PARAMETER ESTIMATION IN GENERALIZED PARTIAL LINEAR MODELS WITH CONIC QUADRATIC PROGRAMMING

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

BY

GÜL ÇELİK

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 CONIC QUADRATIC PROGRAMMING

submitted by **GÜL ÇELİK** 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 AKYILDIZ _____
Director, Graduate School of **Applied Mathematics**

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

Prof. Dr. Gerhard-Wilhelm WEBER _____
Supervisor, **Institute of Applied Mathematics, METU**

Prof. Dr. Bülent KARASÖZEN _____
Co-supervisor, **Department of Mathematics, METU**

Examining Committee Members:

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

Committee Member 2 Prof. Dr. Gerhard-Wilhelm Weber _____
Institute of Applied 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: GÜL ÇELİK

Signature :

ABSTRACT

PARAMETER ESTIMATION IN GENERALIZED PARTIAL LINEAR MODELS WITH CONIC QUADRATIC PROGRAMMING

Çelik, Gül

M.S., Department of Scientific Computing

Supervisor : Prof. Dr. Gerhard-Wilhelm WEBER

Co-Supervisor : Prof. Dr. Bülent KARASÖZEN

September 2010, 103 pages

In statistics, regression analysis is a technique, used to understand and model the relationship between a dependent variable and one or more independent variables. *Multiple Adaptive Regression Spline* (*MARS*) is a form of regression analysis. It is a non-parametric regression technique and can be seen as an extension of linear models that automatically models non-linearities and interactions. MARS is very important in both classification and regression, with an increasing number of applications in many areas of science, economy and technology.

In our study, we analyzed *Generalized Partial Linear Models* (*GPLMs*), which are particular semiparametric models. GPLMs separate input variables into two parts and additively integrates classical linear models with nonlinear model part. In order to smooth this nonparametric part, we use *Conic Multiple Adaptive Regression Spline* (*CMARS*), which is a modified form of MARS. MARS is very benefical for high dimensional problems and does not require any particular class of relationship between the regressor variables and outcome variable of interest. This technique offers a great

advantage for fitting nonlinear multivariate functions. Also, the contribution of the basis functions can be estimated by MARS, so that both the additive and interaction effects of the regressors are allowed to determine the dependent variable. There are two steps in the MARS algorithm: the forward and backward stepwise algorithms. In the first step, the model is constructed by adding basis functions until a maximum level of complexity is reached. Conversely, in the second step, the backward stepwise algorithm reduces the complexity by throwing the least significant basis functions from the model.

In this thesis, we suggest not using backward stepwise algorithm, instead, we employ a *Penalized Residual Sum of Squares* (*PRSS*). We construct PRSS for MARS as a *Tikhonov Regularization Problem*. We treat this problem using continuous optimization techniques which we consider to become an important complementary technology and alternative to the concept of the backward stepwise algorithm. Especially, we apply the elegant framework of *Conic Quadratic Programming* (*CQP*) an area of convex optimization that is very well-structured, hereby, resembling linear programming and, therefore, permitting the use of interior point methods.

At the end of this study, we compare CQP with Tikhonov Regularization problem for two different data sets, which are with and without interaction effects. Moreover, by using two another data sets, we make a comparison between CMARS and two other classification methods which are *Infinite Kernel Learning* (*IKL*) and Tikhonov Regularization whose results are obtained from the thesis [49], which is on progress.

Keywords: Generalized Partial Linear Models, MARS, CMARS, Tikhonov Regularization, Conic Quadratic Programming

ÖZ

GENELLEŞTİRİLMİŞ PARÇALI DOĞRUSAL MODELLERDE İKİNCİ DERECEDEN KONİK KARESEL PROGRAMLAMA YÖNTEMİ İLE PARAMETRE TAHMİNİ

Çelik, Gül
Yüksek Lisans, Bilimsel Hesaplama Bölümü
Tez Yöneticisi : Prof. Dr. Gerhard-Wilhelm Weber
Ortak Tez Yöneticisi : Prof. Dr. Bülent Karasözen

Eylül 2010, 103 sayfa

İstatistikde, regresyon analizi, bağımlı değişken ve bir veya daha fazla bağımsız değişken arasındaki ilişkiyi anlamak ve modellemek için kullanılan bir yöntemdir. Çok değişkenli uyarlanabilir regresyon eğrileri (MARS), regresyon analizinin bir formudur. MARS parametrik olmayan bir regresyon tekniğidir ve doğrusal olmayan ve etkileşimli modelleri otomatik modelleyen doğrusal modellerin gelişmiş halidir. Hem sınıflandırma hem de regresyonda çok büyük bir öneme sahip olan MARS, ekonomi, bilim ve teknoloji alanında giderek artan bir şekilde uygulanmaktadır.

Bu çalışmada biz, belirli parçalı modeller olan genelleştirilmiş parçalı doğrusal modelleri (GPLMs) inceledik. GPLMs bağımsız değişkenleri iki kısma ayırarak, klasik doğrusal modellerle doğrusal olmayan modelleri eklemeli olarak birleştirir. Doğrusal olmayan kısmı düzenlemek için MARS'ın düzeltilmiş şekli olan konik çok değişkenli uyarlanabilir regresyon eğrilerini (CMARS) kullanmayı amaçlamaktayız. MARS, çok boyutlu problemlerin çözümünde elverişli bir yöntemdir; ve bağımsız değişkenlerle bağımlı değişken arasında belirli bir ilişki biçimi öngörmez. Bu teknik, doğrusal ol-

mayan çok değişkenli fonksiyonlara uygun model oluşturma için büyük bir avantaj sunar. Ayrıca, bağımlı değişkeni tanımlamak için bağımsız değişkenlerin eklemeli ve etkileşimsel katkılarına yer vermektedir. MARS algoritması ekleyerek ve eleyerek ilerleyen iki aşamalı bir algoritmadan oluşmaktadır. İlk aşamada, maksimum karmaşıklık düzeyine ulaşıncaya dek temel fonksiyonlar eklenerek model yapılandırılır. İkinci aşamada ise, modele katkısı en az fonksiyonlar elenir.

Bu tezde biz, MARS'ın ikinci aşamasını oluşturan geriye doğru eleme yöntemini kullanmayı önermiyor, onun yerine penaltı yöntemini kullanmayı önermekteyiz. Bu sebeple, bir Tikhonov düzenleme problemi olarak MARS için cezalandırılmış hata kareler toplamı oluşturduk. Bu problemi ele alırken, geriye doğru eleme yöntemine bir alternatif ve önemli bir tamamlayıcı teknik olarak düşündüğümüz sürekli optimizasyon tekniklerini kullandık. Özellikle, iyi yapılandırılmış, doğrusal programlamaya benzeyen ve bundan dolayı da iç nokta yöntemlerini kullanmaya olanak sağlayan ikinci dereceden konik karesel programlamayı (CQP) kullandık.

Son olarak bu çalışmada biz, etkileşimli ve etkileşimsiz iki farklı veri kümesi için, ikinci dereceden konik karesel programlamayı, Tikhonov düzenleme problemi ile karşılaştırıyoruz. Ayrıca, başka iki veri kümeleri için, konik çok değişkenli uyarlanabilir regresyon eğrileri (CMARS) metodunu, sonuçları yayınlanacak olan tezden [49] gelen, diger iki sınıflandırma yöntemi olan Tikhonov düzenleme problemi ve sonsuz çekirdek öğrenimi (IKL) ile kıyaslıyoruz.

Anahtar Kelimeler: Genelleştirilmiş Parçalı Doğrusal Modeller, MARS, CMARS, Tikhonov Düzenleme, İkinci Dereceden Konik Karesel Programlama

To my family

ACKNOWLEDGMENTS

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

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

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

Also, I would like to thank to Assist. Prof. Dr. Süreyya Özoğur-Akyüz for her valuable contribution to this study and I am thankful to MSc. Gürkan Üstünkar for his support, patience and help.

In this thesis, I and Belgin Kayhan studied together and I am thankful her for all her helps, her friendship, understanding, encouragement and not letting me feel alone.

I especially deeply thank to my friend Ayşe Özmen for her friendship, motivating me, her valuable contribution and guidance throughout this study.

I would like thank to MSc. Fatma Yerlikaya for her help and valuable contribution to this study.

I am grateful to my manager Yeliz Bıyıkoğlu and my associate manager Sarp Kutlay for their support and understanding.

My studies wouldn't have been possible without the support of my family. I thank to them for their patience, believing and motivating me and endless love. They always encouraged me.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

The analysis of regression includes various methods for analyzing and modeling a finite number of variables, when the emphasis of interest is on the connection between one dependent variable and one or several independent variables. We may investigate how the typical value of the dependent one changes when any one of the independent variables becomes changed a bit, while the other independent variables are kept fixed ("ceteris paribus"). Regression is mostly employed for forecasting and prediction, where it strongly overlaps with the field of machine learning.

Various regression models are in wide use. The most famous is *Linear Regression Models (LRM). Generalized Linear Models (GLM)* mean an extension of process of the linear modeling which allows models to fit into data that obey probability distributions different from the Normal distribution, e.g., the Poisson, Binomial, Gamma. Furthermore, in classical linear models, GLM mean a relaxation of the requirement of the constant variance that is required for hypothesis testing [63]. The second widely applied statistical models are generalized linear models. They encompass traditional linear models with normal errors, probit and logistic models for binary data, loglinear models for multinomial data and many other models, e.g., the Binomial, Poisson, Normal and Gamma distribution. They can be formulated as generalized linear models through a suitable link function and response probability distribution.

As a linear technique GLM, GLM shares the usual shortcomings of linear modeling (LM) approach sometimes. At the beginning, both base on the assumption that the data follow a distribution of the exponential family. Moreover, they are affected from multi-collinearity, missing values and outliers in the data set. What is more, it is hard

to employ GLM for selecting significant predictors and their interactions. Finally, categorical predictors with a big numbers of categories may cause unreliable results for sparsity-related issues [51].

There exists a popular approach handling these problems effectively, called *Data Mining*. These techniques are usually fast, and they easily choose predictors and interactions. Further, they are minimally affected with outliers, missing values and collinearity, and they process high-level categorical predictors effectively [51]. Data mining approach is one of the most important techniques of scientific and technologic studies. It is an interdisciplinary complicated process, dealing with results of experiments, records, questionnaires and measurements, etc.. This process implies some difficulties, e.g., inaccurate predictions and computational time, interpretability and transfering results into different computational systems. Furthermore, complex data sets are another challenge in data mining. This motivates innovative data mining techniques.

An important data mining tool, *Multiple Adaptive Regression Spline* (*MARS*), is very beneficial, for high-dimensional problems. In fact, it does not impose any specific dependence between predictor and dependent variables. However, it estimates the contribution of basis functions so that the additive and interaction effects of the predictors as well can determine the dependent variable.

Employing MARS to enhance GLM speeds up the model-building process considerably and makes it more efficient [51]. In this thesis, we shall analyze *Generalized Partial Linear Models* (*GPLMs*), a particular semiparametric model of interest, which is an extension of GLM by a nonparametric component. There, in GPLM, a single nonparametric component joins the usual parametric terms. This means that, GPLM decomposes input variables into two sets and additively combines traditional linear models with nonlinear part of the model.

GPLMs are appreciated as a popular statistical modeling methodology because of its flexibility to many statistical problems and its availability by software to fit these models. The special form of GPLMs can be identified as semiparametric models, because the usual parametric terms are enhanced by a nonparametric component of some continuous covariate. A great advantages of semiparametric models is made up

of a *certain* grouping, e.g., linear and nonlinear or parametric and nonparametric that could be done for the features or input dimensions to select suitable submodels for them in a specific manner.

Our aim in this study is to combine GPLM with a different form of MARS. The algorithm of MARS has two steps, the forward and backward stepwise algorithms, in order to estimate the model function. In the forward step, the model is constructed by adding basis functions until a maximum level of complexity is achieved. In the backward stepwise algorithm, it removes the least significant basis functions from the model. In this thesis, we suggest to use *Penalized Residual Sum of Squares* ($PRSS$), instead of the backward algorithm in order to handle the complexity and the accuracy of the model. We built PRSS, changing the form of MARS into a *Tikhonov Regularization Problem*. In order to solve this problem, we use a continuous optimization technique called *Conic Quadratic Programming* (CQP), providing 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.

In this thesis, we give brief information about a literature review of regression models in Chapter 2. Moreover, this chapter includes an exhaustive information about Conic Quadratic Programming and the two other classification methods, that are *Infinite Kernel Learning* (IKL), a modern method of Machine Learning (support vector machine) and Tikhonov Regularization Problem which are detailly included in the thesis [49], which is on progress. Besides, after giving detailed explanation about MARS and its modified version CMARS algorithm, we mention the regularization of both linear and nonlinear parts theoretically in Chapter 3. This chapter also contains a numerical example of regularization for nonlinear part by using CMARS and comparison of the two methods, Tikhonov Regularization and CQP for with and without interaction data sets. Furthermore, in this chapter, we make a comparison by using some statistical performance measures between CMARS and two other classification methods that are IKL and Tikhonov Regularization whose results are obtained from the thesis [49], which is on progress, for two data sets. At Chapter 4, we conclude with an outlook to future studies.

CHAPTER 2

LITERATURE SURVEY AND BACKGROUND

2.1 Linear Regression Models

In modern statistics, *linear regression* is an approach to model the relationship between a scalar variable y, and one or more variables denoted by X and, as a vector, by $\boldsymbol{y}$. In linear regression, models of the unknown parameters are estimated from the data using linear functions so that such models are called "linear models". The general form of a Linear Regression Model (LRM) has the following form [68]:

$$\begin{aligned} y &= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \varepsilon \\ &= \boldsymbol{X}^T \boldsymbol{\beta} + \varepsilon, \end{aligned}$$

where y is the *response* variable, x_i $(i = 1, 2, \ldots, k)$ are the independent variables and ε is the unobserved random variable that adds noise to the linear relationship between variables. Error terms are assumed to be *white noise* that means they are normally distributed and mutually independent zero mean random variables, each with the same variance σ^2. The intercept term β_0, also referred as '*bias*' in some fields, and the regression coefficients β_i are the unknown parameters representing the degree of the relationship between independent and dependent variables. In fact, statistical estimation and inference in linear regression focuses on the vector $\boldsymbol{\beta}$.

Many methods have been developed for parameter estimation and inference in linear regression. However, *Least-Squares Estimation* (LSE) is the simplest and most popular one. The logic of the LSE method is to minimize the sum of squared residuals. In some situations, it is not practical to use LSE, instead, a more general form is attractive, known as *Maximum Likelihood Estimation* (MLE) [40]. Both techniques aims to get

the best line minimizing the sum of the squares of the vertical distances of the points from the hyperplane.

2.1.1 Least-Squares Estimation Technique

As a powerful but still simple prediction technique, the least-squares estimation can be considered as a method for fitting data. The model which is simple univariate linear with N observations has the following form:

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

Here, N is the number of the data with $E(\varepsilon_i) = 0$ and $Var(\varepsilon_i) = \sigma^2$. β_0 and β_1 are the unkown regression coefficients and they can be estimated by least-squares method. Here, the aim is to minimize the function of residual sum of the squares (RSS) between y and its expected value. This RSS function can be displayed as follows:

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

To minimize this function, the following system of equations should be solved

$$\frac{\partial RSS}{\partial \beta_0} = 0, \quad \frac{\partial RSS}{\partial \beta_1} = 0,$$
$$\sum_{i=1}^{N} y_i = n\hat{\beta}_0 + \hat{\beta}_1 \sum_{i=1}^{N} x_i, \quad \sum_{i=1}^{N} x_i y_i = \hat{\beta}_0 \sum_{i=1}^{N} x_i + \hat{\beta}_1 \sum_{i=1}^{N} x_i^2.$$

Then, the least-square (LS) estimates of β_0 and $beta_1$ can be found

$$\hat{\beta}_0 = \frac{\sum_{i=1}^{N} x_i^2 \sum_{i=1}^{N} y_i - \sum_{i=1}^{N} x_i \sum_{i=1}^{N} x_i y_i}{n \sum_{i=1}^{N} x_i^2 - (\sum_{i=1}^{N} x_i)^2}, \quad \hat{\beta}_1 = \frac{N \sum_{i=1}^{N} x_i y_i - \sum_{i=1}^{N} x_i \sum_{i=1}^{N} y_i}{N \sum_{i=1}^{N} x_i^2 - (\sum_{i=1}^{N} x_i)^2}.$$

As the LS estimators have minimum variance among all linear unbiased estimators, they are also known as *Best Linear Unbiased Estimators* ($BLUEs$) [5].

Furthermore, there can be more than one independent variable, let us say k variables, then, *Multiple Linear Regression* (MLR) model is employed. In this model, the data can be shown as in Table 2.1 [65]:

Table 2.1: Data for Multiple Linear Regression

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

It is possible to state the model as follows:

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

Here, we suppose that there is no correlation between errors and they are random variables having a zero mean and constant variance $Var(\varepsilon_i) = \sigma^2$. RSS in MLR can be displayed as:

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

Actually, as there are N equations with $k+1$ unknown regression parameters and a quadratic function of parameters, it is more practical to represent it in matrix notation [40]:

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

where N shows the number of observations in the data set, and $\boldsymbol{X}$ is the $N \times (k+1)$ independent variable matrix, $\boldsymbol{y}$ is the $N \times 1$ response vector; $\boldsymbol{\beta}$ is the $(k+1) \times 1$ regression coefficients vector including the intercept term and $\boldsymbol{\epsilon}$ is the $N \times 1$-vector of random errors. Hence, we can represent RSS in the following form [40]:

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

where $\|.\|_2$ denotes the Euclidean norm.

If we differentiate RSS with respect to $\boldsymbol{\beta}$, we obtain

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

By equating the first derivative of RSS to zero, we reach the *normal equations* $\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) = 0$ [40]. We can write it as follows:

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

When $\boldsymbol{X}^T\boldsymbol{X}$ is not singular, then, we can get the unique solution by using the following form:

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

where the fitted values can be represented by [40]

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

However, when the $\boldsymbol{X}^T\boldsymbol{X}$ is singular, then, we can use the *Singular Value Decomposition (SVD)* method to solve the normal equations [40].

2.1.2 Maximum Likelihood Estimation Technique

Least-squares estimation is a very useful technique, however, it does not make much sense in some situations. Then, MLE is an alternative estimation method as long as the distribution of the errors is known. Actually, MLE is a more common approach and shows better statistical properties than LSE [40] such as being more efficient. For instance, *Least-Square (LS)* estimators have the minimum variance among linear estimators only, however, *Maximum Likelihood (ML)* estimators have minimum variance among all other unbiased estimators.

