.0133\theta_5) +$$
$$\vdots$$
$$(0.056 - \theta_0 - 10.29\theta_1 - 1.24\theta_3 - 122,76\theta_4)^T(0.056 - \theta_0 - 10.29\theta_1 - 1.24\theta_3 - 122,76\theta_4) +$$
$$(0.068 - \theta_0 - 16.29\theta_1 - 1.24\theta_3 - 122,76\theta_4)^T(0.068 - \theta_0 - 16.29\theta_1 - 1.24\theta_3 - 122,76\theta_4).$$

As shown for the previous data set, we can change the form of the above summation into vector notation and get the representation below. Thus, RSS which is the first part of PRSS is as follows:

$$\begin{aligned}\sum_{i=1}^{32}\left(y_i - \boldsymbol{\theta}^T\boldsymbol{\psi}(\tilde{\boldsymbol{d}}_i)\right)^2 &= \left(y - \boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta}\right)^T\left(y - \boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta}\right)\\ &= \|y - \boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta}\|_2^2.\end{aligned}$$

As mentioned in the previous example, the multi-dimensional integral in the second part of the equation (3.12) takes the form of (3.15) after discretization and this discretized form is denoted by $\boldsymbol{L}$. Then, we reach the formulation in equation (2.41).

The L_m $(m = 1, \ldots, 5)$ values corresponding to BFs ψ_1, ψ_2, ψ_3, ψ_4 and ψ_5 are calculated as follows:

$$\begin{aligned}L_1 &= \sum_{i=1}^{(N+1)^{K_1}}\left[\left(\sum_{\substack{|\boldsymbol{\alpha}|=1\\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2}\sum_{\substack{r<s\\ r,s\in V_1}}\left[D_{r,s}^{\boldsymbol{\alpha}}\left(\max\{0, x_2 - 2.21\}\right)\right]^2\right)\left(\tilde{x}_{l_{\sigma^{\kappa_1^1}}^{\kappa_1^1}+1,\kappa_1^1} - \tilde{x}_{l_{\sigma^{\kappa_1^1}}^{\kappa_1^1},\kappa_1^1}\right)\right]\\ &= 3.9497,\end{aligned}$$

$$\begin{aligned}L_2 &= \sum_{i=1}^{(N+1)^{K_2}}\left[\left(\sum_{\substack{|\boldsymbol{\alpha}|=1\\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2}\sum_{\substack{r<s\\ r,s\in V_2}}\left[D_{r,s}^{\boldsymbol{\alpha}}\left(\max\{0, 2.21 - x_2\}\right)\right]^2\right)\left(\tilde{x}_{l_{\sigma^{\kappa_1^1}}^{\kappa_1^2}+1,\kappa_1^2} - \tilde{x}_{l_{\sigma^{\kappa_1^1}}^{\kappa_1^2},\kappa_1^2}\right)\right]\\ &= 1.5875,\end{aligned}$$

$$\begin{aligned}L_3 &= \sum_{i=1}^{(N+1)^{K_3}}\left[\left(\sum_{\substack{|\boldsymbol{\alpha}|=1\\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2}\sum_{\substack{r<s\\ r,s\in V_3}}\left[D_{r,s}^{\boldsymbol{\alpha}}\left(\max\{0, x_4 - 0.26\}\right)\right]^2\right)\left(\tilde{x}_{l_{\sigma^{\kappa_1^1}}^{\kappa_1^3}+1,\kappa_1^3} - \tilde{x}_{l_{\sigma^{\kappa_1^1}}^{\kappa_1^3},\kappa_1^3}\right)\right]\\ &= 1.1958,\end{aligned}$$

$$
\begin{aligned}
L_4 &= \\
&\sum_{i=1}^{(N+1)^{K_4}} \left[\left(\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_4}} \left[D^{\boldsymbol{\alpha}}_{r,s} \psi_4(\boldsymbol{t}^4) \right]^2 \right) \left(\tilde{x}_{l^{\kappa_1^4}_{\sigma^{\kappa_1}}+1,\kappa_1^4} - \tilde{x}_{l^{\kappa_1^4}_{\sigma^{\kappa_1}},\kappa_1^4} \right) . \left(\tilde{x}_{l^{\kappa_2^4}_{\sigma^{\kappa_2}}+1,\kappa_2^4} - \tilde{x}_{l^{\kappa_2^4}_{\sigma^{\kappa_1}},\kappa_2^4} \right) \right] \\
&= 9.9015,
\end{aligned}
$$

$$
\begin{aligned}
L_5 &= \\
&\sum_{i=1}^{(N+1)^{K_5}} \left[\left(\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_5}} \left[D^{\boldsymbol{\alpha}}_{r,s} \psi_5(\boldsymbol{t}^5) \right]^2 \right) \left(\tilde{x}_{l^{\kappa_1^5}_{\sigma^{\kappa_1}}+1,\kappa_1^5} - \tilde{x}_{l^{\kappa_1^5}_{\sigma^{\kappa_1}},\kappa_1^5} \right) . \left(\tilde{x}_{l^{\kappa_2^5}_{\sigma^{\kappa_2}}+1,\kappa_2^5} - \tilde{x}_{l^{\kappa_2^5}_{\sigma^{\kappa_1}},\kappa_2^5} \right) \right] \\
&= 0.1975.
\end{aligned}
$$

Hence, the $\boldsymbol{L}$ matrix becomes a (6×6)-diagonal matrix as given below:

$$
\boldsymbol{L} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 3.9497 & 0 & 0 & 0 & 0 \\
0 & 0 & 1.5875 & 0 & 0 & 0 \\
0 & 0 & 0 & 1.1958 & 0 & 0 \\
0 & 0 & 0 & 0 & 9.9015 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.1975
\end{bmatrix}.
$$

In the $\boldsymbol{L}$ matrix, the first column elements of $\boldsymbol{L}$ are all zero and the diagonal elements of this matrix $\boldsymbol{L}_m$ $(m = 1, 2, ..., 5)$ are as given above. Then, $\boldsymbol{L\theta}$ is as follows:

$$
\begin{aligned}
\|\boldsymbol{L\theta}\|_2^2 &= (\theta_1 \cdot (3.9497))^2 + (\theta_2 \cdot (1.5875))^2 + (\theta_3 \cdot (1.1958))^2 + (\theta_4 \cdot (9.9015))^2 + \\
&\quad (\theta_5 \cdot (0.1975))^2 .
\end{aligned}
$$

This matrix $\boldsymbol{L}$ is used as an input in MATLAB Tikhonov regularization toolbox and the results are displayed in Section 3.5.