Providing the selected probability distribution model, the likelihood of a set of data shows the probability of getting that particular set of data. The values of the unknown parameters which maximize the sample likelihood are called as the Maximum Likelihood Estimates or MLE's [75].

In LS method, there is no need for distributional assumptions, however, in MLE we have to know the distribution. Assuming that errors are random, uncorrelated and normally distributed with variances σ_i^2 $(i = 1, 2, \ldots, N)$, we can obtain the ML estimates of (2.1). The probability density function for y_i $(i = 1, 2, \ldots, N)$ can be shown as follows:

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

Here, $\boldsymbol{\sigma}$ is a diagonal matrix with diagonal entries σ_1, σ_2,..., σ_N, which are assumed to be equal to a constant term, σ. Since the likelihood function is the joint multiplications

of each density function y_i, the likelihood function of (2.3) has the following form:

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

and, if $\sigma_i = \sigma$ $(i = 1, 2, \ldots, N)$:

$$= (2\pi\sigma^2)^{-N/2} \exp\left[\sum_{i=1}^{N}\left(\frac{-1}{2\sigma^2}(y_i - (\boldsymbol{X}\boldsymbol{\beta})_i)^2\right), \right]$$

For mathematical convenience, it is better to take the logarithm of the function. Hence, it becomes:

$$\begin{aligned} \ln L &= -\frac{N}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - (\boldsymbol{X}\boldsymbol{\beta})_i)^2 \qquad (2.4) \\ &= -\frac{N}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) \\ &= -\frac{N}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\, RSS(\boldsymbol{\beta}). \end{aligned}$$

Here, $RSS(\boldsymbol{\beta})$ is same as in (2.2). Clearly, the former part of the equation consists of constant terms like N, π and σ so it can be ignored. Whereas in the latter part, RSS is not constant and to maximize the function, RSS should be minimized regarding β. Thus, it looks as the same least-square problem mentioned previously. That means the MLE method gives completely same estimates with LSE if the errors are random and distributed normally [5, 40, 75].

If variance is not constant and there is heteroscedasticity ($\sigma_i \neq \sigma_j$ for all $i \neq j$) among uncorrelated error terms which have a multivariate normal distribution with a known covariance matrix, then, we also should include standard deviation σ_i in equation (2.4). Our new minimization problem gets the following form:

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

Employing a diagonal weight matrix $\boldsymbol{W} := \mathrm{diag}(1/\sigma_1, 1/\sigma_2, \ldots, 1/\sigma_N)$, the system of equations becomes

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

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 not singular, then we reach the MLE of $\boldsymbol{\beta}$ for a weighted system by using the following equation:

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

These two methods, MLE and LSE, provide parameter estimators which have many good properties. However, both of them are sensitive to the outliers [75].

2.2 Nonlinear Regression Models

In world, the relationship between variables is not always linear. In some situations, the actual relationship can have a curvature model, instead of a straight line or a flat plane. Thus, there exist nonlinear regression models to fit these nonlinear relationships.

Nonlinear regression models can form any kind of relationship between dependent and independent variables. The response variables are nonlinear functions of model parameters along with one or more regressor variables. Actually, the general form of all regression models has the following notation:

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

Here, $\boldsymbol{\theta}$=$(\theta_1, \theta_2, \ldots, \theta_k)^T$ is a $(k \times 1)$-vector of unknown parameters, ϵ is an uncorrelated random error term with zero mean and variances σ_i^2 $(i = 1, 2, \ldots, N)$. $f(\boldsymbol{x},\boldsymbol{\theta})$ represents the expectation function for the nonlinear regression model and $\boldsymbol{x} = (x_1, x_2, \ldots, x_k)^T$ is an input vector of regressor variables [68]. Besides, the whole equation may be displayed in vector notation as follows:

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

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 $\boldsymbol{\epsilon}$ represents the residual vector. There are various techniques used for nonlinear regression modeling such as Nonlinear Regression methods, Maximum Likelihood Estimation method, the Levenberg-Marquardt Method and the Gauss-Newton method [106].

2.3 Generalized Linear Models

Generalized Linear Models (GLM) has a wide range of application fields such as classification and regression. It can search for linear and nonlinear relationships between a continuous, or binomial, multinomial categorical response variable and categorical

or continuous regressor variables in a flexible way. This method may be used even when the assumptions of normality and constant variance have failed [68].

Some widely used types of GLM can be regarded as special applications of generalized linear models, e.g., binomial and multinomial logit and prohibit regression models. In a GLM, the dependence of the mean value of a response variable to linear predictors is provided by a nonlinear link function allowing the response variable Y to be any member of an exponential family of distributions. The basic structure of a GLM is as follows:

$$\mu_i = h(\eta_i) = h(\boldsymbol{X}_i^T \boldsymbol{\beta}), \qquad \text{where} \quad \mu_i = E(Y_i), \text{ for } i = 1, 2, \ldots, N, \tag{2.5}$$

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

Moreover, a GLM generally makes the following assumptions; the response variable is independent and its distribution can be any of member of exponential density family. It has the following form [105]:

$$f_\theta(y) = \exp\left(\frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi)\right), \tag{2.6}$$

where b, a, c represents arbitrary functions, ϕ is an arbitrary *scale parameter* and θ is known as *natural* or *canonical parameter*.

There are many statistical models widely in GLMs. For instance: classical linear models with normal errors, logistic and prohibit models for binary data, log-linear models for multinomial data. There are also some other distributions such as Poisson, Binomial, Gamma and Normal Distributions, etc.. It is possible to represent them as a GLM by choosing a suitable link function and a response probability distribution. When the identity function is preferred as the link function and it is normally distributed, then ordinary linear models becomes a special case of GLMs.

In this part, let us give an example to illustrate the logic. The exponential form of

normal distribution can be displayed in the following way:

$$\begin{aligned} f(y) &:= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right) \\ &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{y^2+\mu^2-2y\mu}{2\sigma^2}\right) \\ &= \exp\left(-\frac{y^2}{2\sigma^2} - \frac{\mu^2}{2\sigma^2} + \frac{y\mu}{\sigma^2} - \log(\sqrt{2\pi\sigma^2})\right). \end{aligned}$$

It can turn into the form of an exponential family by replacing

$$\begin{aligned} &\mu = \theta \qquad \text{and} \qquad b(\theta) = \tfrac{\theta^2}{2}, \\ &\sigma = \phi \qquad \text{and} \qquad a(\phi) = \sigma^2, \\ &c(y,\phi) = \tfrac{-y^2}{4\phi^2} - \log(\sqrt{2\pi\phi^2}). \end{aligned}$$

Thus, our distribution takes the form in (2.6) and looks as follows:

$$f(y,\theta,\phi) = \exp\left(\frac{2y\theta-\theta^2}{2\phi^2} - \frac{y^2}{2\phi^2} - \log(\sqrt{2\pi\phi^2})\right). \tag{2.7}$$

2.4 Generalized Partial Linear Models

Generalized Partial Linear Model (*GPLM*) is an extension of the generalized linear models with a modification that there is a single nonparametric component. The model of GPLM is represented as follows [94]

$$E(Y|\boldsymbol{X},\boldsymbol{T}) = G(\boldsymbol{X}^T\boldsymbol{\beta} + \gamma(\boldsymbol{T})). \tag{2.8}$$

Here, $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_m)^T$ is a finite dimensional parameter and $\gamma(\cdot)$ is a smooth function that is estimated by *B-splines*. The term $\boldsymbol{X}$ denotes an m-variable random vector typically represents discrete covariables, while $\boldsymbol{T}$ means a q-variate random vector of continuous covariables that are modeled in a nonparametric way.

The log-likelihood function of L is represented by the composite form $L(\theta(\boldsymbol{\beta},\gamma))$ to emphasize the roles of predictors, parameters, and of the unknown curve. Thus, the straightforward maximization of the log-likelihood function is no longer suitable as an estimation technique. This cause overfitting when there no constraints on $\boldsymbol{\beta}$. In fact, it usually renders the parameters $\boldsymbol{\beta}$ unidentifiable. However, the maximization is possible via maximizing a penalized form of log-likelihood, as long as we make weak

constraints on $\boldsymbol{\gamma}$ by the smoothness assumption. Hence, the penalized log-likelihood is maximized [94]

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

Here, $H(\boldsymbol{\mu}) := \eta(\boldsymbol{X}, \boldsymbol{T}) = \boldsymbol{X}^T\boldsymbol{\beta} + \gamma(\boldsymbol{T})$ and $G := H^{-1}$ is a function linking the mean of the dependent variable to the regressors.

Besides, ℓ represents the log-likelihood of the linear predictor and the second term with integral is the part for penalization, and τ is a smoothing parameter. This parameter controls the balance between accuracy of the data fitting and its complexity (or smoothness) [15]. Smoothing provides us the guarantee that the estimation is robust enough regarding noise in data and any forms of perturbation [94].

2.4.1 B-Splines

Introduced by Isaac Jacob Schoenberg, B-spline is the short form of basis spline. Functions of B-spline have a minimal support about a given degree, domain partition and smoothness. Every spline function of a given degree, smoothness and domain partition can be represented as a linear combination of B-splines of that same degree and smoothness, and over that partition regarding a fundamental theorem [9].

B-splines composed of polynomial pieces where a special connection among pieces exists. In a B-spline, every control point is linked to a basis function. The curve is as follows [94]:

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

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

Here are 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 [94]:

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

for $k \geq 2$, its derivative is

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

Moreover, B-spline bases overlap with each other. For instance, first-degree B-spline bases overlap with two neighbors, second-degree B-spline bases with four-degree B-splines, and so far.

Some characteristics of B-splines are as follows [94]:

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

2.4.2 Methods of Estimation

Maximization of likelihood turns out to need an iterative least-squares approach; however, estimation and inference for GLMs base on the theory of maximum likelihood estimation. generalized partial linear model (GPLM) is a particular semiparametric model of interest which augments the generalized linear models in that the usual parametric terms are extended by a single nonparametric component. In general, the estimation methods for GPLM base on the approach 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 the present thesis, we shall focus on different types of estimation of $\boldsymbol{\gamma}(\cdot)$ and $\boldsymbol{\beta}$ based on B-splines.

2.4.2.1 Penalized Maximum Likelihood

We want to consider the GPLM model (2.8), where we assume that $G = H^{-1}$ is a link function. However, the model can be regarded as semiparametric GLM since all terms are linear except one; this means:

$$H(\boldsymbol{\mu}) = \eta(\boldsymbol{X}, \boldsymbol{T}) = \boldsymbol{X}^T\boldsymbol{\beta} + \gamma(\boldsymbol{T}) = \sum_{j=1}^{m} X_j\beta_j + \gamma(\boldsymbol{T}) \qquad (i = 1, 2, \ldots, N). \quad (2.9)$$

For the sake of simplicity, the observation values t_i of $\boldsymbol{T}$ in GPLM are thought to be one-dimensional. On that framework, $\mu_i = G(\eta_i)$ and

$$\eta_i = H(\mu_i) = \boldsymbol{X}_i^T\boldsymbol{\beta} + \gamma(t_i). \quad (2.10)$$

Let us apply penalized maximum likelihood estimation to prevent from overfitting. That technique is characterized by a score function $\partial\ell(\eta, y)/\partial\eta$. For our model, the penalized maximum criterion is given by [94]

$$j(\boldsymbol{\beta}, \gamma) = \ell(\eta, y) - \frac{1}{2}\tau \int_a^b (\gamma''(t))^2 dt. \quad (2.11)$$

Since we estimate the model by penalized maximum likelihood, we want to maximize (2.11); for this we desire to minimize the second part. We shall do this by employing B-splines through the local scoring algorithm. Therefore, we write a k degree B-spline with knots at the value t_i $(i = 1, 2, \ldots, N)$ instead of $\gamma(t)$. We will have $N-2$ interior points and $N + k - 1$ unknown parameters.

Thus, we arrive at a representation

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

where λ_j are coefficients, $\nu{=}N + k - 1$ and $B_{j,k} = B_j$ are B-spline basis functions. The vectorial form looks this way:

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

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

Defining 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 (2.11) can be stated by

$$j(\boldsymbol{\beta}, \boldsymbol{\gamma}) := l(\boldsymbol{\eta}, \boldsymbol{y}) - \frac{1}{2}\tau\boldsymbol{\lambda}^T\boldsymbol{K}\boldsymbol{\lambda}. \quad (2.12)$$

Assuming $N \geq \nu$ and that $\boldsymbol{B}$ is of full rank, let us insert the least-squares estimation $\boldsymbol{\lambda} = (\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T\boldsymbol{\gamma}(\boldsymbol{t})$ into (2.12) and we write $\boldsymbol{M}{:=}\boldsymbol{B}(\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{K}(\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T$. Then, we obtain

$$j(\boldsymbol{\beta},\boldsymbol{\gamma}) = l(\boldsymbol{\eta},\boldsymbol{y}) - \frac{1}{2}\tau\boldsymbol{\gamma}^T\boldsymbol{M}\boldsymbol{\gamma}. \tag{2.13}$$

To solve the minimization problem of (2.13) now, we have to find the optimal estimators $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\gamma}}$. We introduce $\boldsymbol{g}_1 {:=}\boldsymbol{X}\ \boldsymbol{\beta}$ and $\boldsymbol{g}_2 {:=}\boldsymbol{\gamma}(\boldsymbol{t})$; then (2.10) becomes

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

here $\boldsymbol{X}$ 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 subsequent system of equations needs be solved to maximize (2.11) over ($\boldsymbol{g_1}$ and $\boldsymbol{g_2}$):

$$\begin{aligned}\frac{\partial j(\boldsymbol{\beta},\boldsymbol{\gamma})}{\partial \boldsymbol{g_1}} &= (\frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{g_1}})^T\frac{\partial \ell(\boldsymbol{\eta},\boldsymbol{y})}{\partial \boldsymbol{\eta}} = 0, \\ \frac{\partial j(\boldsymbol{\beta},\boldsymbol{\gamma})}{\partial \boldsymbol{g_2}} &= (\frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{g_2}})^T\frac{\partial \ell(\boldsymbol{\eta},\boldsymbol{y})}{\partial \boldsymbol{\eta}} - \tau\boldsymbol{M}\boldsymbol{g}_2 = 0.\end{aligned} \tag{2.14}$$

The system equations are nonlinear in $\boldsymbol{\eta}$ and $\boldsymbol{g_2}$. For finding a solution, they are linearized around a current guess $\boldsymbol{\eta^0}$ and yield a Newton-Raphson type equation:

$$\frac{\partial \ell(\boldsymbol{\eta},\boldsymbol{y})}{\partial \boldsymbol{\eta}} \approx \frac{\partial \ell(\boldsymbol{\eta},\boldsymbol{y})}{\partial \boldsymbol{\eta}}\ |_{\boldsymbol{\eta}^0} + \frac{\partial^2 \ell(\boldsymbol{\eta},\boldsymbol{y})}{\partial \boldsymbol{\eta}\boldsymbol{\eta}^T}\ |_{\boldsymbol{\eta}^0}\ (\boldsymbol{\eta} - \boldsymbol{\eta^0}) = \boldsymbol{0}. \tag{2.15}$$

Using (2.15) in (2.14), setting $\boldsymbol{r}{:=}\partial \ell(\boldsymbol{\eta},\boldsymbol{y})/\partial \boldsymbol{\eta}$ and $\boldsymbol{C}{:=}-\partial^2 \ell(\boldsymbol{\eta},\boldsymbol{y})/\partial \boldsymbol{\eta}\boldsymbol{\eta}^T$, we come to the following matrix representation:

$$\begin{pmatrix} \boldsymbol{C} & \boldsymbol{C} \\ \boldsymbol{C} & \boldsymbol{C}+\tau\boldsymbol{M} \end{pmatrix} \begin{pmatrix} \boldsymbol{g_1}^1 - \boldsymbol{g_1}^0 \\ \boldsymbol{g_2}^1 - \boldsymbol{g_2}^0 \end{pmatrix} = \begin{pmatrix} \boldsymbol{r} \\ \boldsymbol{r} - \tau\boldsymbol{M}\boldsymbol{g}_2^0 \end{pmatrix}, \tag{2.16}$$

with $(\boldsymbol{g_1^0},\boldsymbol{g_2^0}) \rightarrow (\boldsymbol{g_1^1},\boldsymbol{g_2^1})$ being a Newton-Raphson step, $\boldsymbol{C}$ and $\boldsymbol{r}$ are calculated at $\boldsymbol{\eta^0}$. For a more simple form for (2.16), we set $\boldsymbol{h}{:=}\boldsymbol{\eta^0}{+}\boldsymbol{C^{-1}r}$ and $\boldsymbol{S}_B{:=}(\boldsymbol{C}{+}\tau\boldsymbol{M})^{-1}\boldsymbol{C}$, which is a weighted B-spline operator. Herewith, (2.16) becomes

$$\begin{pmatrix} \boldsymbol{C} & \boldsymbol{C} \\ \boldsymbol{S}_B & \boldsymbol{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{g_1^1} \\ \boldsymbol{g_2^1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{C} \\ \boldsymbol{S}_B \end{pmatrix} \boldsymbol{h}. \tag{2.17}$$

Multiplying the upper row with $\boldsymbol{C^{-1}}$ and the second row with $(\boldsymbol{C}+\tau\boldsymbol{M})^{-1}$, we may transform it into the form

$$\begin{pmatrix} \boldsymbol{g_1^1} \\ \boldsymbol{g_2^1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{X}\boldsymbol{\beta^1} \\ \boldsymbol{\gamma^1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{h} - \boldsymbol{g_2^1} \\ \boldsymbol{S}_B(\boldsymbol{h} - \boldsymbol{g_1^1}) \end{pmatrix}. \tag{2.18}$$

Then, $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\gamma}}$ can be explicitly found without any iteration (inner loop backfitting), and

$$\begin{aligned} \hat{\boldsymbol{g_1}} &= \boldsymbol{X}\hat{\boldsymbol{\beta}} = \boldsymbol{X}\boldsymbol{X}^T\boldsymbol{C}(\boldsymbol{I}-\boldsymbol{S}_B)\boldsymbol{X}^{-1}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{C}(\boldsymbol{I}-\boldsymbol{S}_B)\boldsymbol{h}, \\ \hat{\boldsymbol{g_2}} &= \hat{\boldsymbol{\gamma}} = \boldsymbol{S}_B(\boldsymbol{h}-\boldsymbol{X}\hat{\boldsymbol{\beta}}). \end{aligned} \tag{2.19}$$

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

Newton-Raphson updates serve to solve a weighted, penalized quadratic criterion. That criterion local by approximates the penalized log-likelihood. From the updated $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\gamma}})$, the outer loop has to be iterated to redefine $\boldsymbol{\eta}$ and, by this, $\boldsymbol{h}$ and $\boldsymbol{C}$. Then, the loop is repeated until convergence is regarded sufficient [28]. Since the outer loop is just some Newton-Raphson step, some step size optimization is conducted, and the outer loop will converge. Now, we consider a trial value, of the form

$$\boldsymbol{\eta}^{\phi} := \boldsymbol{\phi}\boldsymbol{\eta}^1 + (1-\boldsymbol{\phi})\boldsymbol{\eta}^0, \tag{2.20}$$

with $\boldsymbol{g_s}$ $(s=1,2)$ defined. Therefore, (2.20) becomes a Newton-Raphson step of size ϕ; we maximize $j(\boldsymbol{\eta}^{(\phi)})$ with respect to ϕ [94]. Convergence is ensured by the standard results on the Newton-Raphson procedure [76].

For asymptotic properties of these models we refer to [28, 39]. Considering the equations (2.19), we get

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

Here $\{\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{C}(\boldsymbol{I}\text{-}\boldsymbol{S}_B)\boldsymbol{X}\}^{-1}\boldsymbol{X}^T\boldsymbol{C}(\boldsymbol{I}\text{-}\boldsymbol{S}_B)\boldsymbol{B}\boldsymbol{\lambda}$ can be seen as the estimated correction term.

Furthermore, considering equations (2.18)-(2.19), the functions $\boldsymbol{g_1}$ and $\boldsymbol{g_2}$ are estimated by linear mapping, or smoother, applied to the adjusted dependent variable $\boldsymbol{h}$, with weight $\boldsymbol{C}$ given by the *information matrix*. With $\boldsymbol{R}_B$ being the weighted additive fit operator, by convergence we get

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

with $\hat{\boldsymbol{r}}=\partial\ell(\boldsymbol{\eta},\boldsymbol{y})/\partial\boldsymbol{\eta}|_{\hat{\boldsymbol{\eta}}}$ [94]. Changing from $\boldsymbol{h}$, $\boldsymbol{R}_B$ and $\boldsymbol{C}$ to their asymptotic versions $\boldsymbol{h_0}$, $\boldsymbol{R}_{B_0}$ and $\boldsymbol{C_0}$, with $\boldsymbol{h}\approx\boldsymbol{h_0}$ having 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

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

In fact, $\boldsymbol{R}_{B_j}$ is the matrix producing $\hat{g}_j$ from $\boldsymbol{h}$ based on B-splines. We note that $\hat{\boldsymbol{\eta}}$ is asymptotically distributed as $N(\boldsymbol{\eta}_0,\ \boldsymbol{R}_{B_0}\boldsymbol{C}_0^{-1}\ \boldsymbol{R}_{B_0}^T\phi)$ [39].

2.4.2.2 Least-Squares: Penalized Iteratively Re-Weighted:

By the *penalized iteratively reweighted least-squares* ($P-IRLS$) method, the penalized likelihood is maximized. 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, $G(\eta_i^{[p]})$ being the inverse function of the link at the pth iteration, we can represent (2.17) as the linear system that leads to $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$. Eventually, we minimize the following term to find the $(p+1)$th estimate of the linear predictor $\boldsymbol{\eta}^{[p+1]}$:

$$\|\boldsymbol{C}^{[p]}(\boldsymbol{h}^{[p]}-\boldsymbol{\eta})\|_2+\tau\boldsymbol{\gamma}^T\boldsymbol{M}\boldsymbol{\gamma}. \tag{2.21}$$

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

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

with H' being the first derivative of H with respect to $\boldsymbol{\beta}$, and $\boldsymbol{C}^{[p]}$ being a diagonal weight matrix with elements $C_{ii}^{[p]} := 1/V(\mu_i^{[p]})H'(\mu_i^{[p]})^2$. Here, $V(\mu_i^{[p]})$ is proportional to the variance of Y_i according to the current estimate $\mu_i^{[p]}$. Using $\boldsymbol{\gamma}(\boldsymbol{t})=\boldsymbol{B}\boldsymbol{\lambda}$ in the function (2.21), it looks in the following way:

$$\|\boldsymbol{C}^{[p]}(\boldsymbol{h}^{[p]}-\boldsymbol{X}\boldsymbol{\beta}-\boldsymbol{B}\boldsymbol{\lambda}\|_2+\tau\boldsymbol{\lambda}^T\boldsymbol{K}\boldsymbol{\lambda}. \tag{2.22}$$

Let us assume $\boldsymbol{K}$ to be of rank $z<v$ [28]. We can write $\boldsymbol{J}^T\boldsymbol{K}\boldsymbol{J}=\boldsymbol{I}$, $\boldsymbol{T}^T\boldsymbol{K}\boldsymbol{T}=\boldsymbol{0}$ and $\boldsymbol{J}^T\boldsymbol{T}=\boldsymbol{0}$, $\boldsymbol{J}$ and $\boldsymbol{T}$ being two matrices with ν rows and with full column ranks z and ν-z, respectively. Representing

$$\boldsymbol{\lambda}=\boldsymbol{T}\boldsymbol{\delta}+\boldsymbol{J}\boldsymbol{\varepsilon}, \tag{2.23}$$

with vectors $\boldsymbol{\delta}$ and $\boldsymbol{\varepsilon}$ being vectors of dimensions z and ν-z, respectively. The objective term (2.21) now takes the form

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

Let us split its minimization through separating to solution with respect to $\boldsymbol{\beta}$ and $\boldsymbol{\delta}$ from the one on $\boldsymbol{\varepsilon}$, with an application of *Householder decomposition* [21]. Herewith, we may write

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

$\boldsymbol{Q}$=[$\boldsymbol{Q}_1$,$\boldsymbol{Q}_2$] being orthogonal and $\boldsymbol{R}$ being upper triangular and of full rank $m+\nu-z$. By this, our problem becomes as minimization of

$$\|\boldsymbol{Q}_1^T \boldsymbol{C}^k \boldsymbol{h}^k - \boldsymbol{R} \begin{pmatrix} \boldsymbol{\beta} \\ \boldsymbol{\delta} \end{pmatrix} - \boldsymbol{Q}_1^T \boldsymbol{C}^k \boldsymbol{BJ\varepsilon}\|_2 \tag{2.24}$$

with respect to ($\boldsymbol{\beta}$,$\boldsymbol{\delta}$), provided $\boldsymbol{\varepsilon}$ based on a minimization of

$$\|\boldsymbol{Q}_2^T \boldsymbol{C}^k \boldsymbol{h}^k - \boldsymbol{Q}_2^T \boldsymbol{C}^k \boldsymbol{BJ\varepsilon}\|_2 + \tau \boldsymbol{\varepsilon}^T \boldsymbol{\varepsilon}. \tag{2.25}$$

With an appropriate selection of $\boldsymbol{\beta}$ and $\boldsymbol{\delta}$, given $\boldsymbol{\varepsilon}$, the term (2.24) can be set to zero. When we take $\boldsymbol{H}$:=$\boldsymbol{Q}_2^T \boldsymbol{C}^k \boldsymbol{h}^k$ and $\boldsymbol{V}$:=$\boldsymbol{Q}_2^T \boldsymbol{C}^k \boldsymbol{BJ}$, (2.25) becomes the following problem of minimization

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

which is a kind of a *Tikhonov regularization problem* [3]. Its solution is

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

Now, we can find the other parameters via

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

Then, our vector $\widetilde{\boldsymbol{\lambda}}$ can be computed by (2.23) and, hence, $\boldsymbol{\eta}^{[p+1]}$=$\boldsymbol{X}\widetilde{\boldsymbol{\beta}}$+$\boldsymbol{B}\widetilde{\boldsymbol{\lambda}}$ may be computed. Our matrices $\boldsymbol{J}$ and $\boldsymbol{T}$ can be calculated by a *Cholesky and Householder transformation* [21].

2.4.2.3 The Choice for Penalty Parameters: An Alternative

Both the penalized maximum likelihood method and also the P-IRLS methods include the smoothing parameter τ. For estimating this parameter, there exist two widely applied methods: *Generalized Cross Validation* (GCV) and minimization of an *UnBiased Risk Estimator* ($UBRE$) [15]. But, here, we state an alternative method, which is known as conic quadratic programming [94].

Turning back to equation (2.22) and employing a *Cholesky decomposition*, with $\boldsymbol{K}$ being a $(\nu\times\nu)$-matrix $\boldsymbol{K}$ such that $\boldsymbol{K}$=$\boldsymbol{U}^T\boldsymbol{U}$, then, the objective term is

$$\|\boldsymbol{W}\boldsymbol{\varphi}-\boldsymbol{v}\|_2+\tau\|\boldsymbol{U}\boldsymbol{\lambda}\|_2^2. \tag{2.26}$$

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

By this, our problem (2.26) turns into an optimization constrained problem:

$$\text{minimize}\;\; G(\boldsymbol{\varphi})\;\; \text{subject to}\quad g(\boldsymbol{\lambda})\leq 0, \tag{2.27}$$

with $G(\boldsymbol{\varphi}) := \|\boldsymbol{W}\boldsymbol{\varphi}-\boldsymbol{v}\|_2$ and g($\boldsymbol{\lambda}$):= $\|\boldsymbol{U}\boldsymbol{\lambda}\|_2 - M$, and $M \geq 0$ which is preselected with some tolerance before or adapted within of a process of learning. Now, optimization problem (2.27) can be written in the following equivalent form:

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

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:

$$\begin{aligned} &\text{minimize} \quad t,\\ \text{subject to} \quad &\|\boldsymbol{W}\boldsymbol{\varphi}-\boldsymbol{v}\|_2\leq t,\\ &\|\boldsymbol{U}\boldsymbol{\lambda}\|_2\leq \sqrt{M}. \end{aligned} \tag{2.28}$$

Applying continuous optimization methods, by the general conic quadratic optimization programming [70]

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

it can be understood that our minimization problem is such a conic quadratic programming problem, where

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

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

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

Problem (2.28) is studied and evaluated for stating the *dual* to this problem soon, and our principal looks in the following way now:

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

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

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

The *dual problem* to the latter problem is defined by

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

Traditional polynomial time algorithms may be applied to solve convex optimization problems such as semi-definite programming, geometric programming and, in particular, Conic Quadratic Problems. But, these algorithms employ local information on the objective function only and have constraints. For solving "*well-structured*" convex problems such as conic quadratic problems, *Interior Point Methods* [72, 83],

introduced firstly by Karmarkar in 1984, are preferred. Those methods, also called Barrier Methods, are based on both the primal *and* the dual problem. They admit better complexity bounds and allows better practical performance. Furthermore, they guarantee feasibility throughout the entire iteration procedures. In contrast, Penalty Methods and Tikhonov Regularization can be considered as *Exterior Point Methods* with possible infeasibility [94].

By now, it has been explained that a spline regression problem can be stated either as a Tikhonov regularization problem or as a conic quadratic problem. The following chapters will closely introduce both Tikhonov regularization and Conic quadratic problems that we connect with *multiple adaptive regression splines* ($MARS$) for the nonlinear arbitrary function $\gamma(t)$. This approach is called *adaptive* because the selection of basis functions is data-based and specific to the problem at hand. Via this combination, *conic multivariate adaptive regression splines* ($CMARS$) will be introduced.

2.4.3 On Motivations and Various Applications

A great advantage in Generalized Partial Linear Models (GPLMs) consists in a certain *grouping* which could be done for the input dimensions or features to assign appropriate submodels specifically [94]. We know linear and nonlinear ones, as well as parametrical and nonparametrical ones. We may use Inverse Problem techniques, e.g., Tikhonov regularization [3], to separate linear models from nonlinear or nonparametrical ones, for the linear submodels separately, within the entire GPLMs. Such a particular representation of submodels provides a better accuracy and a better stability (regularity) under noise in the data.

We state the following real-world motivations, all of them related with important modern applications; they all can lead to GPLMs.

(i) 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 is still too little understood yet [94].

(ii) Staying in the field of financial markets and representing different processes by

stochastic differential equations (possibly discretized) and Lévy processes, the deterministic drift term could be stated by a linear submodel, while stochastic diffusion term (possibly simulated) and the compound Poisson processes on jump behavior might be expresed by a nonlinear model [94].

(iii) A linear submodel may easily represent given (open) information, but a nonlinear submodel could encompass hidden information such as, e.g., Hidden Markov Models. This model distinction of "non-hidden" versus "hidden" can be applied in speech or image processing, in the financial sector of loan banking and credit risk, etc. but also in physics [94].

Grouping of input dimensions or features mentioned above is preformed in reality by *data mining*, especially, by *clustering* and by *classification* [102]. Let us give three areas of examples. Actually, (α), Taylan, Weber and Beck (2007) [91] clustered time points of the change of prices at some stock exchange. (β), Weber et al. (2007) [94] regressed credit default to the individual features of the credit takers (firms or countries). (γ), in the modeling and estimation work of Kropat, Weber and Pedamallu (2009) [55] on regulatory networks, a distinction is presented between *target* variables (e.g., from nature, medicine or emissions) and *environmental* variables (e.g., of toxic substances or from finance). Within both categories, items (variables, dimensions of features, or actors) are clustered according to whether they are regarded to be stochastically dependent or correlated with each other. That is practically realized by clustering via the geometrical positions of all the given data points and, as valuations, ellipsoids are raised over the clusters to reflect these mutual relationships. We emphasize that this approach also led to the introduction of ellipsoid collaborative games by Alparslan Gök and Weber (2009) [1, 100, 101].

2.5 Tikhonov Regularization

Problems that have an exist and unique solution depending continuously on the data are known as *well-posed*. However, the ones that are not well-posed are called as *ill-posed*. There are some methods to turn these ill-conditioned problems into well-posed. *Tikhonov regularization*, also known as *ridge regression*, is one of the most commonly used method [52].

The standard approach to solve an over determined system of linear equations given as

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

It is known as linear least-squares and seeks to minimize the residual

$$\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2.$$

Singular value decomposition (SVD) of the coefficient matrix $\boldsymbol{X}$ of a regarded linear systems of equations can simply express the Tikhonov solution [3].

Methods for determining a suitable regularization parameter can be divided into two main classes. The first one consists of the methods based on knowledge, or a good estimate, of error norm and the second one includes the methods that do not require any knowledge about error norm. The *discrepancy principle* is an example of the first class while the *Cross-Validation* and *L-curve* are examples of the second class.

To regularize the solution of a discrete ill-posed problem the discrepancy principle can be used by assuming that a reasonable level for $\delta = \|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2$ is known [3]. 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 or the norm of the error in the data is known. All solutions with $\|\boldsymbol{X}\boldsymbol{\beta}\text{-}\boldsymbol{y}\|_2^2 \leq \delta$ are considered under the discrepancy principle, and the one minimizing the norm of β is preferable. Since the norm (length) $\|\boldsymbol{\beta}\|_2$ of $\boldsymbol{\beta}$ represents the complexity of the possible solution, it is usually preferred to obtain a solution minimizing the norm of first- or second-order derivative of $\boldsymbol{\beta}$ [3],

$$\min_{\boldsymbol{\beta}} \quad \|\boldsymbol{\beta}\|_2 \quad \text{such that} \quad \|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 \leq \delta.$$

As δ increases, the set of feasible models expands, and the minimum value of $\|\beta\|_2$ decreases. We can show this minimization problem by considering problems of the form

$$\min_{\boldsymbol{\beta}} \quad \|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 \quad \text{such that} \quad \|\boldsymbol{\beta}\|_2 \leq \epsilon. \tag{2.29}$$

As ϵ decreases, the set of all feasible solutions becomes smaller, and the minimum value of increases $\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2$ increases.

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 [35].

By applying Lagrange multipliers to the problem (2.29), we obtain:

$$\min_{\boldsymbol{\beta}} \quad \|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 + \varphi^2 \|\boldsymbol{\beta}\|_2^2, \tag{2.30}$$

where the Lagrange multiplier $\lambda = \varphi^2$ is the tradeoff parameter between the two parts. Since an appropriate regularization parameter should properly balance the two parts, *L-curve* is used to control the tradeoff. When plotting 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, the resulting curve often has a characteristic L shape [35]. The transition between under- and over-regularizations regions is taking place the 'corner' of the L-curve, and the value of λ at this corner corresponds to the optimal value of the regularization parameter [34].

Different kinds of Tikhonov regularization represented by minimization problems are mentioned. For some appropriate choice of the values δ, ϵ and φ, these problems have the same solution. These problems can be solved using SVD [3]. However, in many situations, a solution minimizing the norm of first- or second-order derivative of $\boldsymbol{\beta}$ is preferred. First- or second-order difference quotients of $\boldsymbol{\beta}$, regarded as a function evaluated at the 'points' j and $j+1$, give these derivatives. First- and second-order derivatives are approximated by using these difference quotients; all of them are comprised of products $\boldsymbol{L}\boldsymbol{\beta}$ of $\boldsymbol{\beta}$. Here, $\boldsymbol{L}$ is a matrix representing the discrete differential operators of first and second order respectively, and the optimization problem takes the following form:

$$\min_{\boldsymbol{\beta}} \ \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \varphi^2 \, \|\boldsymbol{L}\boldsymbol{\beta}\|_2^2 \, . \tag{2.31}$$

The optimization problem in (2.30), can be considered as a special case of (2.31) where the unit matrix ($\boldsymbol{L} = \mathbf{I}$) is used. As mentioned, this zeroth-order Tikhonov Regularization is solved by using SVD. For higher-order Tikhonov Regularization problems *generalized singular value decomposition* ($GSVD$) is used. (For more information, please refer to [49].)

2.6 Conic Quadratic Programming

Convex programming deals with problems, which arise frequently in many different application fields, consisting of minimizing a convex function over a convex set. These

programs are not only computationally tractable but they also have theoretically efficient solution methods. Convex programming consists of several important specially structured classes of problems like: semidefinite programming, second order cone programming, and geometric programming. These methods are very effective methods for *linear*, *conic quadratic* and *semidefinite programming*, all are examples of conic problems [91].

Several "*generic*" families of conic problems are of special interest, both from the viewpoint of theory and applications. The cones underlying these problems are simple enough, so that one can describe explicitly the dual cone; as a result, the general duality machinery we have developed becomes "*algorithmic*", as in the Linear Programming case. Moreover, in many cases this "*algorithmic duality machinery*" allows to understand more deeply the original model, to convert it into equivalent forms better suited for numerical processing, etc.. The relative simplicity of the underlying cones also enables one to develop efficient computational methods for the corresponding conic problems. The most famous example of a "*nice*" generic conic problem is, doubtless, Linear Programming (LP); however, it is not the only problem of this sort. Two other nice generic conic problems of extreme importance are Conic Quadratic and Semidefinite programs [71]. In this part, we will consider the former one, CQP.