3.5 Validation Approach and Comparison Measures

In our applications, we use two different real-valued continuous data sets. In the first data, there is no interaction between variables, however, in the second one, there

exists interaction between variables. In order to evaluate the performance of Tikhonov regularization, we used various measures [104]. These performance measures and their general notations are as follows;

- y_i is an ith observed response value,
- $\hat{y}_i$ is an ith fitted response,
- $\bar{y}$ is a mean response,
- N is a number of observations,
- p is a number of terms in the model,
- $\bar{\hat{y}}$ is a mean fitted response,
- $s(y)^2$ is a sample variance for observed response,
- $s\hat{(y)}^2$ is a sample variance for observed response,
- $e_i = y_i - \hat{y}_i$ is an ith ordinary residual,
- h_i is a leverage value for the ith observation, which is the ith diagonal element of the hat matrix, $\boldsymbol{H}$. The *hat matrix* is $\boldsymbol{H}=\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T$, where $\boldsymbol{X} : (N\times p)$ design matrix and rank $(\boldsymbol{X}) = p$ $(p \leq N)$.

3.5.1 Comparison Measures

i. Correlation Coefficient r

Correlation coefficient is a measure to explain the linear relationship between the actual and the predicted response values. There exists a strong positive or negative relationship between the actual and the predicted response variables when r is close to either 1 or -1, respectively. If it is zero, there exists no linear association between actual and predicted values [95]. The formula of this measure can be written as follows:

$$r := \frac{\sum_{i=1}^{n} \frac{(y-\bar{y})(\hat{y}-\bar{\hat{y}})}{(n-1)}}{\sqrt{s(y)^2 s(\hat{y})^2}} \quad \text{such that} -1 \leq r \leq +1,$$

with y being the actual dependent variables, $\hat{y}$ being the predicted dependent variables and $\bar{y}$ being the mean of actual values. Here, $s(\hat{y})$ is the standard deviation of predicted response variable and $s(y)$ is the standard deviation of actual response variable.

Here, we expect a high r value so that our estimators are good enough to predict the

response variables as the original ones.

***ii.* Prediction Error Sum of Squares (PRESS)**

PRESS is a measure related with the predictive ability of the model. In fact, it is simply the sum of squares of the prediction error so the smaller the error the better the predictive ability of the model [95]. Its formula has the following form:

$$PRESS := \sum_{i=1}^{n} \left(\frac{e_i}{1 - h_i}^2 \right).$$

***iii.* The Coefficient of Determination R^2**

R^2 is a measure to explain the ability of a model to predict new values. As the value of R^2 increases, the model fit to data gets better [95]. The formula of the coefficient of determination can be represented as follows:

$$R^2 := 1 - \frac{RSS}{SSTotal} = 1 - \left(\frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{N} (y_i - \bar{y}_i)^2} \right).$$

***iv.* Adjusted R^2**

Different from the R^2, this measure adjusts regarding the number of explanatory terms in a model. That's, R^2 can increase when you add a new variable to the model but the *Adjusted* R^2 increases only if the new term improves the model more than would be expected by chance. Therefore, it is beneficial to compare models with different numbers of variables. Besides, the higher the *Adjusted* R^2, the better the model fits data [95]. Its formula is

$$R^2_{Adj} := 1 - \frac{MSError}{MSTotal} = 1 - \left(\frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{N} (y_i - \bar{y}_i)^2} \right) \left(\frac{N - 1}{N - p - 1} \right).$$

***v.* Predicted R^2**

The *predicted*R^2 measures the ability of the model to predict responses for new observations. The predictive ability of the model gets better when the *Predicted* R^2 value

increases [95]. Its formula can be written as follows:

$$R^2(pred) := 1 - \frac{PRESS}{SSTotal} = 1 - \frac{\sum_{i=1}^{N} \left(\frac{e_i}{1-h_i}^2\right)}{1 - \sum_{i=1}^{N}(y_i - \bar{y})^2}.$$

***vi.* Mean Square Error (MSE)**

MSE measures the difference between predicted and actual values. If it is small, it means that the error is small and the estimates are good enough. Therefore, the smaller the MSE, the better it is [95]. Its formula has the following form:

$$MSE := \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i).$$

***vii.* Root Mean Square Error (RMSE)**

RMSE also quantifies the difference between predicted and actual values. It is frequently used to measure the precision of the model. Again, as its value gets smaller, the precision gets better [95]. A model independent formula is

$$RMSE := \sqrt{MSE} = \sqrt{\frac{1}{N-p-1}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}.$$

***viii.* Average Absolute Error (AAE)**

AAE is a quantity used to measure how close predictions are to the actual outcomes. As it is the average magnitude of error, it is better when its value gets smaller [95]. The formula can be represented as follows:

$$AAE := \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|.$$

3.6 Numerical Results of the Tikhonov Regularization Problems

3.6.1 Introduction

In our numerical examples, we used MATLAB programming language (MATLAB R2007a) and Tikhonov regularization toolbox introduced in Section 2.5. In the application, we firstly employ the command *generalized singular value decomposition* ($GSVD$) and use the matrices BF and L as inputs of this command. Moreover, we benefit from the outputs of $GSVD$ in Tikhonov solver (with the command 'Tikhonov') together with the vectors $\boldsymbol{y}$ and λ. Then, this command returns us the unknown regression coefficients.

In this study, as the regularization parameter λ plays a key role in the estimation of coefficients, it should be chosen with care. The L-curve criterion can be used to decide this parameter. The corner of this curve, the point with maximum curvature, corresponds to the the place this parameter should be chosen.

In this study, we run the program many times, each time with a different λ value ($\lambda > 0$), and observe how it changes. For each solution, we calculated the RSS and $\|\boldsymbol{L\theta}\|_2$ values. We decided the range of λ as the points where these two, RSS and $\|\boldsymbol{L\theta}\|_2$ are stabilized. For example; at the upper bound, as alpha increases, $\|\boldsymbol{L\theta}\|_2$ is always zero. As well, in the lower bound of λ, it starts from the point after which $\|\boldsymbol{L\theta}\|_2$ starts to change.

We examined two different types of data; with and without interaction and measured the performance of our solver by using some statistical tools such as ${R^2}_{Adj}$, r, $RMSE$ and AAE.