We consider a conic quadratic (and, in particular, a quadratically constrained) optimization problem with uncertain data, known only to reside in some uncertainty set U. The robust counterpart of such a problem leads usually to an NP-hard semidefinite problem; this is the case for example when U is given as intersection of ellipsoids, or as an n-dimensional box. For these cases we build a single, explicit semidefinite program, which approximates the NP-hard robust counterpart, and we derive an estimate on the quality of the approximation, which is independent of the dimensions of the underlying conic quadratic problem [7].

A *generic conic problem* can be written as follows [71]:

$$\min_{\boldsymbol{x}} \quad \boldsymbol{c}^{\boldsymbol{T}}\boldsymbol{x}, \quad \text{where} \quad \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \in K, \tag{2.32}$$

where K is a cone of direct product of m cones, each of them being either semidefinite or second-order cones:

$$K := S_+^{k_1} \times \ldots \times S_+^{k_p} \times L^{k_{p-1}} \times \ldots \times L^{k_m} \subseteq E := S_+^{k_1} \times \ldots \times S_+^{k_p} \times \mathbb{R}^{k_{p-1}} \times \ldots \times \mathbb{R}^{k_m}.$$

A *conic quadratic problem* is a conic problem [71]. Geometrically, a conic problem is to minimize a linear functional over the intersection of affine plane and cone [71].

$$\min_{\boldsymbol{x}} \quad \boldsymbol{c}^{\boldsymbol{T}}\boldsymbol{x} \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \geq_K \boldsymbol{0}$$

for which the cone K is a direct product of several ice-cream cones:

$$K := L^{k_1} \times \ldots \times L^{k_m} \subseteq E, \tag{2.33}$$

and the k-dimensional ice-cream (second-order, Lorentz) cone L^k is given as follows:

$$L^k := \left\{ x = (x_1, x_2, \ldots, x_k)^T \in \mathbb{R}^k \mid x_k \geq \sqrt{x_1^2 + x_2^2 + \ldots + x_{k-1}^2} \right\} \quad (k \geq 2).$$

In general, a conic quadratic problem can be defined as an optimization problem with linear objective and finitely many "*ice-cream constraints*"

$$\boldsymbol{A}_i\boldsymbol{x} - \boldsymbol{b}_i \geq_{L^{k_i}} \boldsymbol{0} \quad (i = 1, 2, ..., N).$$

Thus, a *CQ* problem can be written as [71]

$$\min_{\boldsymbol{x}} \quad \boldsymbol{c}^{\boldsymbol{T}}\boldsymbol{x} \quad \text{subject to} \quad \boldsymbol{A}_i\boldsymbol{x} - \boldsymbol{b}_i \geq_{L^{k_i}} \boldsymbol{0} \quad (i = 1, 2, ..., N).$$

By partitioning the data matrix $[\boldsymbol{A}_i, \boldsymbol{b}_i]$ given by

$$[\boldsymbol{A}_i, \boldsymbol{b}_i] = \begin{bmatrix} \boldsymbol{D}_i & \boldsymbol{d}_i \\ \boldsymbol{p}_i^T & q_i \end{bmatrix},$$

where $\boldsymbol{D}_i$ is of the size $(k_i - 1)\times$ (dim $\boldsymbol{x}$), we can write the problem as follows:

$$\min_{\boldsymbol{x}} \quad \boldsymbol{c}^{\boldsymbol{T}}\boldsymbol{x}, \quad \text{subject to} \quad \|\boldsymbol{D}_i\boldsymbol{x} - \boldsymbol{d}_i\|_2 \leq \boldsymbol{p}_i^T\boldsymbol{x} - q_i \quad (i = 1, 2, ..., m). \tag{2.34}$$

This is the most explicit form that is preferred to use. In this form, $\boldsymbol{D}_i$ are matrices of the same row dimensions as $\boldsymbol{x}$, $\boldsymbol{d}_i$ are vectors of the same dimensions as the column dimensions of the matrices $\boldsymbol{D}_i$, $\boldsymbol{p}_i$ are vectors of the same dimensions as $\boldsymbol{x}$ and q_i are real numbers [71].

2.6.1 Solution Methods for Conic Quadratic Programming

In order to solve convex optimization problems like LP, semidefinite programming, geometric programming and also, conic quadratic problems, *classical polynomial time*

algorithms can be applied. Because of using local information on the objective function and constraints, these algorithms have some disadvantages. Thus, to solve "well-structured" convex problems such as CQ problems, there are *Interior Point* algorithms (IPMs) [72]. If an optimization problem is given by

$$\min_{\boldsymbol{x}} \quad \boldsymbol{c}^{\boldsymbol{T}}\boldsymbol{x}, \quad \text{where} \quad \boldsymbol{x} \in \Omega \subseteq \mathbb{R}^n,$$

Ω is generally assumed to be closed and convex, and IPMs generally base on the interior points of this feasible set. Then, an *interior penalty function* (*barrier*) $F(\mathbf{x})$ is considered, well defined (smooth and strongly convex) in the interior of $\boldsymbol{\Omega}$ and "blowing up" as a sequences from the interior int $\boldsymbol{\Omega}$ approaches a boundary point of $\boldsymbol{\Omega}$ [106]:

$$\boldsymbol{x}_k \in \text{int } \boldsymbol{\Omega} \ (k \in \mathbb{N}_0), \quad \lim_{k\to\infty} \boldsymbol{x}_k \in \partial\boldsymbol{\Omega} \ \Rightarrow \ F(\boldsymbol{x}_k) \to \infty \quad (k \to \infty).$$

Now, we arrive at a parametric family of functions, $F_t(\boldsymbol{x})$, generated by our objective interior *penalty function* $F_t(\boldsymbol{x}) := t\boldsymbol{c}^T\boldsymbol{x} + F(\boldsymbol{x}) : \text{int } \boldsymbol{\Omega} \to \mathbb{R}$. Here, we assume that the *penalty parameter* t is nonnegative. Under mild regularity assumptions,

• every function $F_t(\cdot)$ gets its minimum over the interior of $\boldsymbol{\Omega}$, the minimizers $\boldsymbol{x}_*(t)$ being unique,

• the central path $\boldsymbol{x}_*(t)$ is a smooth curve, and all its limiting points (as $t \to \infty$) belong to the set of optimal solution of above optimization problem [106].

The advantages of these algorithms can be stated as follows; they can employ the structure of the problem, allow better complexity bounds for the indicated generic problems and exhibit a much better practical performance [106].

2.6.2 Complexity of Conic Quadratic Programming

Let us consider the following conic quadratic optimization program:

$$\min_{x} \quad \boldsymbol{c}^{\boldsymbol{T}}\boldsymbol{x}, \quad \text{subject to} \quad \|\boldsymbol{D}_i\boldsymbol{x} - \boldsymbol{d}_i\|_2 \le \boldsymbol{p}_i^T\boldsymbol{x} - q_i \quad (i = 1, 2, \ldots, k), \quad \|\boldsymbol{x}\|_2 \le t,$$

where the matrices D_i are of the type $n_i \times n$, p_i, $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{d}_i \in \mathbb{R}_i^n$. Let further use the data above be represented in the way of [71]

$$\text{Data}(2.34) := [k; n; n_1; \ldots; \boldsymbol{c}; \boldsymbol{D}_1, \boldsymbol{d}_1, \boldsymbol{p}_1, q_1; \ldots, \boldsymbol{D}_k, \boldsymbol{d}_k, \boldsymbol{p}_k, q_k; t]$$

and

$$\text{Size}(2.34) := \dim \text{Data}(2.34) := \left(k + \sum_{i=1}^{k} n_i \right) (n+1) + k + n + 3.$$

The arithmetic complexity of ε-solution is given by

$$\text{Compl}(2.34, \varepsilon) := O(1)(k+1)^{1/2} n \left(n^2 + k + \sum_{i=1}^{k} {n_i}^2 \right) \text{Digits}(2.34, \varepsilon),$$

where

$$\text{Digits}(2.34, \varepsilon) := \ln((\text{Size}(2.34)) + \|\text{Data}(2.34)\|_1 \varepsilon^2)/\varepsilon$$

is defined as the number of accuracy digits in an ε-solution to (2.34), referring to the sum (or l_1) norm [91].

2.6.3 MOSEK

The MOSEK as a MATLAB add-on toolbox is an optimization tool for solving large-scale mathematical optimization problems. More information can be found in the website of MOSEK, http://www.mosek.com.

By using MOSEK optimization toolbox, it is possible to solve the following large-scale optimization problems:

- *Linear problems*,
- *Conic quadratic problems*,
- *Quadratic and quadratically constrained problems*,

- *General convex nonlinear problems*,
- *Mixed integer linear, conic and quadratic problems*.

Besides, it can be used to solve (constrained) linear least-squares and, one and infinity norm estimation problems. Each of these optimization problems are solved by one of the following optimizers in MOSEK:

- *Interior-point optimizer*,
- *Conic interior-point optimizer*,
- *Primal simplex optimizer*,
- *Mixed integer optimizer*.

As there are different optimizers, they can produce different types of solutions. For example, the interior-point optimizers produces a general interior-point solution while the simplex optimizer produces a basic solution.

MOSEK has some technical highlights [95]. Problem size is only limited by the available memory. It has an interior-point optimizer with basis identification. Besides, there are both primal and dual simplex optimizers for linear programming. It has a very efficient presolver for reducing problem size before optimization. Moreover, it has a capability to solve one problem with different optimizers simultaneously and to read and write industry standard formats such as MPS, LP and XML.

In addition to Matlab Toolbox, there are some other programming languages that MOSEK is compatible such as C/C++, Java, NET and Python [95].

2.7 Infinite Kernel Learning

Data classification is a popular subject in *machine learning*, which is a scientific discipline that is concerned with the design and development of algorithms that allow computers to evolve behaviors based on empirical data, such as from sensor data or databases. One of the characteristics of machine learning is to automatically learn

to find complex patterns and arrive intelligent decisions regarding data. Here, the problem arises in that the set of all possible behaviors given all feasible inputs is too complex to express. *Support vector machines*, Bayes point machines, Gaussian processes, and Kernel principal component analysis are some Kernel-based methods which represent a major development in machine learning algorithms [47].

2.7.1 Support Vector Machines

SVMs are a set of related supervised learning techniques employed for regression and classification. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that predicts whether a new example falls into one category or the other. A SVM produce a hyperplane or set of hyperplanes in a high or infinite dimensional space, that may be applied in classification, regression or other tasks. Generally, the larger the margin, the lower the generalization error of the classifier. Thus, a good separation is provided by the hyperplane which has the largest distance to the nearest training data points of any class (the so-called *functional margin*) [45].

2.7.2 Kernel Learning

Nonlinear data can also be classified by *Machine learning* algorithms. Multiple kernel methods are helpful, particularly, when the data is heterogeneous and large-scale. The main idea of *multiple kernel learning* is to combine finitely many pre-chosen kernels in a convex combination [85]

$$k_\beta(\boldsymbol{x}_i, \boldsymbol{x}_j) := \sum_{\kappa=1}^{K} \beta_\kappa k_\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) \ , \ \text{where } i, j = 1, 2, \ldots, N. \tag{2.35}$$

In this study, an integral refine the sum in (2.35). In [4], semi-definite programming models a multiple kernel reformulation to choose the optimum weights of corresponding kernels. However, this is not good regarding computation time because of semi-definite programming. This reformulation is advanced in [85] by *semi-infinite linear*

programming with the optimization model:

$$\begin{aligned} &\max_{\theta,\boldsymbol{\beta}} \quad \theta \quad (\theta \in \mathbb{R}, \boldsymbol{\beta} \in \mathbb{R}^K) \\ &\text{such that} \quad \boldsymbol{\beta} \geqslant \boldsymbol{0},\ \sum_{\kappa=1}^{K} \beta_\kappa = 1, \\ &\sum_{\kappa=1}^{K} \beta_\kappa S_\kappa(\boldsymbol{\alpha}) \geqslant \theta \ \ \forall \boldsymbol{\alpha} \in \mathbb{R}^N \text{ with } \boldsymbol{0} \leqslant \boldsymbol{\alpha} \leqslant C\mathbf{1} \text{ and } \sum_{i=1}^{N} y_i \alpha_i = 0, \end{aligned} \tag{2.36}$$

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

As the finite combinations of kernels are limited up to a finite choice, this may not represent the similarity or dissimilarity of data points, particularly, for highly non-linearly distributed and large-scaled ones. Therefore, a new combination of *infinitely* many kernels in Riemann-Stieltjes integral form is proposed in [78, 80] by employing infinite and semi-infinite programming considering all elements in kernel space, named as *infinitely kernel learning* (IKL) [78, 79, 80]. Thus, the problem becomes infinite in both its dimension and its number of constraints, and called as *infinite programming* (IP). An infinite combination has the following form:

$$k_\beta(x_i, x_j) := \int_\Omega k(x_i, x_j, \omega) d\beta(\omega), \tag{2.37}$$

where β is a monotonically increasing function of integral 1, or just a probability measure on Ω and $\omega \in \Omega$ is a kernel parameter. The kernel function $k(x_i, x_j, \omega)$ is assumed to be a twice continuously differentiable function with respect to ω, i.e., $k(x_i, x_j, \cdot) \in C^2$. As infinitely many kernels is offered to cope with the restriction of the kernel combination given by finitely many pre-chosen kernels. Here, the questions on *which* combination of kernels and on the *structure* of the mixture of kernels arise.

We can record ("scanning") all possible alternatives of kernels from the kernel space by using this new formulation, and thus, the uniformity is also protected. Infinitely many kernels mean infinitely many coefficients where they are expressed with an *increasing monotonic function* via *positive measures* [78, 79]. The formulation of IKL in [78, 79, 80] is represented as follows:

$$\begin{aligned} &\max_{\theta,\beta} \ \theta \ \ (\theta \in \mathbb{R},\ \beta : \ \text{ a positive measure on } \Omega) \\ &\text{such that} \ \ \theta - \textstyle\int_\Omega T(\omega, \boldsymbol{\alpha}) d\beta(\omega) \leqslant 0 \ \ (\boldsymbol{\alpha} \in A), \\ &\qquad\qquad\qquad\qquad\qquad\qquad \textstyle\int_\Omega d\beta(\omega) = 1, \end{aligned} \tag{2.38}$$

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.

Because of the inequality constraint, there are infinitely various inequality constraints, uniform in $\boldsymbol{\alpha} \in A$, and the state variable β is from an infinite dimensional space. Thus, our problem is one of *infinite programming (IP)* [2]. The *dual* of (2.38) can be displayed as

$$\begin{aligned} &\min_{\sigma,\rho} \ \sigma \quad (\sigma \in \mathbb{R},\ \rho: \text{ a positive measure on } A) \\ &\text{such that } \ \sigma - \textstyle\int_A T(\omega, \boldsymbol{\alpha}) d\rho(\boldsymbol{\alpha}) \geqslant 0 \ (\omega \in \Omega), \\ &\qquad\qquad\qquad\qquad \textstyle\int_A d\rho(\boldsymbol{\alpha}) = 1. \end{aligned} \tag{2.39}$$

Because of the conditions $\int_\Omega d\beta(\omega) = 1$ and $\int_A d\rho(\boldsymbol{\alpha}) = 1$, positive measures β (or ρ) are probability measures, which are parameterized in this thesis through the probability density functions as in [78, 79].

Therefore, it can be seen that the primal IKL formulation (2.38) and the dual IKL formulation (2.39) are very familiar such that there is maximization instead of minimization and the direction of inequalities in the constraints are reversed in (2.39). As well, the index set A and the variable $\boldsymbol{\alpha}$ turn into Ω and ω, respectively. The objective functions of both the dual and the primal, θ and σ, are continuous and both index sets are compact. On the other hand, the primal and the dual problem are not same on the way that the sets of inequality constraints are explained [81]. (For more information, please refer to [49].)

CHAPTER 3

METHODS

3.1 Multivariate Adaptive Regression Splines

As an adaptive regression procedure, *Multivariate Adaptive Regression Splines*($MARS$) is useful technique for solving high dimensional problems (many explanatory variables). It is developed by Friedman in 1991 [26] and an important tool in statistics as well as in classification and regression. Besides, it shows a great promise for fitting nonlinear multivariate functions. By using piecewise linear regressions, MARS builds flexible models and nonlinearity of the models is approximated by having different regression slopes in the corresponding intervals of each predictor. The intervals underlying those pieces are closed and non-overlapping except of their boundaries, so the slope of each regression line can change from one interval to another one if there is a "*knot*" defined in between.

Predictor variables in the final model and their respective knots are found by a fast but intensive search procedure. As well as searching variables one by one, MARS also looks for interactions between variables in any degree [20]. The procedure of MARS can be thought of as a generalization of stepwise linear regression but it also considers transformations and interactions between the variables as well as using a stepwise procedure to introduce and delete explanatory variables. The algorithm of MARS works by partitioning each of the explanatory variables into regions, with each region having its own regression equation. Moreover, MARS has an advantage to estimate the contributions of the basis functions so that both the additive and the interactive effects of the predictors are allowed to determine the response variable [92].

The MARS method has a two-stage process to generate a model: forward and backward. In the first stage, an extra large number of basis functions (BFs) is constructed that overfit the data. Although an overfit model has a good fit to the data used to build the model, it is not generalized well to new data. To build a model with a better generalization ability, the backward pass prunes the model.

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, in the forward construction, it includes many incorrect terms. Thus, in the backward step, the "*complexity*" of the model is reduced without decreasing the fit to the data. MARS is capable of reliably tracking very complex data structures that often hide in high dimensions by allowing arbitrary shapes of BFs and their interactions [20].

Before introducing the deep concepts of MARS, let us give the 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 if they do not contribute enough to the performance of the final model. 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 that are used in applications requiring data interpolation or smoothing. A spline can be developed by dividing the region into a conventional number of regions. A knot is the boundary between regions. By obtaining a sufficient number of knots, any shape can be well approximated [106].

3.1.1 The Procedure of MARS

MARS is a nonparametric modeling approach. Although parametric modeling methods such as linear regression are relatively easy to improve and interpret, compared with nonparametric ones, they have a limited flexibility and work well only if the true underlying relationship is close to the pre-specified approximated function in the model. In order to overcome the drawbacks of the usual parametric approaches, nonparametric models are developed locally over specific subregions of the data. The data are searched for an optimum number of subregions and a simple function is optimally fit to the realizations in each subregion [107]. The nonlinearity of a model is approximated by using separate linear regression slopes in separate intervals of the independent variable space.

Let us state the general model:

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

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

MARS can be expressed 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 $x \in \mathbb{R}$ [40]:

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

In equation (3.1), $(\cdot)_+$ means that only the positive parts are used, otherwise it is given a zero value. These truncated functions are piecewise linear nonsmooth splines. The two functions are named as a *reflected pair*. Here, the objective 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 [12]

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

If all of the input values are different from each other, there will be $2Np$ BFs in total. Besides, even each BF depends only on a single x_j, it is considered as a function over the whole input space $\mathbb{R}^p$ [40].

In higher dimensions, BFs that are the tensor products of univariate spline functions are used to generalize spline fitting. Hence, 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 $s_{km} \in \{\pm 1\}$ [106].

Although the model-building strategy is similar to a forward stepwise linear regression, it is allowed to use functions from the set C and their products, instead of using the original inputs. Hence, the model takes the following form:

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

where c_0 is the intercept term and M is the number of BFs in the current model [20].

As in linear regression, given some choices for the B_m, the coefficients c_m are estimated by using the least-squares method. The construction of the functions B_m is the most important concept to generate the model. 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 [54]:

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

Each BF must have different input variables in the MARS algorithm. Therefore, the BFs above 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 [40]. Here, the coefficients $\hat{C}_{M+1}$ and $\hat{C}_{M+2}$ are estimated by least-squares method as well as all the other $M+1$ coefficients in the model. The process keeps continuing until the model set $\mathbf{M}$ includes some preset maximum number of terms. This process shows that the model set $\mathbf{M}$ actually has an *iterative* built up procedure.

There are some possible basis function candidates [54]:

- $x_j \qquad (j = 1, 2, \ldots, p)$,
- $(x_j - t_k)_+$, 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.

As these conditions force linear terms to be involved, this provides a better interpretability of the final model.

At the end of this process, a large model equation (3.2) is obtained. However, it includes some unnecessary variables and typically overfits the data. Thus, a backward deletion procedure is needed to detect and discard these variables. For this, the term whose removal causes the smallest increase in RSS is deleted at each stage. This process provides an estimated best model $\hat{f}_M$ of each size (number of terms) M. In order to estimate the optimal value of M, cross-validation can be used. However, for computational savings, the MARS procedure uses *generalized cross-validation*. This

criterion, also known as *lack-of-fit* criterion, is defined as [26]

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

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

where 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, which is penalized by the denominator. This denominator helps to balance the increasing variance in the case where the model complexity increases.

Moreover, when there are r linearly independent BFs in the model and K knots were selected in the forward stage, then, $C(M) = r + cK$. Here, the quantity c shows a cost for each BF optimization, generally equal to 3 [40]. However, if the model is additive, then a penalty of $c = 2$ is used. Besides, a smaller $C(M)$ generates a larger model with more BFs while a larger $C(M)$ creates a smaller model with less BFs. Using lack of fit criteria, the best model is reached along the backward sequence that minimizes generalized cross-validation [20, 40].

MARS is a special procedure in that it uses piecewise linear BFs and has a particular model strategy. A key characteristics of the piecewise linear BFs is their ability to operate locally; they are zero over a part of their range. When they are multiplied together, 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. Other basis functions such as polynomials can be used, however, this would produce a nonzero product everywhere, and would not work as well.

The fact that each input can appear at most once in a product is a limitation put on the formation of model terms. This avoids the formation of higher-order powers of an input, which increases or decreases too sharply near the boundaries of the factor space. Such higher-order powers can be approximated in a more stable way by using piecewise linear functions.

It is a useful option in the MARS procedure to set an upper limit on the order of

interaction. For example, choosing two as a limit allows pairwise products of piecewise linear functions but not a three-fold or any higher way of products. This can be helpful to interpret the final model. An upper limit of one results in an additive model [40].

3.1.2 Software of MARS

In this thesis, the MARS models are fitted using *MARS* (*Version 2, Salford Systems, San Diego, Calif., USA*) [106]. The MARS package developed by Salford Systems is available at [14]. MARS allows the user to set control parameters to explore different models and find the "best" model. The maximum number of knots is determined by trial and error. Besides, the maximum number of interactions can be more than the degree of two (2-way interaction). It is a well designed piece of software that implements MARS technique with user-friendly graphical interface.

3.1.3 Advantages and Disadvantages of MARS Compared other Algorithms

MARS is a useful and flexible regression technique that applies a modified recursive partitioning strategy for simplifying high-dimensional problems. Although recursive partitioning regression is a powerful method, it has some shortcomings such as discontinuity at the subregion boundaries and MARS overcomes these limitations with a modified form of it [107].

Besides, it is able to identify a relatively small number of predictor variables which are complex transformations of initial variables. As well, it is not computationally intensive and is straightforward to implement in order to look for interactions. MARS identifies interactions and also produces graphs that help visualize and understand interactions [19]. Both the additive and the interactive effects of the predictors are allowed to determine the response variable. MARS can handle complex (nonlinear) relationships and interactions as well as providing an interpretable model. MARS has automated capabilities for handling missing data, a common feature of large databases [11].

MARS is a similar approach to *CART*. They are both capable of modeling complex

relationship between variables without strong model assumptions. Besides, unlike neural networks, both of them can detect "important" independent variables through the built tree and basis functions when there are many potential variables. Moreover, CART and MARS train huge data in a short time and decrease the modeling time as well as producing easily interpreted models [108]. The similarity of the two methods is mainly on the partitioning of intervals, where two symmetric BFs are created at the knot location. However, in two ways MARS differs from decision tree techniques: (i) In the recursive partitioning strategy, MARS assigns a coefficient (a slope) to each part. In other words, while techniques such as CART use step functions to *model* the response variable, which leads to discontinuous models, MARS uses continuous piecewise linear functions. This continuity provides a more effective way to model nonlinearities and a more accurate model [99]. (ii) The recursive partitioning often results in a poor predictive ability for even low-order performance functions when new data are introduced. The MARS method overcomes these two problems of recursive partitioning regression to increase accuracy. In fact, the MARS algorithm is a modified recursive partitioning algorithm which has important advantages compared to other recursive partitioning algorithms.

As it is an exhaustive procedure to search for nonlinearities and interactions, there is a risk of overfitting. However, setting a lower maximum number of BFs and a higher "cost" per knot helps to prevent from this problem [26].

In the following section, we will present a modification on the theory of MARS by the use of modern continuous optimization. While the backward stepwise algorithm and GCV are mentioned as model-free approaches, now we are going to turn to an integrated model-based approach in the next. For this one, continuous optimization will be used, in the form of a penalized optimization problem and, then, a conic quadratic optimization problem. Thus, an alternative version of MARS, called $CMARS$ is achieved.

3.2 Conic Multivariate Adaptive Regression Splines

In the previous section, we explained MARS as a nonparametric regression procedure for high dimensional data with details. In this section, however, we introduce a

modified version of MARS called as *Conic Multivariate Adaptive Regression Splines* (*CMARS*). Here, "C" represents the word *conic* as well as *convex* and *continuous*.

MARS is a nonparametric regression procedure that makes no specific assumption about the underlying functional relationship between the dependent and independent variables to estimate general functions of high dimensional arguments given sparse data. The algorithm of MARS for estimating the model function consists of two algorithms, the forward and the backward stepwise algorithms. In CMARS, instead of using the backward stepwise algorithm, we can construct a penalized residual sum of squares for MARS as a Tikhonov regularization problem and treat this problem employing continuous optimization techniques. These techniques, in particular the framework of conic quadratic programming, are considered as an important complementary technology and model based alternative to the concept of the backward stepwise algorithm [92].

For CMARS, we use the following notation for the piecewise linear BFs:

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

where $(q)_{+} := \max\{0, q\}$ and τ is an univariate knot. Besides, we consider again the notation introduced previously in Subsection 3.1.1 to represent the relationship between input and dependent variables:

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

where Y is a response variable, $\boldsymbol{X} = (X_1, X_2, \ldots, X_p)^T$ is a vector of predictor variables and ϵ is additive random variable with zero mean and finite variance. 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. Such a nearby placement indicates a slight modification as in the previous section, the knots' values are equal to input values. Indeed, it may be assumed that without loss of generality $\tau_{i,j} \neq \tilde{x}_{i,j}$ for all i and j, so that it is possible to prevent from nondifferentiability in our optimization problem later on. Even, if the knots $\tau_{i,j}$ far away from the input values $\tilde{x}_{i,j}$ provide a better data fit, they can be chosen.

Let use the following formulation for the set of BFs:

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

If all of the input values are different from each other, there will be $2Np$ BFs in total. Thus, we can represent $f(\boldsymbol{x}_i)$ by a linear combination which is successively built up by basis functions and the intercept θ_0, such that (3.4) becomes

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

Here, 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. By multiplying an existing basis function with a truncated linear function involving a new variable, interaction basis functions are created. In this case, both the existing basis function and the newly created interaction basis function are used in the MARS approximation.

Provided the observations represented by the data $\boldsymbol{x_i}$ ($i = 1, \ldots, N$), the form of the mth basis function is as follows [92]:

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

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.

To compare the possible basis functions, a *lack-of-fit* criterion can be used. Besides, it is possible to restrict the search for new basis functions to a maximum order of interactions. For example, if only up to two-factor interactions are permitted, then $K_m \leq 2$ would be a suitable restriction.

In MARS, the backward stepwise algorithm is used to prevent from over-fitting by decreasing the complexity of the model without degrading the fit to the data. This

algorithm does this by removing from the model basis functions that contributes to the smallest increase in the residual squared error 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 which shows the lack-of-fit when using MARS. In CMARS, this criterion is defined as follows [15]:

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

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 as well as a smoothing parameter for the procedure.

In this study, we propose to not employ the backward stepwise algorithm to estimate the function $f(\boldsymbol{x})$. Instead, we will use penalty terms 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 Problem

Let us use the penalized residual sum of squares with basis functions having been accumulated in the forward stepwise algorithm. PRSS has the following form:

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

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*. In this thesis, we tackle that tradeoff by penalty methods, such as regularization techniques [3] and by conic quadratic programming [8, 70].

The integrals of the first-order derivatives measure the flatness of the model functions

while the integrals of the second-order derivatives measure unstability and complexity inscribed into the model (via the model functions) [40, 90]. Furthermore, 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 $\alpha_1, \alpha_2 \in (0, 1)$.

Indeed, we 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 [93].

If we consider the representations (3.6) and (3.7) in (3.9), then the objective function (3.9) will be as follows [93]:

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

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 output $\tilde{y}_i$.

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

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

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. 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, we write the discretized form of the integrals as follows:

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

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

We can rearrange PRSS in this form:

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

where $(\sigma^{\kappa_j})_{j\in\{1,2,\ldots,p\}} \in \{0,1,2,\ldots,N+1\}^{K_m}$. Let us introduce some more notation related with the sequence (σ^{κ_j}) [93]:

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

It is possible to approximate $PRSS$ by using (3.14) as follows:

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

For a short representation, we can rewrite the approximate relation (3.13) as in the following:

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

where $\boldsymbol{\psi}(\tilde{\boldsymbol{d}}) = \left(\boldsymbol{\psi}(\tilde{\boldsymbol{d}}_1), \ldots, \boldsymbol{\psi}(\tilde{\boldsymbol{d}}_N)\right)^T$ is an $(N \times (M_{max}+1))$-matrix and the numbers

L_{im}^2 are defined by their roots

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

3.2.2 Tikhonov Regularization Applied

In this part, we will represent $PRSS$ as a Tikhonov regularization problem [3]. If we consider equation (3.16), we can write $PRSS$ as follows [93]:

$$\begin{aligned} PRSS &\approx \left\| \boldsymbol{y} - \boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta} \right\|_2^2 + \sum_{m=1}^{M_{max}} \lambda_m \sum_{i=1}^{(N+1)^{K_m}} L_{im}^2 \theta_m^2 \\ &= \left\| \boldsymbol{y} - \boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta} \right\|_2^2 + \left(\left[L_{1m}\theta_m, L_{2m}\theta_m, \ldots, L_{(N+1)^{K_m} m}\theta_m \right] \begin{bmatrix} L_{1m}\theta_m \\ L_{2m}\theta_m \\ \vdots \\ L_{(N+1)^{K_m} m}\theta_m \end{bmatrix} \right) \\ &= \left\| \boldsymbol{y} - \boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta} \right\|_2^2 + \sum_{m=1}^{M_{max}} \lambda_m \| \boldsymbol{L}_m \theta_m \|_2^2 \\ &= \left\| \boldsymbol{y} - \boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta} \right\|_2^2 + \lambda_1 \| \boldsymbol{L}_1 \theta_1 \|_2^2 + \lambda_2 \| \boldsymbol{L}_2 \theta_2 \|_2^2 + \ldots + \lambda_{M_{max}} \| \boldsymbol{L}_{M_{max}} \theta_{M_{max}} \|_2^2, \end{aligned} \tag{3.17}$$

where $\boldsymbol{L}_m := (L_{1m}, L_{2m}, \ldots, L_{(N+1)^{K_m},m})^T$ $(m = 1, 2, \ldots, M_{max})$. To turn this equation into a *Tikhonov Regularization Problem* with a single tradeoff parameter, we make a uniform penalization by taking the same λ for each derivative term, i.e., $\lambda_1 = \lambda_2 = \ldots = \lambda_{M_{max}} =: \lambda$, where $\lambda_m \geq 0$ $(m = 1, 2, \ldots, M_{max})$. Thus, the approximation is in the following form:

$$PRSS \approx \left\| \boldsymbol{y} - \boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta} \right\|_2^2 + \lambda \| \boldsymbol{L}\boldsymbol{\theta} \|_2^2, \tag{3.18}$$

where $\boldsymbol{\theta}$ is 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:

$$\boldsymbol{L} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & L_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & L_{M_{max}} \end{bmatrix}. \tag{3.19}$$

Hence, our PRSS problem turns into a *Tikhonov regularization problem* (2.31), where $\lambda = \varphi^2$ for some $\varphi \in \mathbb{R}$ [3].

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 $\|\boldsymbol{L\theta}\|_2^2$. We select the solution such that it minimizes both first objective function $\left\| \boldsymbol{y} - \boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta} \right\|_2^2$ and second objective ($\|\boldsymbol{L\theta}\|_2^2$) in the sense of a compromise (tradeoff) solution. For a new contribution to the dependence of locally linear embedding on regularization parameter(s) we refer to [67].

Moreover, our *PRSS* problem also can be turned into a *Conic Quadratic Problem*. In Section 3.4, the implementation of CMARS algorithm will be explained with two numerical examples. Furthermore, we will compare the Tikhonov regularization solutions whose results are obtained from the thesis [49], which is on progress, with the ones coming from Conic Quadratic Programming.

3.2.3 An Alternative for Tikhonov Regularization Problem with Conic Quadratic Programming

The Tikhonov regularization problem (3.18) can be expressed as a CQP problem. Indeed, based on an appropriate choice of a bound $\tilde{M}$, we state the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \left\| \boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta} - \boldsymbol{y} \right\|_2^2 \\ \text{subject to} \quad & \|\boldsymbol{L\theta}\|_2^2 \leq \tilde{M}. \end{aligned} \tag{3.20}$$

Let us underline that this choice of $\tilde{M}$ should be the outcome of a careful learning process, with the help of model-free or model-based methods [40]. In (3.20), we have the least-squares objective function $\left\| \boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta} - \boldsymbol{y} \right\|_2^2$ and the inequality constraint function $-\|\boldsymbol{L\theta}\|_2^2 + \tilde{M}$ which is requested to be nonnegative for feasibility. Now, we equivalently write our optimization problem as follows:

$$\begin{aligned} & \min_{t,\boldsymbol{\theta}} \quad t, \\ \text{subject to} \quad & \left\| \boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta} - \boldsymbol{y} \right\|_2^2 \leq t^2, \\ & \|\boldsymbol{L\theta}\|_2^2 \leq \tilde{M}, \quad t \geq 0. \end{aligned} \tag{3.21}$$

or equivalently again,

$$\begin{aligned} &\min_{t,\boldsymbol{\theta}} \quad t, \\ \text{subject to} \quad &\left\|\boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta}-\boldsymbol{y}\right\|_2 \le t, \\ &\|\boldsymbol{L}\boldsymbol{\theta}\|_2 \le \sqrt{\tilde{M}}. \end{aligned} \tag{3.22}$$

By using modern methods of *continuous optimization techniques*, especially, from CQP where we use the basic notation as follows [91]:

$$\min_{\boldsymbol{x}} \quad \boldsymbol{c}^T\boldsymbol{x}, \quad \text{subject to} \quad \|\boldsymbol{D}_i\boldsymbol{x}-\boldsymbol{d}_i\|_2^2 \le \boldsymbol{p}_i^T\boldsymbol{x}-\boldsymbol{q}_i \;\; (i=1,2,...,k). \tag{3.23}$$

In fact, we see that our optimization problem is such a CQP program with

$$\boldsymbol{c}=(1,\boldsymbol{0}_{M_{max}+1}^T)^T,\ \boldsymbol{x}=(t,\boldsymbol{\theta}^T)^T,\ \boldsymbol{D}_1=(\boldsymbol{0}_N,\boldsymbol{\psi}(\tilde{\boldsymbol{d}})),\ \boldsymbol{d}_1=\boldsymbol{y},\ \boldsymbol{p}_1=(1,0,...,0)^T,\ q_1=0,$$

$$\boldsymbol{D}_2=(\boldsymbol{0}_{M_{max}+1},\boldsymbol{L}),\ \boldsymbol{d}_2=\boldsymbol{0}_{M_{max}+1},\ \boldsymbol{p}_2=\boldsymbol{0}_{M_{max}+2} \text{ and } q_2=-\sqrt{\tilde{M}}.$$