3.6.2 Numerical Results of the Data with No Interaction

Tikhonov regularization is usually preferred for huge dimension data sets. In this study, we first applied this technique to the data with no interaction. Firstly, we examined it at end and corner λ values. Then, we observe the changes by using the following performance criteria given in the following table:

Table 3.1: The performance results at the end points

	No Interaction		
Measure	λ_{first}	λ_{corner}	λ_{last}
AAE	0.9571	0.9548	3.6566
$R^2{}_{adj}$	0.9264	0.9260	-0.2631
$RMSE$	1.2861	1.2895	5.3269
r	0.9704	0.9704	0.9703

By looking at this table, it is observed that the measure value of RMSE is increasing as the tradeoff parameter λ increases. Correlation coefficient (r) is high for all λ values which means that there is a good linear relationship between actual and predicted y values for all solutions coming from Tikhonov regularization for this data set. Besides, as λ increases, the RMSE gets bigger while the $R^2{}_{Adj}$ decreases. This means that our model gets worse to explain the variation in response variable and does not have a good fit as it has before. This is due to the over regularization as the larger the λ (equivalent to a large amount of regularization), the smaller the solution seminorm at the cost of a large residual norm, while it has a reverse effect for a small λ.

Moreover, we get the following loglog curve for this data set:

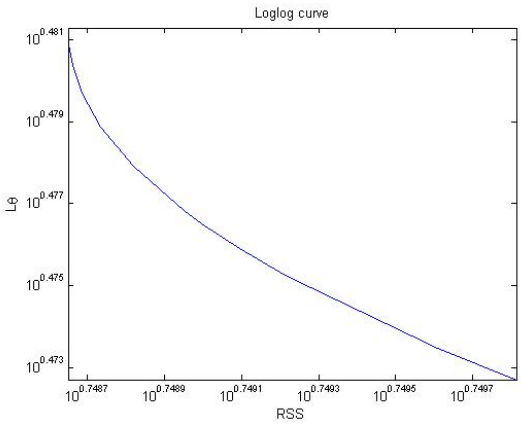

Figure 3.1: The Loglog curve for the data with no interaction

3.6.3 Numerical Results of the Data with Interaction

In this part, we applied the Tikhonov solver to the data with interaction. We examined it at the end and corner λ values, then we observe the following results:

Table 3.2: Tikhonov Regularization for the two data sets

	Interaction		
Measure	λ_{first}	λ_{corner}	λ_{last}
AAE	0.0014	0.0015	0.0079
${R^2}_{adj}$	0.9663	0.96	-0.2072
$RMSE$	0.0022	0.0024	0.0130
r	0.9863	0.9838	0.9318

In this table, the measure value of RMSE and AAE is increasing as the tradeoff parameter λ increases. Even the correlation coefficient (r) is decreasing as λ increases, it is very high for all λ values which means that there is a good linear relationship between actual and predicted y values for all solutions coming from Tikhonov regularization for this data set. Besides, we observe a decreasing ${R^2}_{adj}$. This is due to the over regularization again.

Moreover, we get the following loglog curve for this data set which is:

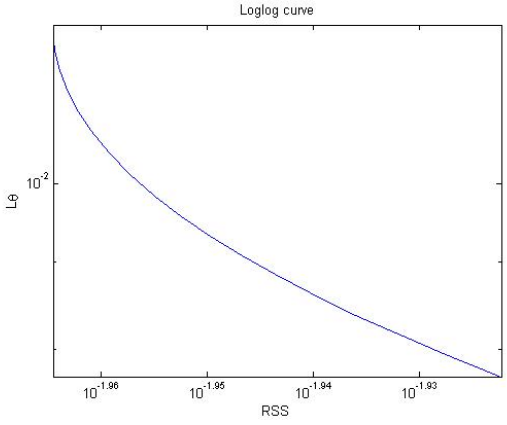

Figure 3.2: The Loglog curve for the data with interaction

3.6.4 Comparison of the Results Regarding Data Types

We can compare results of the two data sets and see which one has better results when Tikhonov regularization is employed. The results are illustrated in the following table:

Table 3.3: Tikhonov Regularization for the two data sets

	Interaction			No Interaction		
Measure	λ_{first}	λ_{corner}	λ_{last}	λ_{first}	λ_{corner}	λ_{last}
AAE	0.0014	0.0015	0.0079	0.9571	0.9548	3.6566
${R^2}_{adj}$	0.9663	0.96	-0.2072	0.9264	0.9260	-0.2631
$RMSE$	0.0022	0.0024	0.0130	1.2861	1.2895	5.3269
r	0.9863	0.9838	0.9318	0.9704	0.9704	0.9703

From Table 3.3, we see that the error rate (AAE and $RMSE$) is lower for data set with interaction than the set without interaction indicating that the accuracy rate is higher for interaction data. Besides, other performance measures (r and ${R^2}_{adj}$) are also higher for the data with interaction. Both data sets have high R^2 criteria values which shows that the model fit is good for both data sets at λ_{first} and λ_{last}.

3.7 IKL Analysis

In Chapter 2.6, we introduced one of classifying techniques used for heterogeneous and large-scale data sets; infinite kernel learning. Here, we apply IKL on some data sets and compare it with Tikhonov regularization. These data sets, Votes, Bupa and Hepatitis, are from the well-known standard UCI machine learning repository[1]. Except the homogeneous data set *Votes*, all are heterogeneous which means that data set includes both discrete and continuous variables. Data descriptions are given in te following table:

Here, in Table 3.4, first column shows the name of the data set, second is the number of data, third is the number of features, and fourth one represents the types of data sets, respectively.

[1] available from $http://archive.ics.uci.edu/ml/$

Table 3.4: Data set description

Data set	# instances	# attributes	attribute characteristics
Votes	52	16	categorical
Bupa	345	6	integer, real and categorical
Hepatitis	155	19	integer, real and categorical

In this study, we focused on PCRM and PEM algorithms with 5-fold cross-validation and used Knitro solver on normalized data sets. Besides, *active set*, *interior CG* and *interior direct* are the algorithms of the solver Knitro. The results are displayed in Table B.1 in Appendix B.

In Table B.1, the values of Mean AUC, Std Dev Error and Mean Error show that there are small differences between two methods except that the run time of PEM is longer than that of PCRM.

Moreover, we like to see how the performance of Tikhonov regularization and IKL differs compared to each other. In order to observe this, we applied them on two data sets; Votes and Hepatitis, and compared the methods over these data sets. In the following part, we share the results of this comparison.

3.8 Comparison of the Results for Tikhonov Regularization and IKL

In previous parts, we make separate analysis using Tikhonov regularization and IKL on different data sets. However, in this part, we focus on the comparison of these methods and for this aim, we also applied Tikhonov regularization on Votes and Hepatitis data sets. We share the results in Table 3.8.