In order to write the optimality condition, the dual problem and the primal dual optimal solution for this problem, and we firstly reformulate the problem (3.22) as follows:

$$\begin{aligned} &\min_{t,\boldsymbol{\theta}} \quad t, \\ \text{such that} \quad &\boldsymbol{\chi} := \begin{bmatrix} \boldsymbol{0}_N & \boldsymbol{\psi}(\tilde{\boldsymbol{d}}) \\ 1 & \boldsymbol{0}_{M_{max}+1}^T \end{bmatrix} \begin{bmatrix} t \\ \boldsymbol{\theta} \end{bmatrix} + \begin{bmatrix} -\boldsymbol{y} \\ 0 \end{bmatrix}, \\ &\boldsymbol{\eta} := \begin{bmatrix} \boldsymbol{0}_{M_{max}+1} & \boldsymbol{L} \\ 0 & \boldsymbol{0}_{M_{max}+1}^T \end{bmatrix} \begin{bmatrix} t \\ \boldsymbol{\theta} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0}_{M_{max}+1} \\ \sqrt{\tilde{M}} \end{bmatrix}, \\ &\boldsymbol{\chi} \in L^{N+1},\ \boldsymbol{\eta} \in L^{M_{max}+2}, \end{aligned} \tag{3.24}$$

where L^{N+1}, $L^{M_{max}+2}$ are the $(N+1)$- and $(M_{max}+2)$-dimensional *ice-cream (or second-order, or Lorentz) cones*, defined by:

$$L^{p+1} := \left\{\boldsymbol{x}=(x_1,x_2,...,x_{p+1})^T \in \mathbb{R}^{p+1} \mid x_{p+1} \ge \sqrt{x_1^2+x_2^2+...+x_p^2}\right\} \;\; (p \ge 1).$$

The *dual problem* to the latter primal one is given by

$$\begin{aligned} &\max \quad (\boldsymbol{y}^T,0)\omega_1 + \left(\boldsymbol{0}_{M_{max}+1}^T, -\sqrt{\tilde{M}}\right)\omega_2 \\ \text{such that} \quad &\boldsymbol{\chi} := \begin{bmatrix} \boldsymbol{0}_N^T & 1 \\ \boldsymbol{\psi}(\tilde{\boldsymbol{d}}) & \boldsymbol{0}_{M_{max}+1}^T \end{bmatrix} \boldsymbol{\omega}_1 + \begin{bmatrix} \boldsymbol{0}_{M_{max}+1}^T & 0 \\ \boldsymbol{L}^T & \boldsymbol{0}_{M_{max}+1} \end{bmatrix} \boldsymbol{\omega}_2 = \begin{bmatrix} 1 \\ \boldsymbol{0}_{M_{max}+1} \end{bmatrix}, \\ &\boldsymbol{\omega}_1 \in L^{N+1},\ \boldsymbol{\omega}_2 \in L^{M_{max}+2}. \end{aligned} \tag{3.25}$$

Furthermore, $(t, \boldsymbol{\theta}, \boldsymbol{\chi}, \boldsymbol{\eta}, \boldsymbol{\omega}_1, \boldsymbol{\omega}_2)$ is a *primal dual optimal solution* if and only if [93]

$$\boldsymbol{\chi} := \begin{bmatrix} \mathbf{0}_N & \boldsymbol{\psi}(\tilde{\boldsymbol{d}}) \\ 1 & \mathbf{0}_{M_{max}+1}^T \end{bmatrix} \begin{bmatrix} t \\ \boldsymbol{\theta} \end{bmatrix} + \begin{bmatrix} -\boldsymbol{y} \\ 0 \end{bmatrix},$$

$$\boldsymbol{\eta} := \begin{bmatrix} \mathbf{0}_{M_{max}+1} & \boldsymbol{L} \\ 0 & \mathbf{0}_{M_{max}+1}^T \end{bmatrix} \begin{bmatrix} t \\ \boldsymbol{\theta} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{M_{max}+1} \\ \sqrt{\tilde{M}} \end{bmatrix},$$

$$\begin{bmatrix} \mathbf{0}_N^T & 1 \\ \boldsymbol{\psi}(\tilde{\boldsymbol{d}}) & \mathbf{0}_{M_{max}+1}^T \end{bmatrix} \boldsymbol{\omega}_1 + \begin{bmatrix} \mathbf{0}_{M_{max}+1}^T & 0 \\ \boldsymbol{L}^T & \mathbf{0}_{M_{max}+1} \end{bmatrix} \boldsymbol{\omega}_2 = \begin{bmatrix} 1 \\ \mathbf{0}_{M_{max}+1} \end{bmatrix},$$

$$\boldsymbol{\omega}_1^T \boldsymbol{\chi} = 0, \quad \boldsymbol{\omega}_2^T \boldsymbol{\eta} = 0,$$

$$\boldsymbol{\omega}_1 \in L^{N+1}, \quad \boldsymbol{\omega}_2 \in L^{M_{max}+2},$$

$$\boldsymbol{\chi} \in L^{N+1}, \quad \boldsymbol{\eta} \in L^{M_{max}+2}. \tag{3.26}$$

In order to provide with some fundamental facts on the solution methods for CQP and convex problem classes beyond [93], we have stated the Section 2.6 of this thesis.

In this section, we investigated CMARS model which based on the regularization of the complexity term. In the following section, we also mention the regularization of linear part (RSS term) which is investigated in [89].

3.3 The Generalized Partial Linear Model with CMARS

The GPLM model is given in Section 2.4 by the following formula:

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

where $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_m)^T$ is a finite dimensional parameter and $\gamma(\cdot)$ is a smooth function which we try to estimate by CMARS. Also, we assume that some vectors $\boldsymbol{X}$ and $\boldsymbol{T}$ come from a decomposition of explanatory variables. 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.

There are different kinds of estimation methods for a generalized partial linear model (GPLM). Müller (2001) [89] studied different estimation methods based on kernel methods and test procedures on the correct specification of this model. In this thesis,

we focus on special types of estimation of $\gamma(\cdot)$ by CMARS and $\boldsymbol{\beta}$ by least-square estimation with Tikhonov Regularization [89].

3.3.1 Least-Squares Estimation with Tikhonov Regularization

Let us recall the model (2.8), where $G = H^{-1}$ is assumed to be a known *link function* which connects the mean of the dependent variable, $\mu = E(Y|\mathbf{X}, \boldsymbol{T})$, to the predictors. Here, the equation (2.8) can be considered as a semiparametric GLM as in the equation (2.9), because all terms are linear except one; i.e,

$$H(\mu) = \eta(\mathbf{X}, \boldsymbol{T}) = \mathbf{X}^T\boldsymbol{\beta} + \gamma(\boldsymbol{T}) = \sum_{j=1}^{m} X_j\beta_j + \gamma(\boldsymbol{T}). \tag{3.27}$$

Now, to obtain the GPLM, we consider observation values y_i, $\boldsymbol{x}_i$, $\boldsymbol{t}_i$ $(i = 1, 2, \ldots, n)$. Then, $\mu_i = G(\eta_i)$ and $\eta_i{=}H(\mu_i){=}\boldsymbol{x}_i^T\boldsymbol{\beta}{+}\gamma(\boldsymbol{t}_i)$ with smooth function $\gamma(\cdot)$.

To determine the knots of MARS based on the resulting residuals, we apply linear least-squares estimation with Tikhonov regularization on the given data to find a vector $\boldsymbol{\beta}$ (including intercept term), for a pre-estimation of parametric part:

$$y^{preproc} = \boldsymbol{X}^T\boldsymbol{\beta}^{preproc} + \epsilon = \beta_0 + \sum_{j=1}^{m} X_j\beta_j + \epsilon, \tag{3.28}$$

In order to estimate the regression coefficients, the method of least-squares is used; $\boldsymbol{\beta}^{preproc} = (\beta_0, \beta_1, \beta_2, \ldots, \beta_m)^T$ in $y^{preproc} = \beta_0 + \sum_{j=1}^{m} X_j\beta_j$ to minimize the residual sum of squares (RSS). Tikhonov Regularization proposed an approximate solution to (3.28) by minimizing a quadratic functional:

$$\min_{\boldsymbol{\beta}^{preproc}} \|\boldsymbol{y}^{preproc} - \boldsymbol{X}\boldsymbol{\beta}^{preproc}\|_2^2 + \lambda\|\boldsymbol{L}\boldsymbol{\beta}^{preproc}\|_2^2, \tag{3.29}$$

where λ is a regularization parameter between the first and the second part. The terms $\boldsymbol{y}^{preproc}$ and $\boldsymbol{\beta}^{preproc}$ represent the response vector and unknown coefficients, respectively. They are obtained by solving a Tikhonov regularization problem (3.29). Generally, a Tikhonov regularization problem may comprise higher-order Tikhonov regularization and it can be solved using generalized singular value decomposition (GSVD). After getting the regression coefficients, the linear least-squares model is subtracted (without intercept) at the data from corresponding responses

$$\boldsymbol{y} - \bar{\boldsymbol{X}}\boldsymbol{\beta}^{preproc} = \hat{\boldsymbol{y}} = \eta. \tag{3.30}$$

where $\bar{\boldsymbol{X}}$ is the design matrix $\boldsymbol{X}$, except its first column, and $\hat{\boldsymbol{y}}$ is the resulting vector of residuals, regarded as our new observations. On our input data and these new responses, we establish our knot selection by MARS.

3.3.2 CMARS Method for the Nonparametric Part

In the model (2.8), $\gamma(\cdot)$ is a smooth function which we try to estimate by Conic Multivariate Adaptive Regression Splines (CMARS) which is an alternative technique to multivariate adaptive regression splines (MARS).

Here, as done previously in (3.6), $\gamma(\boldsymbol{t}_i)$ can be represented by a linear combination of successively built up by basis functions and the intercept θ_0. Then, (2.9) becomes as follows:

$$\eta_i = H(\mu) = \boldsymbol{x}_i^T\boldsymbol{\beta} + \theta_0 + \sum_{m=1}^{M} \theta_m \psi_m(\boldsymbol{t_i}). \tag{3.31}$$

Here, θ_m is the unknown coefficient for the mth basis function ($m = 1, \ldots, M$), θ_0 is the constant term, $\boldsymbol{\psi_m}$ ($m = 1, \ldots, M$) 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. A set of eligible knots $\tau_{i,j}$ (selected by MARS with reference to the residual vector) is assigned separately for each input variable dimension, and it is chosen to approximately coincide with the input levels represented in the data. By multiplying an existing basis function with a truncated linear function involving a new variable, interaction basis functions are created. In this case, both the existing basis function and the newly created interaction basis function are used in the MARS approximation.

Provided the observations represented by the data $\boldsymbol{t_i}$ ($i = 1, \ldots, N$), the form of the mth basis function is as follows [92]:

$$\psi_m(\boldsymbol{t}) := \prod_{j=1}^{K_m} (s_{\kappa_j^m} \cdot (t_{\kappa_j^m} - \tau_{\kappa_j^m}))_+, \tag{3.32}$$

where K_m is the number of truncated linear functions multiplied in the mth basis function, $t_{\kappa_j^m}$ 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.

In MARS, the backward stepwise algorithm is used to prevent from over-fitting by decreasing the complexity of the model without degrading the fit to the data. 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. To estimate the optimal value of α, generalized cross-validation can be used which shows the lack-of-fit when using MARS. In CMARS, this criterion is defined as follows [15]:

$$GCV := \frac{\sum_{i=1}^{N}(\eta_i - \boldsymbol{x}_i^T\hat{\boldsymbol{\beta}} - \hat{\gamma}_\alpha(\boldsymbol{t}_i))^2}{(1 - \boldsymbol{M}(\alpha)/N)^2}, \tag{3.33}$$

where $\boldsymbol{M}(\alpha) := u + dK$ [89]. 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.

As in CMARS model, in order 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 in order 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

It is possible to write the equation (2.9) as follows:

$$\eta = H(\mu_i) = \boldsymbol{x}_i^T\boldsymbol{\beta} + \boldsymbol{\psi}^T(\boldsymbol{t}_i)\boldsymbol{\theta}, \tag{3.34}$$

where $\boldsymbol{\theta}$=$(\theta_0, \theta_1, \ldots, \theta_M)^T$ and $\boldsymbol{\psi}(\tilde{\boldsymbol{d}}_{\boldsymbol{i}})$=$(\psi_1(t_i), \psi_2(t_i), \ldots, \psi_M(t_i))$. The penalized residual sum of squares (PRSS) with basis functions having been accumulated in the forward stepwise algorithm for the GPLM model with CMARS is as follows:

$$PRSS = \sum_{i=1}^{N}\left(\eta_i - \boldsymbol{x}_i^T\boldsymbol{\beta} - \boldsymbol{\psi}^T(\boldsymbol{t}_i)\boldsymbol{\theta}\right)^2 +$$
$$\sum_{m=1}^{M_{max}}\lambda_m \sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V(m)}} \int \theta_m^2\left[D_{r,s}^{\boldsymbol{\alpha}}\psi_m(\boldsymbol{t}^m)\right]^2 d\boldsymbol{t}^m, \tag{3.35}$$

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. Furthermore, we refer to

$$D_{r,s}^{\boldsymbol{\alpha}}\psi_m(\boldsymbol{t}^m) := \frac{\partial^{\,\boldsymbol{\alpha}}\psi_m}{\partial^{\alpha_1}\boldsymbol{t}_r^m\,\partial^{\alpha_2}\boldsymbol{t}_s^m}(\boldsymbol{t}^m) \tag{3.36}$$

for $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)^T$, $|\,\boldsymbol{\alpha}| = \alpha_1 + \alpha_2$, where $\alpha_1, \alpha_2 \in \{0, 1\}$. The optimization problem bases on the *tradeoff* between both *accuracy*, i.e., a small sum of error squares, and *not too high a complexity*. This tradeoff is established through the penalty parameters λ_m. In this study, we tackle that tradeoff by penalty methods, such as regularization techniques and by conic quadratic programming. In subsection 3.3.4, we shall extend this regularization by including Tikhonov regularization of the linear part.

If we consider the representations (3.31) and (3.32) in (3.35), then the objective function (3.35) will be as follows [93]:

$$\begin{aligned} PRSS = \sum_{i=1}^{N}\left(\eta_i - \boldsymbol{x}_i^T\boldsymbol{\beta} - \theta_0 - \sum_{m=1}^{M}\theta_m\psi_m(\boldsymbol{t}_i^m) - \sum_{m=M+1}^{M_{max}}\theta_m\psi_m(\boldsymbol{t}_i^m)\right)^2 \\ + \sum_{m=1}^{M_{max}}\lambda_m \sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2}\ \sum_{\substack{r<s \\ r,s\in V(m)}}\int \theta_m^2\left[D_{r,s}^{\boldsymbol{\alpha}}\psi_m(\boldsymbol{t}^m)\right]^2 d\boldsymbol{t}^m, \end{aligned} \tag{3.37}$$

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. We recall that those coordinates are collected in the set $V(m)$. Let us note here that the second-order derivatives of the piecewise linear functions ψ_m $(m = 1, 2, ..., M)$ and, hence, the penalty terms related are vanishing. Now, we can rearrange the representation of $PRSS$ as follows:

$$\begin{aligned} PRSS \ :=\ & \sum_{i=1}^{N}(y_i - \boldsymbol{x}_i^T\boldsymbol{\beta} - \psi(\tilde{d}_i)^T\theta)^2 \\ +\ & \sum_{m=1}^{M_{max}}\lambda_m \sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2}\ \sum_{\substack{r<s \\ r,s\in V(m)}}\int \theta_m^2\left[D_{r,s}^{\boldsymbol{\alpha}}\psi_m(\boldsymbol{t}^m)\right]^2 d\boldsymbol{t}^m, \end{aligned} \tag{3.38}$$

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, we write the discretized form of the integrals as follows:

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

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

We can rearrange PRSS in this form:

$$\begin{aligned} PRSS &\approx \sum_{i=1}^{N} \left(\eta_i - \boldsymbol{x}_i^T\boldsymbol{\beta} - \boldsymbol{\psi}(\tilde{\boldsymbol{d}}_i)^T\boldsymbol{\theta}\right)^2 \\ &+ \sum_{m=1}^{M_{max}} \lambda_m \sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V(m)}} \sum_{(\sigma^{\kappa_j})} \theta_m^2 \cdot \left[D_{r,s}^{\boldsymbol{\alpha}} \psi_m(t_{\substack{l^{\kappa_j^m},\kappa_j^m \\ \sigma^{\kappa_j^m}}}, \ldots, t_{\substack{l^{\kappa_{K_m}^m},\kappa_{K_m}^m \\ \sigma^{\kappa_{K_m}^m}}})\right]^2 \\ &\cdot \prod_{j=1}^{K_m} \left(t_{\substack{l^{\kappa_j^m},\kappa_j^m \\ \sigma^{\kappa_{j+1}}}} - t_{\substack{l^{\kappa_j^m},\kappa_j^m \\ \sigma^{\kappa_j}}}\right), \end{aligned} \tag{3.39}$$

where $(\sigma^{\kappa_j})_{j\in\{1,2,\ldots,p\}} \in \{0,1,2,\ldots,N+1\}^{K_m}$. Let us introduce some more notation related with the sequence (σ^{κ_j}) [93]:

$$\hat{\boldsymbol{t}}_i^m = \left(t_{\substack{l^{\kappa_j^m},\kappa_j^m \\ \sigma^{\kappa_j^m}}}, \ldots, t_{\substack{l^{\kappa_{K_m}^m},\kappa_{K_m}^m \\ \sigma^{\kappa_{K_m}^m}}}\right), \quad \Delta\hat{\boldsymbol{t}}_i^m := \prod_{j=1}^{K_m} \left(t_{\substack{l^{\kappa_j^m},\kappa_j^m \\ \sigma^{\kappa_{j+1}}}} - t_{\substack{l^{\kappa_j^m},\kappa_j^m \\ \sigma^{\kappa_j}}}\right). \tag{3.40}$$

It is possible to approximate $PRSS$ by using (3.40) as follows:

$$\begin{aligned} PRSS &\approx \sum_{i=1}^{N} \left(\eta_i - \boldsymbol{x}_i^T\boldsymbol{\beta} - \boldsymbol{\psi}(\tilde{\boldsymbol{d}}_i)^T\boldsymbol{\theta}\right)^2 \\ &+ \sum_{m=1}^{M_{max}} \lambda_m \theta_m^2 \sum_{i=1}^{(N+1)^{K_m}} \left(\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V(m)}} \left[D_{r,s}^{\boldsymbol{\alpha}} \psi_m(\hat{\boldsymbol{t}}_i^m)\right]^2\right) \Delta\hat{\boldsymbol{t}}_i^m. \end{aligned} \tag{3.41}$$

For a short representation, we can rewrite the approximate relation (3.39) as in the following:

$$PRSS \approx \left\|\boldsymbol{\eta} - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta}\right\|_2^2 + \sum_{m=1}^{M_{max}} \lambda_m \sum_{i=1}^{(N+1)^{K_m}} L_{im}^2 \theta_m^2, \tag{3.42}$$

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

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

3.3.4 Tikhonov Regularization Applied in GPLM with CMARS

If we consider equation (3.42), we can write $PRSS$ as follows [93]:

$$PRSS \approx \|\boldsymbol{\eta} - \boldsymbol{X}^*\boldsymbol{\beta}^*\|_2^2 + \sum_{m=1}^{M_{max}} \lambda_m \sum_{i=1}^{(N+1)^{K_m}} L_{im}^2 \theta_m^2, \tag{3.43}$$

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. This problem may be ill-posed (irregular or unstable). For this reason, we approach our problem $PRSS$ as a *Tikhonov regularization problem* [70] because Tikhonov regularization belongs to the most commonly used methods of making ill-posed problems well-posed (regular or stable). 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 $\boldsymbol{\eta} = \boldsymbol{X}^*\boldsymbol{\beta}^*$.