In both techniques, normalized data sets are used and a 5-fold cross validation is applied. In IKL analysis, we used *Primal Conceptual Reduction Method* ($PCRM$) and active set algorithm for both data sets.

We make the comparison over some performance measures which are automatically obtained from IKL toolbox; Mean Error rate, Std Dev Error and Mean AUC. Here, the Std Dev Error is the standard deviation of errors over 5-fold cross-validation. However, for Tikhonov regularization we calculated these values by hand so that we

can compare them over the same scales. The results are displayed in the following table:

	Votes		Hepatitis	
Measure	IKL	TIKHONOV	IKL	TIKHONOV
Mean Error	0.2091	0.0020	0.1936	0.0019
Std Dev Error	0.1469	0.0010	0.0456	0.0003
Mean AUC	0.81	0.65	0.64	0.98

Table 3.5: Comparison of the methods IKL and Tikhonov Regularization

Here, AUC denotes the true positive rate and as AUC tends to 1, the prediction accuracy gets better [24]. From table above, it is seen that Tikhonov solver performs a lower error rate for $Votes$ data. However, prediction accuracy is smaller than that of IKL. For the Hepatitis data set, on the other hand, Tikhonov solver has a higher prediction accuracy with a smaller error rate. Thus, Tikhonov regularization gives more accurate results for Hepatitis data set than IKL.

Performance measures are closer in each method, however, Table 3.8 shows that Tikhonov regularization provides a better accuracy for the heterogeneous data Hepatitis while IKL displays a better performance for the homogenous data Votes.

CHAPTER 4

CONCLUSION AND FUTURE RESEARCH

The search to define the relationship between a response variable and its predictors and modeling it is a commonly studied field. Regression analysis and classification techniques are just some of them. In this thesis, both methods, a regression and a classification, are studied.

We analysis *Generalized Partial Linear Models* (*GPLMs*) in which there is a single nonparametric component together with usual parametric terms. Indeed, GPLM decomposes input variables into two sets and additively combines classical linear models with nonlinear model part. Thus, it has a great advantage that consists in this *grouping* which could be done for the input dimensions or features in order to assign appropriate submodels specifically [92].

In this thesis, we combined GPLM with a modified form of MARS, named as *Conic Multivariate Adaptive Regression Splines* ($CMARS$). We propose to use *penalized residual sum of squares* ($PRSS$) to control complexity and accuracy of the model instead of the backward algorithm. Then, we turn this PRSS function into a *Tikhonov regularization problem* and solve it by using the regularization toolbox of MATLAB.

As well as studying the regularization of the nonparametric part, we also mentioned theoretically the regularization of the parametric part. However, in the numerical example, we disregard the parametric part for the sake of simplicity. We provided two numeric examples by using two different data type; one has interaction between variables and the other does not have.

In the numerical example, we observed that Tikhonov solver gave better results for

both data at the first parameter values. At that points, estimation errors were small and adjusted R^2s were high which show a good model fit for both data. However, as we increased the parameter value, we saw that error was increasing while model fit was decreasing. Besides, compared two data with each other. For the data with interaction, estimation error was smaller and adjusted R^2 was higher than that of data with no interaction at all parameter values. Actually, this is expected because CMARS gives better results for huge and complex data sets.

Furthermore, we made an analysis by using a modern method of Machine Learning tool, *Infinite Kernel Learning* (IKL). To observe how this method differs from CMARS with Tikhonov regularization, we compared their results on two data sets; homogeneous data set Votes and heterogeneous data set Hepatitis. The data are different in that heterogeneous data set includes both discrete and continuous variables while homogeneous data set can include just one type. Besides, to compare the two methods, we used some statistical performance measures; Mean Error, Std Dev Error and Mean AUC which are automatically achieved from IKL toolbox. Then, we calculated these values for Tikhonov regularization and compared the two methods. Numerical results of these two data sets show that IKL gives more accurate results for the homogenous data set, Votes. Even its error is bigger, it has a bigger Mean AUC value than that Tikhonov solver. However, for the nonhomogeneous data set Hepatitis, Tikhonov solver has better results. It has a smaller error rate with a higher Mean AUC value, showing that prediction accuracy of Tikhonov regularization is better for the heterogeneous data set, Hepatitis.

In the following, there can be done more studies on the comparison of IKL and CMARS. Moreover, future analysis and if possible, partial combination of GPLMs and IKL can be studied.

In this thesis, we focus on Generalized Partial Linear Model and CMARS with Tikhonov regularization. We analyze several data sets and display comparisons. It is also possible to do more studies on the comparison of CMARS by using Tikhonov solver and *conic quadratic programming* CQP [69], an optimization technique. As CQP can employ the structure of the problem and allow better complexity bounds for the generic problems, it can exhibit a much better practical performance [104]. Moreover, as well

as comparing the performance of CMARS with Tikhonov and CQP each other, there can be made comparisons together with IKL on many data sets.

Furthermore, as a future study, a further analysis and algorithmical development of the special subclass of GPLMs of this thesis can be done. In the near future, the utilization of these results and further implementations of the methods to various application areas is possible. Besides, identification and investigation of further important model subclasses of GPLMs can be searched.

REFERENCES

[1] S.Z. Alparslan Gök and G.-W. Weber, *Cooperative games under ellipsoidal uncertainty*, in the proceedings of PCO 2010, 3rd Global Conference on Power Control and Optimization, February 2-4, 2010, Gold Coast, Queensland, Australia (ISBN: 978-983-44483-1-8).

[2] E.J. Anderson and P. Nash, *Linear Programming in Infinite-Dimensional Spaces*, John Wiley and Sons Ltd, 1987.

[3] R.C. Aster, B. Borchers, and C. Thurber, *Parameter Estimation and Inverse Problems*, MA:AcademicPress, Burlington, 2004.

[4] F.R. Bach and G.R.G. Lanckriet, *Multiple kernel learning, conic duality, and the smo algorithm*, In Proceedings of the 21st International Conference on Machine Learning, 2004.

[5] L.J. Bain and M. Engelhardt, *Introduction to Probability and Mathematical Statistics*, Duxbury, Thomas Learning, California, 1991.

[6] B. Bakır, *Defect Cause Modelling With Decision Tree and Regression Analysis: A Case Study in Casting Industry*, Master Thesis, METU, Ankara, 2006.

[7] A. Ben-Tal, *Conic and Robust Optimisation, Lecture Notes for the Course*, Minerva Optimisation Center, Technion - Israel Institute of Technology, 2002.

[8] C. de Boor, *A Practical Guide to Splines*, Springer-Verlag, 1978.

[9] L. Breiman, J. H. Friedman, R. Olshen and C. Stone, *Classification and Regression Trees, Belmont*, Classification, CA: Wadswort Int. Group, 1984.

[10] E. Buyukbebeci, *Comparison of MARS, CMARS and CART in Predicting Default Probabilities for Emerging Markets*, METU, Ankara, 2009.

[11] L. Chen, J. Song and F. Ji, *MARS-based Research of Personal Credit Scoring: Verification of Chinese Data, Management Science and Engineering*, International Conference on; Lille, France, 2006.

[12] S.M. Chou, T.S. Lee, Y.E. Shao and I.F. Chen, *Mining the breast cancer pattern using artificial neural networks and multivariate adaptive regression splines*, in: Expert Systems with Applications, Vol. 27, 2004.

[13] Copyright StatSoft, Inc., *Multivariate Adaptive Regression Splines*, http://www.statsoft.com/textbook/stmars.html, accessed 25 Nov. 2009.

[14] P. Craven and G. Wahba, *Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross-validation*, in: Numerische Mathematik, Vol. 31, 1979, 377-403.

[15] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge, UK: Cambridge University Press, 2000.

[16] C.S. Davis, *Statistical Methods for the Analysis of Repeated Measures*, New York, NY: Springer-Verlag, 2003.

[17] E. Deconinck, D. Coomons and Y.V. Heyden, *Explorations of linear modelling techniques and their combinations with multivariate adaptive regression splines to predict gastro-intestinal absorption of drugs*, in: Journal of Pharmaceutical and Biomedical Analysis, Vol. 43, 2007, 119-130.

[18] J. Deichmann, A. Eshghi, D. Haughton, S. Sayek and N. Teebagy, *Application of multiple adaptive regression splines (MARS) in direct response modeling*, Journal of Direct Marketing, 16, 4 (2002) 15-27.

[19] W. Di, *Long Term Fixed Mortgage Rate Prediction Using Multivariate Adaptive Regression Splines*, School of Computer Engineering, Nanyang Technological University, 2006.

[20] J.J. Dongarra, J.R. Bunch, C.B. Moler and G.W. Stewart, *Linpack User's Guide*, SIAM, Philadelphia, 1979.

[21] S. Dowdy and S. Wearden, *Statistics for Research*, New York: Wiley, 1983

[22] J. Elith and J. Leathwick, *Predicting species distribution from museum and herborium records using multiresponse models fitted with multivariate adaptive regression splines, in: Diversity and Distributions*, Vol. 13, 2007, 265-275.

[23] R.L. Eubank, *Nonparametric Regression and Spline Smoothing*, 2nd Ed. (1999) New York: Marcel, Dekker, Inc.

[24] P.A. Flach. *The many faces of roc analysis in machine learning.* In The Twenty-First International Conference on Machine Learning, 2004.

[25] J.H. Friedman, *Multivariate adaptive regression splines*, The Annals of Statistics, Vol. 19, 1991, 1-141.

[26] L.M. Fu, *Neural Networks in Computer Intelligence*, McGraw-Hill, Inc.,New York, NY, USA, 1994.

[27] P.J. Green and B.S. Yandell, *Semiparametric Generalized Linear Models*, Lecture Notes in Statistics, 1985, 32.

[28] S. Greenland, *Dose-response and trend analysis in epidemiology: alternatives to categorical analysis*, Epidemiology 6(4): 356-365, (1995).

[29] C.W. Groetsch, *The Theory of Tikhonov Regularization for Fredholm Equations of the First Kind*, Pitman, Boston, 1984.

[30] C. Gu, *Smoothing Spline ANOVA Models*, New York: Springer-Verlag New York (2002), Inc.

[31] H. Haas and G. Kubin, *A multi-band nonlinear oscillator model for speech, Conference Record of the Thirty- Second Asilomar Conference on Signals*, Systems and Computers, Vol. 1, 1998, pp. 338-342.

[32] M. Hansen and C. Kooperburg, *Spline Adaptation in Extended Linear Models*, Statistic Science 17.1 (2002).

[33] P.C. Hansen, *Analysis of discrete ill-posed problems by means of the L-curve*, SIAM Rev., SIAM Review. A Publication of the Society for Industrial and Applied Mathematics, volume 34, 1992, number 4, Jin Xi Zhao.

[34] P.C. Hansen, *Rank-deficient and discrete ill-posed problems: Numerical aspects of linear inversion*, SIAM Monographs on Mathematical Modeling and Computation, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998, Jin Xi Zhao.

[35] P.C. Hansen, *Regularization tools: A Matlab package for analysis and solution of discrete ill-posed problems*, Numer. Algorithms, 6 (I-II):1-35, 1994.

[36] P.C. Hansen, *Relations between SVD and GSVD of discrete regularization problems in standard and general form*, Linear Algebra and Its Applications, 1990.

[37] P.C. Hansen and D.P. O'Leary, *The use of the L-curve in the regularization of discrete ill-posed problems*, SIAM J. Sci. Comput., 14, 6 (1993).

[38] T.J. Hastie and R.J. Tibshirani, *Generalized Additive Models*, Chapman and Hall Ltd., New York, 1990.

[39] T.J. Hastie, R.J. Tibshirani and J. Friedman, *The Elements of Statistical Learning, Data Mining, Inference and Prediction*, New York, NY: Springer, 2001.

[40] R. Hettich and H.Th. Jongen. *Semi-infinite programming: conditions of optimality and applications*, In J. Stoer, editor, Optimization Techniques 2, Lecture notes in Control and Information Sci. Springer, Berlin, Heidelberg, New York, 1978.

[41] R. Hettich and O. Kortanek. *Semi-infinite programming: Theory, Methods and Applications*, SIAM Review, 35, 1993.

[42] R.J. Hildeman and H.J. Hamilton, *Applying objective interestingness measures for ranking discovered knowledge*, in: Zighed, D.A., Komorowski, J., Zytkow, J. (Eds.), *Proceedings of the 4th European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD'00)*, Lyon, France, Lecture Notes in Computer Science. Springer-Verlag, 2000, 432-439.

[43] R.J. Hildeman and H.J. Hamilton, *Evaluation of interestingness measures for ranking discovered knowledge*, in: Cheung, G.J., Williams, G.J., Li, Q. (Eds.), *Proceedings of the 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'01)*, Hong Kong, Lecture Notes in Computer Science. Springer-Verlag, 2001, 247-259.