For this purpose we consider formula (3.43) again, arranging it as follows:

$$\begin{aligned} PRSS &\approx \|\boldsymbol{\eta} - \boldsymbol{X}^*\boldsymbol{\beta}^*\|_2^2 + \sum_{m=1}^{M_{max}} \lambda_m \sum_{i=1}^{(N+1)^{K_m}} L_{im}^2 \theta_m^2 \\ &= \sum_{m=1}^{M_{max}} \lambda_m \left[(\boldsymbol{L}_{1m}\theta_m)^2 + (\boldsymbol{L}_{2m}\theta_m)^2 + \ldots + (\boldsymbol{L}_{(N+1)^{K_m}m}\theta_m)^2 \right] \\ &= \|\boldsymbol{\eta} - \boldsymbol{X}^*\boldsymbol{\beta}^*\|_2^2 + \sum_{m=1}^{M_{max}} \lambda_m \|\boldsymbol{L}_m\theta_m\|_2^2, \end{aligned} \tag{3.44}$$

where $\boldsymbol{L}_m := (L_{1m}, L_{2m}, \ldots, L_{(N+1)^{K_m},m})^T$ $(m = 1, 2, \ldots, M_{max})$. To turn this equation into a *Tikhonov Regularization Problem* with a single tradeoff parameter, we make a uniform penalization by taking the same λ for each derivative term, i.e., $\lambda_1 = \lambda_2 = \ldots = \lambda_{M_{max}} =: \lambda$, where $\lambda_m \geq 0$ $(m = 1, 2, \ldots, M_{max})$. Thus, the

approximation is in the following form:

$$PRSS \quad \approx \quad \|\boldsymbol{\eta} - \boldsymbol{X}^*\boldsymbol{\beta}^*\|_2^2 + \lambda\|\boldsymbol{L}\boldsymbol{\theta}\|_2^2, \tag{3.45}$$

where $\boldsymbol{\theta}$ is 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 with first column $\boldsymbol{L}_0 = \boldsymbol{0}_{N+1^{K_m}}$ and the other columns being the vectors $\boldsymbol{L}_m$ introduced above. Let us consider the high-dimensional matrix $\boldsymbol{L}^* = (\boldsymbol{R}^*,\ \boldsymbol{L})$, where $\boldsymbol{R}^*$ is an $((M_{max}+1)\times p)$-regularization matrix with entries being first or second derivative of $\boldsymbol{\beta}$. These derivatives are given by first- or second-order difference quotients of $\boldsymbol{\beta}$, regarded as a function that is evaluated at the points i and $i+1$. These difference quotients approximate first- and second-order derivatives; altogether, they are comprised by products $\boldsymbol{R}^*\boldsymbol{\beta}$ of $\boldsymbol{\beta}$ with matrices $\boldsymbol{R}^*$ that represent the discrete differential operators of first- and second order, respectively. Then, our *PRSS* problem looks as a classical *Tikhonov regularization problem* [70] with $\varphi > 0$, i.e., $\lambda = \varphi^2$ for some $\varphi \in \mathbb{R}$ is as follow. 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 select the solution such that it minimizes both the objective function $\|\boldsymbol{\eta} - \boldsymbol{X}^*\boldsymbol{\beta}^*\|_2^2$ and the regularization objective $\|\boldsymbol{L}^*\boldsymbol{\beta}^*\|_2^2$ in the sense of a compromise (tradeoff) solution. For a new contribution to the dependence of locally linear embedding on regularization parameter(s) we refer to [67]. Now, our *PRSS* problem can again be turned into a *conic quadratic problem.*

3.3.5 An Alternative for Tikhonov Regularization Problem with Conic Quadratic Programming

As we mentioned in the previous section, the Tikhonov regularization problem (3.45) can be expressed as a CQP problem. Indeed, based on an appropriate choice of a bound $\tilde{M}$, we state the following optimization problem:

$$\begin{aligned} \min \quad & \|\boldsymbol{\eta} - \boldsymbol{X}^*\boldsymbol{\beta}^*\|_2^2 \\ \text{subject to} \quad & \|\boldsymbol{L}^*\boldsymbol{\beta}^*\|_2^2 \leq \tilde{M}. \end{aligned} \tag{3.46}$$

Let us underline that this choice of $\tilde{M}$ should be the outcome of a careful learning process, with the help of model-free or model-based methods [40]. In (3.46), we have the least-squares objective function $\|\boldsymbol{\eta} - \boldsymbol{X}^*\boldsymbol{\beta}^*\|_2^2$ and the inequality constraint

function $-\|\boldsymbol{L}^*\boldsymbol{\beta}^*\|_2^2+\tilde{M}$ which is requested to be nonnegative for feasibility. Now, we equivalently write our optimization problem as follows:

$$\begin{aligned} &\min_{z,\boldsymbol{\beta}^*} \quad z, \\ \text{subject to} \quad &\|\boldsymbol{\eta}-\boldsymbol{X}^*\boldsymbol{\beta}^*\|_2^2 \le z^2, \\ &\|\boldsymbol{L}^*\boldsymbol{\beta}^*\|_2^2 \le \tilde{M},\ z \ge 0. \end{aligned} \tag{3.47}$$

or equivalently again,

$$\begin{aligned} &\min_{z,\boldsymbol{\beta}^*} \quad z, \\ \text{subject to} \quad &\|\boldsymbol{\eta}-\boldsymbol{X}^*\boldsymbol{\beta}^*\|_2 \le z, \\ &\|\boldsymbol{L}^*\boldsymbol{\beta}^*\|_2 \le \sqrt{\tilde{M}}. \end{aligned} \tag{3.48}$$

A CQP problem is generally expressed as the following [91]:

$$\min_{\boldsymbol{u}} \quad \boldsymbol{c}^T\boldsymbol{u}, \quad \text{subject to} \quad \|\boldsymbol{D}_i\boldsymbol{u}-\boldsymbol{d}_i\|_2^2 \le \boldsymbol{p}_i^T\boldsymbol{u}-q_i \ \ (i=1,2,\ldots,k). \tag{3.49}$$

In fact, we see that our optimization problem is such a CQP program with

$$\boldsymbol{c}=(1,\boldsymbol{0}_{M_{max}+1}^T)^T,\ \boldsymbol{u}=(z,\boldsymbol{\beta}^{*T})^T,\ \boldsymbol{D}_1=(\boldsymbol{0}_N,\boldsymbol{X}^*),\ \boldsymbol{d}_1=\boldsymbol{\eta},\ \boldsymbol{p}_1=(1,0,\ldots,0)^T,\ q_1=0,$$

$$\boldsymbol{D}_2=(\boldsymbol{0}_{M_{max}+1},\boldsymbol{L}^*),\ \boldsymbol{d}_2=\boldsymbol{0}_{M_{max}+1},\ \boldsymbol{p}_2=\boldsymbol{0}_{M_{max}+p+2} \text{ and } q_2=-\sqrt{\tilde{M}}.$$

Having written the Tikhonov regularization task for GPLM including MARS for $\gamma(\boldsymbol{T})$ and estimating it with a CQP problem, we will call it *CGPLMARS*. CGPLMARS provides a solution by applying the developed CQP techniques. These kinds of well-structured convex optimization problems have also been used by Weber et al. for new approaches to regression and classification. In this respect, CGPLMARS has the advantage of higher speed and less complexity, and it permits the use of interior point methods [39].

In order to write the optimality condition, the dual problem and the primal dual optimal solution for this problem, and we firstly reformulate the problem (3.22) as

follows:

$$\begin{aligned} & \min_{z,\boldsymbol{\eta}^*} \quad z, \\ \text{such that} \quad & \boldsymbol{\chi} := \begin{bmatrix} \mathbf{0}_N & \boldsymbol{X}^* \\ 1 & \mathbf{0}_{M_{max}+1+p}^T \end{bmatrix} \begin{bmatrix} z \\ \boldsymbol{\beta}^* \end{bmatrix} + \begin{bmatrix} -\boldsymbol{\eta} \\ 0 \end{bmatrix}, \\ & \boldsymbol{\tau} := \begin{bmatrix} \mathbf{0}_{M_{max}+1} & \boldsymbol{L}^* \\ 0 & \mathbf{0}_{M_{max}+1+p}^T \end{bmatrix} \begin{bmatrix} z \\ \boldsymbol{\beta}^* \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{M_{max}+1} \\ \sqrt{\tilde{M}} \end{bmatrix}, \\ & \boldsymbol{\chi} \in L^{N+1},\ \boldsymbol{\tau} \in L^{M_{max}+2}, \end{aligned} \tag{3.50}$$

where L^{N+1}, $L^{M_{max}+2}$ are the $(N+1)$- and $(M_{max}+2)$-dimensional *ice-cream (or second-order, or Lorentz) cones*, defined by

$$L^{p+1} := \left\{ \boldsymbol{x} = (x_1, x_2, ..., x_{N+1})^T \in \mathbb{R}^{N+1} \mid x_{N+1} \geq \sqrt{x_1^2 + x_2^2 + ... + x_N^2} \right\} \ (N \geq 1).$$

The *dual problem* to the latter primal one is given by

$$\begin{aligned} & \max \quad (\boldsymbol{\eta}^T, 0)\boldsymbol{\omega}_1 + \left(\mathbf{0}_{M_{max}+1}^T, -\sqrt{\tilde{M}}\right)\boldsymbol{\omega}_2 \\ \text{such that} \quad & \begin{bmatrix} \mathbf{0}_N^T & 1 \\ \boldsymbol{\psi}(\tilde{\boldsymbol{d}}) & \mathbf{0}_{M_{max}+1}^T \end{bmatrix} \boldsymbol{\omega}_1 + \begin{bmatrix} \mathbf{0}_{M_{max}+1}^T & 0 \\ \boldsymbol{L}^{*T} & \mathbf{0}_{M_{max}+1+p} \end{bmatrix} \boldsymbol{\omega}_2 = \begin{bmatrix} 1 \\ \mathbf{0}_{M_{max}+1+p} \end{bmatrix}, \\ & \boldsymbol{\omega}_1 \in L^{N+1},\ \boldsymbol{\omega}_2 \in L^{M_{max}+2}. \end{aligned} \tag{3.51}$$

Moreover, $(z, \boldsymbol{\beta}^*, \boldsymbol{\chi}, \boldsymbol{\tau}, \boldsymbol{\omega}_1, \boldsymbol{\omega}_2)$ is a *primal dual optimal solution* if and only if

$$\begin{aligned} & \boldsymbol{\chi} := \begin{bmatrix} \mathbf{0}_N & \boldsymbol{X}^* \\ 1 & \mathbf{0}_{M_{max}+1+p}^T \end{bmatrix} \begin{bmatrix} z \\ \boldsymbol{\beta}^* \end{bmatrix} + \begin{bmatrix} -\boldsymbol{\eta} \\ 0 \end{bmatrix}, \\ & \boldsymbol{\tau} := \begin{bmatrix} \mathbf{0}_{M_{max}+1} & \boldsymbol{L}^* \\ 0 & \mathbf{0}_{M_{max}+1+p}^T \end{bmatrix} \begin{bmatrix} z \\ \boldsymbol{\beta}^* \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{M_{max}+1} \\ \sqrt{\tilde{M}} \end{bmatrix}, \\ & \begin{bmatrix} \mathbf{0}_N^T & 1 \\ \boldsymbol{X}^{*T} & \mathbf{0}_{M_{max}+1+p}^T \end{bmatrix} \boldsymbol{\omega}_1 + \begin{bmatrix} \mathbf{0}_{M_{max}+1}^T & 0 \\ \boldsymbol{L}^{*T} & \mathbf{0}_{M_{max}+1+p} \end{bmatrix} \boldsymbol{\omega}_2 = \begin{bmatrix} 1 \\ \mathbf{0}_{M_{max}+1+p} \end{bmatrix}, \\ & \boldsymbol{\omega}_1^T \boldsymbol{\chi} = 0,\ \ \boldsymbol{\omega}_2^T \boldsymbol{\eta} = 0, \\ & \boldsymbol{\omega}_1 \in L^{N+1},\ \ \boldsymbol{\omega}_2 \in L^{M_{max}+2}, \\ & \boldsymbol{\chi} \in L^{N+1},\ \ \boldsymbol{\tau} \in L^{M_{max}+2}. \end{aligned} \tag{3.52}$$

The parametrical upper bound $\tilde{M}$ in a constraint of the CQP and the penalty parameter λ in the PRSS are related. We can determine λ via Tikhonov regularization.

According to a large (finite) number of parameter values, this regularization method utilises an efficiency curve that comes from a plotting of the optimal solutions to problem (3.18). There are two axes which can be plotted in a coordinate scheme. At one axis, the complexity is denoted, whereas the other axis stands for the length of the residual vector (or goodness-of-fit). In this method, there is an L-curve under logarithmical scales employed and with some "kink" (corner) kind of a point on the efficiency boundary that has a more pronounced L shape now. This point is regarded to be the closest one to the origin and it is therefore often chosen, together with the corresponding penalty parameter [3].

In the following, we shall focus on the nonlinear part of GPLM and approach it by the help of CMARS. 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.

In the numerical example of this thesis, we restricted ourselves on the regularization of nonparametric part and we explained it with details in the Section 3.4.

3.4 Numerical Examples on the Use of CMARS

In this part we use two continuous data sets; one has interaction and the other has no interaction between variables. We apply both Tikhonov Regularization and Conic Quadratic Programming to predict the response variable. By this, we aim to view how CMARS or Tikhonov Regularization estimates the response variable. Data without interaction has three variables and 25 observations (taken from Mendenhal and Sincich (1994) [59], p. 678) while data within interaction contains five predictor variables and contains 32 observations (taken from Myers and Montgomery (2002) [68], p. 71). We find basis functions by MARS and applied Conic Quadratic Programming with MOSEK to estimate the unknown regression coefficients.

Here, $\boldsymbol{x}$ is a *generic* variable in the space of $\mathbb{R}^l$ ($\{l \in 1, 2, 3\}$) for the data without interaction and in the space of $\mathbb{R}^l$ ($l \in \{1, 2, \ldots, 5\}$) for the data within interaction.

3.4.1 Example without Interaction Data

In order to build the MARS model by trial and error we set the maximum number of BFs to four, i.e., $M_{max} = 4$, with no interaction. Then the number of maximum basis functions which are constructed by using MARS version 2 developed by Salford Systems are as follows:

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

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

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

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

Here, ψ_1 and ψ_2 are the standard BF and mirror image (reflected) BF for the predictor variable x_1. Let us note that the knot point for ψ_1 and for ψ_2 is 14.11. Similarly, ψ_3 and ψ_4 are the standard BF and mirror image (reflected) BF for the predictor variable x_1. Let us note that the knot point for ψ_3 and for ψ_4 is 12.01.

To prevent our optimization problem from nondifferentiability, we choose the knot values very close to the input values of the data point. Below we select knot values for corresponding BFs:

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

For ψ_3: $\tau_{25,1} = 12.01$, $\widetilde{\tau}_{25,1} = 12.00$, which is not equal to $\tau_{25,1} = 12.01$, but very close to it.

For ψ_4: $\tau_{25,1} = 12.01$, $\widetilde{\tau}_{25,1} = 12.00$, which is not equal to $\tau_{25,1} = 12.01$ but very close to it.