[44] *http://en.wikipedia.org/wiki/Supportvectormachines* accessed 5 Apr. 2010.

[45] D. Hurley, J. Hussey, R. McKeown and C. Addy, *An Evaluation of Splines in Linear Regression*, Paper 147-31, http://www2.sas.com/proceedings/sugi31/147-31.pdf accessed 6 March 2010.

[46] O. Ivanciuc, *Applications of Support Vector Machines in Chemistry*, In Rewievs in Computational Chemistry, Volume 23, Eds.: K.B. Lipkowitz and T.R. Cundari. Wiley-VCH, Weinheim, 2007, pp. 291-400.

[47] E. Kartal, *Metamodelling Complex systems Using Liner and Nonlinear Regression Methods*, Master Thesis, METU, Ankara, 2007.

[48] M. Ko and K.M.O. Bryson, *Reexamining the impact of information technology investment an productivity using regression tree and multivariate adaptive regression splines (MARS)*, in: Information Technology and Management, Vol. 9, Springer Netherlands, 2008.

[49] I. Kolyshkina, S. Wong and S. Lim, *Enhancing Generalized Linear Models with Data Mining*, www.casact.org/pubs/dpp/dpp04/04dpp279.pdf, accessed 15/10/2009.

[50] D. Krawczyk-stando and M. Rudnicki, *Regularization Parameter Selection In Discrete Ill-Posed Problems -The Use of the U-Curve*, Int. J. Appl. Math. Comput. Sci.,17, 2 (2007), 157-164.

[51] D. Krawczyk-stando and M. Rudnicki, *The Use of L-Curve and U-Curve in Inverse Electromagnetic Modelling*, Intelligent Computer Techniques in Applied Electromagnetics, Series: Studies in Computational Intelligence, Berlin / Heidelberg, Volume 119/2008.

[52] M. Kriner, *Survival Analysis with Multivariate adaptive Regression Splines*, 2007, Dissertation, LMU Mnchen: Faculty of Mathematics, Computer Science and Statistics.

[53] E. Kropat, G.W. Weber and C.S. Pedamallu, *Regulatory networks under ellipsoidal uncertainty - optimization theory and dynamical systems*, preprint at Institute of Applied Mathematics, METU, submitted to SIAM Journal on Optimization (SIOPT).

[54] C. Lanczos, *Linear Differential Operators*, Dover, Mineola, NewYork, 1997.

[55] C.L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, Engle wood Cliffs, NJ:Prentice-Hall (1974).

[56] T.S. Lee, C.C. Chiu, Y.C. Chou and C.J. Lu, *Mining the customer credit using classification and regression tree and multivariate adaptive regression splines*, in: Computational Statistics and Data Analysis, Vol. 50, 2006.

[57] W. Mandehall and T. Sincich, *Statistics for Engineering and The Sciences*, New Jersey: Prentice Hall, 1995.

[58] MARS from Salford Systems,
http://www.salfordsystems.com/mars/phb (accessed 25 Aug. 2009).

[59] W.L. Martinez and A.R. Martinez, *Computational Statistics Handbook with MATLAB*, London: Chapman and Hall, CRC, 2002.

[60] MATLAB Version 7.5 (R2007b)

[61] P. McCullagh and J.A. Nelder, *Generalized Linear Models*, Chapman and Hall, London, 1989.

[62] G.G. Moisen, and T.S. Frescino, Comparing five modelling techniques for predicting forest characteristics, Ecological Modelling, (2002) 209-225.

[63] D.C. Montgomery, *Design and Analysis of Experiments*, Fifth, John Wiley & Sons Inc., New York, NY, 2001.

[64] M. Müller, *Estimation and testing in generalized partial linear models—a comparative study*, Stat. Comput., Statistics and Computing, volume 11, 2001, 4, 299–309, STACE3, Database Expansion Item.

[65] M. Müller, *Generalized Linear Models*, Statistics and Computing, Stat. Comput., 11, 2004, 4, 591–619, 0960-3174, STACE3, Database Expansion Item.

[66] R.H. Myers and D.C. Montgomery, *Response surface methodology: Process and Product Optimization Using Designed Experiments*, Wiley Series in Probability and Statistics, Second edition, John Wiley & Sons Inc., New York: Wiley, 2002.

[67] M.T. Nair, M. Hegland and R. S. Anderssen, *The Trade-off between Regularity and Stability in Tikhonov Regularization*, Mathematics of Computation, 66, 217 (1997).

[68] A. Nemirovski, *A lectures on modern convex optimisation*, Israel Institute of Technology, 2002.
http://iew3.technion.ac.il/Labs/Opt/opt/LN/Final.pdf (accessed 26 Aug. 2009).

[69] A. Nemirovski, *Five Lectures On Modern Convex Optimization (2002)*, http://iew3.technion.ac.il/Labs/Opt/opt/LN/Final.pdf, accessed 15 Oct. 2009.

[70] Y.E. Nesterov and A.S. Nemirovskii, *Interior Point Methods in Convex Programming*, SIAM, 1993.

[71] Y.E. Nesterov and A.S Nemirovski, *Interior Point Polynomial Algorithms in Convex Programming*, SIAM, 1994.

[72] J. Neter, M. Kutner, W. Wasserman and C. Nachtsheim, *Applied Liear Statistical Models*, Boston, MA: WCB/McGrawHill, 1996.

[73] Nist/Sematech, *e-Handbook of Statistical Methods*,
http://www.itl.nist.gov/div898/handbook/pmd/section4/pmd43.htm and
http://www.itl.nist.gov/div898/handbook/apr/section4/apr412.htm, accessed 15/10/2009.

[74] J.M. Ortega and W.C. Rheinboldt, *Iterative solution of nonlinear equations in several variables*, Academic Press, New York, 1970.

[75] A. Özmen, G-W. Weber and I. Batmaz, *The New Robust CMARS (RCMARS) Method*, to appear in the proceedings of XXII Mini EURO Conference MEC-EurOPT, Izmir, Turkey, June 23-26, 2010.

[76] S. Özöğür-Akyüz and G.-W. Weber, *Learning within finitely many kernels via semi-infinite programming*, ISI Proceedings of 20th Mini-EURO Conference Continuous Optimization and Knowledge-Based Technologies (Neringa, Lithuania, May 20-23, 2008) 342-348.

[77] S. Özöğür-Akyüz and G.-W. Weber, *Infinite kernel learning via infinite and semi-infinite programming*, Optimization Methods and Software, 25: 6, 937-970, 2010.

[78] S. Özöğür-Akyüz and G.-W. Weber, *Modelling of kernel machines by infinite and semi-infinite programming*, In Proceedings of the Second Global Conference on Power Control and Optimization, AIP Conference Proceddings 1159, Bali, Indonesia, Subseries: Mathematical and Statistical Physics; A.H. Hakim, P. Vasant and N. Barsoum, guest eds., 1-3 June 2009.

[79] S. Özöğür-Akyüz and G.-W. Weber *On numerical optimization theory of infinite kernel learning*, Journal of Global Optimization, 2009.

[80] D.J. Poirier, *The Econometrics of Structural Change with Special Emphasis on Spline Functions*, New York: North-Holland Publishing Co. (1976).

[81] J. Renegar, *Mathematical View of Interior Point Methods in Convex Programming*, Society for Industrial and Applied Mathematics (SIAM), 2000.

[82] P.J. Rousseeuw and A.M. Leroy, *Robust regression and outlier detection*, Hoboken, N.J: Wiley-Interscience, 2003.

[83] S. Sonnenburg, G. Rätsch, C. Schafer and B. Schölkopf. *Large scale multiple kernel learning*, J. Machine Learning Research, 2006.

[84] *SPSS 16.0 GPL Reference Guide*, Chicago, IL: SPSS Inc, 2007. http://support.spss.com/ProductsExt/SPSS/Documentation/SPSSforWindows/ accessed 26 Ags. 2009.

[85] R.E. Steuer, *Multiple Criteria Optimisation: Theory, Computation and Application*, New York: John Wiley and Sons, NY, 1986.

[86] G. Still, *Semi-infinite programming: An introduction, preliminary version*, Technical report, University of Twente Department of Applied Mathematics, Enschede, The Netherlands, 2004.

[87] P. Taylan, F. Yerlikaya-Özkurt and G.-W. Weber, *Parameter Estimation For Semiparametric Models with CMARS and its applications*, working paper, IAM, METU, 2010.

[88] P. Taylan and G.-W. Weber, *New approaches to regression in financial mathematics by additive models*, Journal of Computational Technologies, 12, 2 (2007), 3-22

[89] P. Taylan, G.-W. Weber and A. Beck, *New approaches to regression by generalized additive models and continuous optimization for modern applications in finance, science and technology*, Optimization, Optimization. A Journal of Mathematical Programming and Operations Research, 56, 2007, 675-698.

[90] P. Taylan, G.-W. Weber and F. Yerlikaya, *A new approach to multivariate adaptive regression spline by using Tikhonov regularization and continuous optimization*, to appear in TOP (the Operational Research journal of SEIO (Spanish Statistics and Operations Research Society), Selected Papers at the Occasion of 20th EURO Mini Conference *Continuous Optimization and Knowledge-Based Technologies* (Neringa, Lithuania, May 20-23, 2008).

[91] P. Taylan, G.-W. Weber and F. Yerlikaya, *Continuous optimization applied in MARS for modern applications in finance, science and technology*, in the ISI Proceedings of 20th Mini-EURO Conference *Continuous Optimisation and Knowledge-Based Technologies*, (Neringa, Lithuania, May 20-23, 2008, 317-322).

[92] P. Taylan, G.-W. Weber, L. Liu and F. Yerlikaya-Özkurt, *On foundations of parameter estimation for Generalized Partial Linear Models with B-Splines and Continuous Optimization* to appear in journal Computers and Mathematics with Applications.

[93] The MOSEK optimization toolbox for MATLAB manual. Version 5.0 (Revision 105). www.mosek.com accessed 29 March 2010

[94] J.C.C. Tsai and V.C.P. Chen, *Flexible and robust implamentations of multivariate adaptive regression splines within a wastewater treatment stochastic dynamic program*, Quality and Reliability Engineering International, Vol. 21, 2005, 689-699.

[95] G. Upton and I. Cook, *The Dictionary of Statistics*, Oxford University Press Inc., New York, 2008.

[96] A.I.F. Vaz, E.M.G.P. Fernandes, and M.P.S.F. Gomes, *Discretization methods for semiinfinite programming*, *Investigac ăo Operacional*, 21 (1), 2001.

[97] R.D. Veaux, D.C. Psichogios and L. H. Ungar, *A Comparison of Two Nonparametric Schemes: MARS and Neural Networks*, Computers in Chemical Engineering, 17, (1993).

[98] G.-W. Weber, R. Branzei and S.Z. Alparslan Gök, *On cooperative ellipsoidal games*, to appear in the ISI Proceedings of 24th MEC-EurOPT 2010 - Continuous Optimization and Information-Based Technologies in the Financial Sector, Izmir, Turkey, June 23-June 26, 2010.

[99] G.-W. Weber, R. Branzei and S.Z. Alparslan Gök, *On the ellipsoidal core for cooperative games under ellipsoidal uncertainty*, submitted to the proceedings of 2nd International Conference on Engineering Optimization (Lisbon, Portugal, September 6-9, 2010).

[100] G.-W. Weber, B. Akteke-Öztürk, A. İşcanoğlu, S. Özöğür and P. Taylan, *Data mining: clustering, classification and regression*, four lectures given at the Graduate Summer School on New Advances in Statistics, August 11-24, 2007, Middle East Technical University, Ankara, Turkey.

[101] G.-W. Weber, *Generalized Semi-Infinite Optimization and Related Topics*, volume 29 of Research and Exposition in Mathematics. Heldermann Verlag, Germany, 2003.

[102] G.-W. Weber, P. Taylan, D. Sezer, G. Koksal, I. Batmaz, F. Yerlikaya, S. Ozogur, J. Shawe-Taylor, F. Ozbudak and E. Akyildiz, *New Pathways of Research at IAM of METU and Collaboration Proposed - MARS - SVM with Infinitely Many Kernels, Coding Theory and Cryptography Indicated*, seminar presentation, distributed at Technion, Israel Institute of Technology, Haifa, Israel, January 20-25, 2008.

[103] S.N. Wood, *Generalized additive models*, Texts in Statistical Science Series, An Introduction with *R*, Chapman & Hall/CRC, Boca Raton, FL, 2006.

[104] F. Yerlikaya, *A New Contribution to Nonlinear Robust Regression and Classification with MARS and Its Application to Data Mining for Quality Control in Manufacturing*, MSc. Thesis at the Institute of Applied Mathematics of METU, Ankara, 2008.

[105] H. Zareipour, K. Bhattacharya, and C.A. Canizares, Forecasting the hourly Ontario energy price by multivariate adaptive regression splines, IEEE, Power Engineering Society General Meeting, 2006.

[106] Y. Zhou and H. Leung, *Predicting object-oriented software maintainability using multivariate adaptive regression splines*, in: Journal of Systems and Software, Vol. 80, 2007, 1349-1361.

APPENDIX A

RSS in Numerical Examples

When the maximum functions are computed, the terms of the RSS with a tabular form are as follows:

Table A.1: Function RSS became addressed in Subsection 3.4.1

	Y	θ_0	θ_1	θ_2	θ_3	θ_4
d_1	13.6	1	0	0.01	2.9	0
d_2	16.6	1	1.89	0	3.99	0
d_3	23.5	1	15.77	0	17.87	0
d_4	10.20	1	0	6.11	0	4.01
d_5	5.4	1	0	10.01	0	7.91
d_6	15	1	0.89	0	2.99	0
d_7	9	1	0	5.31	0	3.21
d_8	12.3	1	0	1.71	0.39	0
d_9	16.3	1	2.49	0	4.59	0
d_{10}	15.4	1	0.79	0	2.79	0
d_{11}	13	1	0	0.41	1.69	0
d_{12}	14.4	1	0.99	0	3.09	0
d_{13}	10	1	0	6.31	0	4.21
d_{14}	10.2	1	0	2.71	0	0.61
d_{15}	9.5	1	0	5.11	0	3.01
d_{16}	1.5	1	0	13.11	0	11.01
d_{17}	18.5	1	2.89	0	4.99	0
d_{18}	12.6	1	0	1.31	0.79	0
d_{19}	17.5	1	1.69	0	3.79	0
d_{20}	4.9	1	0	9.61	0	7.51
d_{21}	15.9	1	0.39	0	2.49	0
d_{22}	8.5	1	0	6.81	0	4.71
d_{23}	10.6	1	0	5.51	0	3.41
d_{24}	13.9	1	1.09	0	3.19	0
d_{25}	14.9	1	0	2.11	0	0.01

When the maximum functions are computed, the terms of the RSS with a tabular form are as follows:

Table A.2: Function RSS became addressed in Subsection 3.4.2

	Y	θ_0	θ_1	θ_2	θ_3	θ_4	θ_5
d_1	0.13	1	0	1.63	0	0	0
d_2	0.016	1	0	1.55	0.07	3.43	0.0133
d_3	0.015	1	0	1.55	0.07	3.43	0.0133
d_4	0.016	1	0	1.55	0.07	3.43	0.0168
d_5	0.015	1	0	1.55	0.07	0	0.0203
d_6	0.016	1	0	1.55	0.07	0	0.0203
d_7	0.014	1	0	1.21	0.24	11.76	0.0216
d_8	0.021	1	0	1.04	0.32	15.68	0.0288
d_9	0.018	1	0	1.04	0.32	15.68	0.0288
d_{10}	0.019	1	0	1.04	0.32	15.68	0.0288
d_{11}	0.021	1	0	1.04	0.32	15.68	0.0608
d_{12}	0.019	1	0	1.04	0.32	15.68	0.0608
d_{13}	0.021	1	0	1.04	0.32	15.68	0.0608
d_{14}	0.025	1	0	1.01	0.84	41.16	0.0756
d_{15}	0.025	1	0	0.21	0.74	36.26	0.0666
d_{16}	0.026	1	0	0.21	0.84	41.16	0.0756
d_{17}	0.024	1	0	0.01	0.84	41.16	0.0756
d_{18}	0.025	1	0	0.01	0.84	41.16	0.0756
d_{19}	0.024	1	0	0.01	0.84	41.16	0.0756
d_{20}	0.025	1	0	0.01	0.84	41.16	0.1596
d_{21}	0.027	1	0	0.01	0.84	41.16	0.1596
d_{22}	0.026	1	0	0.01	1.24	60.76	0.2356
d_{23}	0.029	1	0.79	0	1.24	60.76	0.1116
d_{24}	0.03	1	0.79	0	1.24	60.76	0
d_{25}	0.028	1	0.79	1.24	1.24	60.76	0.0496
d_{26}	0.032	1	0.79	0	1.4	68.6	0.196
d_{27}	0.033	1	1..12	0	1.24	60.76	0.1116
d_{28}	0.039	1	1.79	0	1.24	122.76	0
d_{29}	0.04	1	1.79	0	1.24	60.76	0
d_{30}	0.035	1	1.79	0	1.24	60.76	0.1736
d_{31}	0.056	1	10.29	0	1.24	122.76	0
d_{32}	0.068	1	16.29	0	1.24	122.76	0

APPENDIX B

IKL Analysis of Three Data Sets

Figure B.1: Results of normalized data sets

DATA	METHOD	ALGORITHM	MEAN ERROR	STD DEV ERROR	MEAN AUC	RUN TIME
VOTES	PCRM	Active Set	0,2091	0,1469	0,8080	23min 12 sec
		Conjugate Gradient	0,2473	0,1011	0,7851	22 min 54 sec
		Primal-Dual Interior Point	0,2873	0,0591	0,7405	22min 35 sec
	PEM	Active Set	0,2527	0,0959	0,7500	65min 33 sec
		Conjugate Gradient	0,2091	0,1385	0,8170	64 min 31 sec
		Primal-Dual Interior Point	0,2073	0,1138	0,8110	64 min 32 sec
BUPA	PCRM	Active Set	0,4783	0,0571	0,4699	197min 31sec
		Conjugate Gradient	0,5304	0,0729	0,4515	205min 35sec
		Primal-Dual Interior Point	0,4609	0,0574	0,4906	140min 57 sec
	PEM	Active Set	0,4754	0,0565	0,4828	233min 8 sec
		Conjugate Gradient	0,4638	0,0542	0,4815	245min 36sec
		Primal-Dual Interior Point	0,4841	0,0628	0,4915	229min 55sec
HEPATITIS	PCRM	Active Set	0,1935	0,0456	0,6401	27min 33sec
		Conjugate Gradient	0,2000	0,0421	0,6414	28min 49sec
		Primal-Dual Interior Point	0,2065	0,0540	0,6416	29min 11sec
	PEM	Active Set	0,6774	0,2912	0,3841	79min 52sec
		Conjugate Gradient	0,8258	0,0433	0,3072	78min 25sec
		Primal-Dual Interior Point	0,7032	0,3055	0,3580	79min 23sec