Then, the BFs of the (3.7) 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

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

As a result, the large model becomes

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

Next, we can write the PRSS objective function in (3.9) as follows:

$$\begin{aligned}
PRSS &= \sum_{i=1}^{25} \left(y_i - \boldsymbol{\theta}^T \boldsymbol{\psi}(\tilde{\boldsymbol{d}}_i) \right)^2 + \sum_{m=1}^{4} \lambda_m \sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s \in V_m}} \int \theta_m^2 \left[D_{r,s}^{\boldsymbol{\alpha}} \psi_m(\boldsymbol{t}^m) \right]^2 d\boldsymbol{t}^m \\
&= \sum_{i=1}^{25} \left(y_i - \boldsymbol{\theta}^T \boldsymbol{\psi}(\tilde{\boldsymbol{d}}_i) \right)^2 + \lambda_1 \left(\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s \in V_1}} \int \theta_1^2 \left[D_{r,s}^{\boldsymbol{\alpha}} \psi_1(\boldsymbol{t}^1) \right]^2 d\boldsymbol{t}^1 \right) \\
&\quad + \lambda_2 \left(\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s \in V_2}} \int \theta_2^2 \left[D_{r,s}^{\boldsymbol{\alpha}} \psi_2(\boldsymbol{t}^2) \right]^2 d\boldsymbol{t}^2 \right) \\
&\quad + \lambda_3 \left(\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s \in V_3}} \int \theta_3^2 \left[D_{r,s}^{\boldsymbol{\alpha}} \psi_3(\boldsymbol{t}^3) \right]^2 d\boldsymbol{t}^3 \right) \\
&\quad + \lambda_4 \left(\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s \in V_4}} \int \theta_4^2 \left[D_{r,s}^{\boldsymbol{\alpha}} \psi_4(\boldsymbol{t}^4) \right]^2 d\boldsymbol{t}^4 \right).
\end{aligned}$$

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

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

Besides, the corresponding derivatives for the BFs $D_{r,s}^{\boldsymbol{\alpha}}\psi_m(\boldsymbol{t}^m)$ (for $m = 1, \ldots, 4$) are stated below. For the BF $\psi_1(\boldsymbol{t}^1)$, there is no interaction, so: $r = s = 1$. 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

$$\begin{aligned} |\,\boldsymbol{\alpha}| = 1: \qquad & D_1^1\psi_1(\boldsymbol{t}^1) := \frac{\partial\psi_1}{\partial t_1^1}(\boldsymbol{t}^1) = \frac{\partial\psi_1}{\partial x_1}(x_1) = \begin{cases} 1, & \text{if } x_1 > 14.11, \\ 0, & \text{otherwise;} \end{cases} \\ |\,\boldsymbol{\alpha}| = 2: \qquad & D_1^2\psi_1(\boldsymbol{t}^1) := \frac{\partial^2\psi_1}{\partial t_1^1 \partial t_1^1}(\boldsymbol{t}^1) = \frac{\partial^2\psi_1}{\partial x_1 \partial x_1}(x_1) = 0 \ \text{ for all } \ x_1. \end{aligned}$$

For the BF $\psi_2(\boldsymbol{t}^2)$, there is no interaction, so: $r = s = 1$. 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

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

For the BF $\psi_3(\boldsymbol{t}^3)$, there is no interaction, 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

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

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

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

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

If $\lambda_1 = \lambda_2 = ... = \lambda_{M_{max}} =: \lambda$, our problem becomes a *Tikhonov regularization* problem. In fact, the PRSS function to be minimized becomes approximated in the following way:

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

The first part of the PRSS objective function and that of the Tikhonov regularization problem are equal to each other. But the second part is equal in an approximative sense:

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

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

The following values represent RSS by its parts, as we list them below. The whole RSS with a tabular form can be seen in Appendix A.

$$\begin{aligned}\sum_{i=1}^{25}\left(y_i - \boldsymbol{\theta}^T\boldsymbol{\psi}(\tilde{\boldsymbol{d}}_i)\right)^2 = (13.6 - \theta_0 - (\max\{0, 14.1 - 14.11\})\theta_1 - \\ (\max\{0, 14.11 - 14.1\})\theta_2 - \\ (\max\{0, 14.1 - 12.01\})\theta_3 - \\ (\max\{0, 12.01 - 14.1\})\theta_4)^2 + \\ (16.6 - \theta_0 - (\max\{0, 16 - 14.11\})\theta_1 - \\ (\max\{0, 14.11 - 16\})\theta_2 - \\ (\max\{0, 16 - 12.01\})\theta_3 - \\ (\max\{0, 12.01 - 16\})\theta_4)^2 + \\ \vdots \\ (14.9 - \theta_0 - (\max\{0, 12 - 14.11\})\theta_1 - \\ (\max\{0, 14.11 - 12\})\theta_2 - \\ (\max\{0, 12 - 12.01\})\theta_3 - \\ (\max\{0, 12.01 - 12\})\theta_4)^2.\end{aligned}$$

When the maximum functions are computed, the RSS looks as follows:

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

and writing into vector notation gives us the following form:

$$= (13.6 - \theta_0 - 0.01\theta_2 - 2.9\theta_3)^T(13.6 - \theta_0 - 0.01\theta_2 - 2.9\theta_3) +$$

$$(16.6-\theta_0-1.89\theta_1-3.99\theta_3)^T(16.6-\theta_0-1.89\theta_1-3.99\theta_3)+$$
$$\vdots$$
$$(14.9-\theta_0-2.11\theta_2-0.01\theta_4)^T(14.9-\theta_0-2.11\theta_2-0.01\theta_4).$$

If we turn the above summation into matrix notation, we get the subsequent representation. So, the value of the first part of PRSS, which is RSS, has been found:

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

By discretizing, the multi-dimensional integral in the second part of equation (3.12) takes the form of (3.15). The discretized form is denoted by $\boldsymbol{L}$ and at the end, the formulation as given in equation (3.18) can be obtained.

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

$$\begin{aligned}L_1 &= \sum_{i=1}^{(N+1)^{K_1}}\left[\left(\sum_{\substack{|\boldsymbol{\alpha}|=1\\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2}\sum_{\substack{r<s\\ r,s\in V_1}}\left[D_{r,s}^{\boldsymbol{\alpha}}\left(\max\{0,x_1-14.11\}\right)\right]^2\right)\left(\tilde{x}_{l^{\kappa_1^1}_{\sigma^{\kappa_1}}+1,\kappa_1^1}-\tilde{x}_{l^{\kappa_1^1}_{\sigma^{\kappa_1}},\kappa_1^1}\right)\right],\\ &= 3.9243,\\ L_2 &= \sum_{i=1}^{(N+1)^{K_2}}\left[\left(\sum_{\substack{|\boldsymbol{\alpha}|=1\\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2}\sum_{\substack{r<s\\ r,s\in V_2}}\left[D_{r,s}^{\boldsymbol{\alpha}}\left(\max\{0,14.11-x_1\}\right)\right]^2\right)\left(\tilde{x}_{l^{\kappa_1^2}_{\sigma^{\kappa_1}}+1,\kappa_1^2}-\tilde{x}_{l^{\kappa_1^2}_{\sigma^{\kappa_1}},\kappa_1^2}\right)\right],\\ &= 3.6878,\\ L_3 &= \sum_{i=1}^{(N+1)^{K_3}}\left[\left(\sum_{\substack{|\boldsymbol{\alpha}|=1\\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2}\sum_{\substack{r<s\\ r,s\in V_3}}\left[D_{r,s}^{\boldsymbol{\alpha}}\left(\max\{0,x_1-12.01\}\right)\right]^2\right)\left(\tilde{x}_{l^{\kappa_1^3}_{\sigma^{\kappa_1}}+1,\kappa_1^3}-\tilde{x}_{l^{\kappa_1^3}_{\sigma^{\kappa_1}},\kappa_1^3}\right)\right],\\ &= 4.1833,\\ L_4 &= \sum_{i=1}^{(N+1)^{K_4}}\left[\left(\sum_{\substack{|\boldsymbol{\alpha}|=1\\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2}\sum_{\substack{r<s\\ r,s\in V_4}}\left[D_{r,s}^{\boldsymbol{\alpha}}\left(\max\{0,12.01-x_1\}\right)\right]^2\right)\left(\tilde{x}_{l^{\kappa_1^4}_{\sigma^{\kappa_1}}+1,\kappa_1^4}-\tilde{x}_{l^{\kappa_1^4}_{\sigma^{\kappa_1}},\kappa_1^4}\right)\right],\\ &= 3.3912.\end{aligned}$$

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

Note that the first column elements of $\boldsymbol{L}$ are all zero and the diagonal elements of this matrix are $\boldsymbol{L}_m$ $(m = 1, 2, 3, 4)$ as introduced above.

In the equation (3.18), $\|\boldsymbol{L\theta}\|_2^2$ is the squared norm of $\boldsymbol{L\theta}$ which is:

$$\begin{aligned} \|\boldsymbol{L\theta}\|_2^2 &= (\theta_1 \cdot (3.9243))^2 + (\theta_2 \cdot (3.6878))^2 + (\theta_3 \cdot (4.1833))^2 + \\ & \quad (\theta_4 \cdot (3.3912))^2 . \end{aligned} \tag{3.53}$$

From the equations (3.16) and (3.44), we can calculate the objective function PRSS for the numerical example. As we mentioned before, PRSS is a Tikhonov Regularization Problem. To solve this problem, we can reformulate PRSS as a CQP problem as follows:

$$\begin{aligned} & \min_{t,\boldsymbol{\theta}} \quad t, \\ \text{subject to} \quad & \left\| \boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta} - \boldsymbol{y} \right\|_2 \leq t, \\ & \|\boldsymbol{L\theta}\|_2 \leq \sqrt{\tilde{M}}. \end{aligned}$$

PRSS and CQP have different notations, but they have the same solution for appropriate choice of the λ and $\sqrt{\tilde{M}}$. When decreasing the values of λ and $\sqrt{\tilde{M}}$ a bit, the minimum value of the $\|\boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta} - \boldsymbol{y}\|_2$ increases for both PRSS and CQP that are minimization problems.

Our previous CQP problem can be rewritten as follows:

$$\min_{t,\boldsymbol{\theta}} \quad t,$$

$$\begin{aligned} \text{subject to} \quad & 13.6 - \theta_0 - 0.01\theta_2 - 2.9\theta_3 = \theta_5, \\ & 16.6 - \theta_0 - 1.89\theta_1 - 3.99\theta_3 = \theta_6, \\ & 23.5 - \theta_0 - 15.77\theta_1 - 17.87\theta_3 = \theta_7, \\ & 10.20 - \theta_0 - 6.11\theta_2 - 4.01\theta_4 = \theta_8, \\ & 5.4 - \theta_0 - 10.01\theta_2 - 7.91\theta_4 = \theta_9, \\ & 15 - \theta_0 - 0.89\theta_1 - 2.99\theta_3 = \theta_{10}, \\ & 9 - \theta_0 - 5.31\theta_2 - 3.21\theta_4 = \theta_{11}, \\ & 12.3 - \theta_0 - 1.71\theta_2 - 0.39\theta_3 = \theta_{12}, \end{aligned}$$

$$
\begin{aligned}
16.3 - \theta_0 - 2.49\theta_1 - 4.59\theta_3 &= \theta_{13},\\
15.4 - \theta_0 - 0.79\theta_1 - 2.79\theta_3 &= \theta_{14},\\
13 - \theta_0 - 0.41\theta_2 - 1.69\theta_3 &= \theta_{15},\\
14.4 - \theta_0 - 0.99\theta_1 - 3.09\theta_3 &= \theta_{16},\\
10 - \theta_0 - 6.31\theta_2 - 4.21\theta_4 &= \theta_{17},\\
10.20 - \theta_0 - 2.71\theta_2 - 0.61\theta_4 &= \theta_{18},\\
9.5 - \theta_0 - 5.11\theta_2 - 3.01\theta_4 &= \theta_{19},\\
1.5 - \theta_0 - 13.11\theta_2 - 11.01\theta_4 &= \theta_{20},\\
18.5 - \theta_0 - 2.89\theta_1 - 4.99\theta_3 &= \theta_{21},\\
12.6 - \theta_0 - 1.31\theta_2 - 0.79\theta_3 &= \theta_{22},\\
17.5 - \theta_0 - 1.69\theta_1 - 3.79\theta_3 &= \theta_{23},\\
4.9 - \theta_0 - 9.61\theta_2 - 7.51\theta_4 &= \theta_{24},\\
15.9 - \theta_0 - 0.39\theta_1 - 2.49\theta_3 &= \theta_{25},\\
8.5 - \theta_0 - 6.81\theta_2 - 4.71\theta_4 &= \theta_{26},\\
10.6 - \theta_0 - 5.51\theta_2 - 3.41\theta_4 &= \theta_{27},\\
13.9 - \theta_0 - 1.09\theta_1 - 3.19\theta_3 &= \theta_{28},\\
14.9 - \theta_0 - 2.11\theta_2 - 0.01\theta_4 &= \theta_{29},\\
\left(\sum_{i=5}^{29} \theta_i^2\right)^{1/2} &\leq t,\\
\left(\sum_{i=30}^{34} \theta_i^2\right)^{1/2} &\leq \sqrt{\widetilde{M}},
\end{aligned}
$$

where $\theta_{30} = 0\theta_1$, $\theta_{31} = 3.9243\theta_1$, $\theta_{32} = 3.6878\theta_2$, $\theta_{33} = 4.1833\theta_3$, $\theta_{34} = 3.3912\theta_4$.

As can be seen from the equation (3.18), our problem involves 25 linear constraints and two quadratic cones. For solving our numerical problem, we transform it into the *MOSEK* format. For this transformation, we introduce new unknown variables (θ_5,...,θ_{34}), to the linear notations in these two quadratic cones. Therefore, the notations in the cones are simplified and we write them as constraints. MOSEK uses an interior-point optimizer to solve the considered CQP problem. It is a well-recognized implementation of the homogeneous and self-dual algorithm. We use model-free (train and error) method for different $\sqrt{\widetilde{M}}$ values in our example. By using different $\sqrt{\widetilde{M}}$

values when solving our CMARS model in MOSEK, we reach several solutions that are based on four BFs.

3.4.2 Example with Interaction Data

While implementing the CMARS algorithm, first, the MARS model is built by using the Salford MARS v.2 [60]. In the construction of the model, the maximum number of BFs (M_{max}) and the highest degree of interactions are determined by trial and error. In this example, M_{max} and the highest degree of interactions are assigned to be five and two, respectively. As a result 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 BF and mirror image (reflected) BF for the predictor variable x_2. Let us note that the knot point for ψ_1 and for ψ_2 is 2.21. BF ψ_4, on the other hand, uses the function ψ_3 to express the interaction between the predictor variables x_1 and x_4. Similarly, ψ_5 represents the interaction between the predictor variables x_4 and x_5.

To prevent our optimization problem from nondifferentiability, we choose the knot values very close to the input values of the data point. Below we select knot values for corresponding BFs:

For ψ_1:

$\tau_{18,2} = 2.21$, $\widetilde{\tau}_{18,2} = 2.20$, which is not equal to $\tau_{18,2} = 2.21$, but very close to it.

For ψ_2:

$\tau_{18,2} = 2.21$, $\widetilde{\tau}_{18,2} = 2.20$, which is not equal to $\tau_{18,2} = 2.21$, but very close to it.

For ψ_3:

$\tau_{1,4} = 0.26$, $\widetilde{\tau}_{1,4} = 0.25$, which is not equal to $\tau_{1,4} = 0.26$, but very close to it.

For ψ_4:

$\tau_{6,1} = 1601$, $\widetilde{\tau}_{6,1} = 1600$, which is not equal to $\tau_{6,1} = 1601$, but very close to it.
$\tau_{1,4} = 0.26$, $\widetilde{\tau}_{1,4} = 0.25$, which is not equal to $\tau_{1,4} = 0.26$, but very close to it.

For ψ_5:

$\tau_{25,5} = 0.71$, $\widetilde{\tau}_{25,5} = 0.70$, which is not equal to $\tau_{25,5} = 0.71$, but very close to it.
$\tau_{6,4} = 0.26$, $\widetilde{\tau}_{6,4} = 0.25$, which is not equal to $\tau_{6,4} = 0.26$, but very close to it.

Then, the BFs of the (3.7) 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

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

As a result, the large model becomes

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

Next, we can write the PRSS objective function in (3.9) 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:

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

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

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 $\psi_4(\boldsymbol{t}^4)$, on the other hand, there is an interaction between the predictors x_1 and x_4, and $r = 1$ 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 $\psi_5(\boldsymbol{t}^5)$, on the other hand, there is 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

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

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

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

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

$$\begin{aligned} PRSS &= \sum_{i=1}^{32}\left(y_i - \boldsymbol{\theta}^T\boldsymbol{\psi}(\tilde{\boldsymbol{d}}_i)\right)^2 \\ &+ \sum_{m=1}^{5}\lambda_m \sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_m}} \int \theta_m^2 \left[D_{r,s}^{\boldsymbol{\alpha}}\psi_m(\boldsymbol{t}^m)\right]^2 d\boldsymbol{t}^m. \end{aligned}$$

If $\lambda_1 = \lambda_2 = ... = \lambda_{M_{max}} =: \lambda$, then the namely by a *Tikhonov regularization* problem: PRSS function to be minimized becomes approximated is as follows:

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

The first part of the PRSS objective function and that of the Tikhonov regularization problem are equal to each other. But the second part is equal in an approximative sense:

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

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

The following values represent RSS by its parts, as we list them below. The whole RSS with a tabular form can be seen in Appendix A.

$$\begin{aligned} \sum_{i=1}^{32}\left(y_i - \boldsymbol{\theta}^T\boldsymbol{\psi}(\tilde{\boldsymbol{d}}_i)\right)^2 = (0.013 - \theta_0 - (\max\{0, 0.58 - 2.21\})\theta_1 - \\ (\max\{0, 2.21 - 0.58\})\theta_2 - \\ (\max\{0, 0.25 - 0.26\})\theta_3 - \\ (\max\{0, 1650 - 1601\})(\max\{0, 0.25 - 0.26\})\theta_4 - \\ (\max\{0, 0.9 - 0.71\})(\max\{0, 0.25 - 0.26\})\theta_5)^2 + \\ (0.016 - \theta_0 - (\max\{0, 0.66 - 2.21\})\theta_1 - \\ (\max\{0, 2.21 - 0.66\})\theta_2 - \\ (\max\{0, 0.33 - 0.26\})\theta_3 - \\ (\max\{0, 1650 - 1601\})(\max\{0, 0.33 - 0.26\})\theta_4 - \end{aligned}$$

$$(\max\{0, 0.9 - 0.71\})(\max\{0, 0.33 - 0.26\})\theta_5)^2 +$$

$$\vdots$$

$$(0.068 - \theta_0 - (\max\{0, 18.5 - 2.21\})\theta_1 -$$

$$(\max\{0, 2.21 - 18.5\})\theta_2 -$$

$$(\max\{0, 1.5 - 0.26\})\theta_3 -$$

$$(\max\{0, 1700 - 1601\})(\max\{0, 1.5 - 0.26\})\theta_4 -$$

$$(\max\{0, 0.7 - 0.71\})(\max\{0, 1.5 - 0.26\})\theta_5)^2.$$

When the maximum functions are computed, the RSS terms looks as follows:

$$\sum_{i=1}^{32} \left(y_i - \boldsymbol{\theta}^T \boldsymbol{\psi}(\tilde{\boldsymbol{d}}_i) \right)^2 = (0.013 - \theta_0 - 1.63\theta_2)^2 +$$

$$(0.016 - \theta_0 - 1.55\theta_2 - 0.07\theta_3 - 3.43\theta_4 - 0.0133\theta_5)^2 +$$

$$\vdots$$

$$(0.068 - \theta_0 - 16.29\theta_1 - 1.24\theta_3 - 122,76\theta_4)^2.$$

and writing into vector notation gives us the following form:

$$= (0.013 - \theta_0 - 1.63\theta_2)^T (0.013 - \theta_0 - 1.63\theta_2) +$$

$$(0.016 - \theta_0 - 1.55\theta_2 - 0.07\theta_3 - 3.43\theta_4 - 0.0133\theta_5)^T$$

$$(0.016 - \theta_0 - 1.55\theta_2 - 0.07\theta_3 - 3.43\theta_4 - 0.0133\theta_5) +$$

$$\vdots$$

$$(0.068 - \theta_0 - 16.29\theta_1 - 1.24\theta_3 - 122,76\theta_4)^T (0.068 - \theta_0 - 16.29\theta_1 - 1.24\theta_3 - 122,76\theta_4).$$

If we turn the above summation into matrix notation, we get the subsequent representation. So, the value of the first part of PRSS, which is RSS, has been found:

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

By discretizing, the multi-dimensional integral in the second part of equation (3.12) takes the form of (3.15). The discretized form is denoted by $\boldsymbol{L}$ and at the end, the formulation as given in equation (3.18) can be obtained.

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

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

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

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

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

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

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

Note that the first column elements of $\boldsymbol{L}$ are all zero and the diagonal elements of this matrix are L_m $(m = 1, 2, ..., 5)$ as introduced above.

In the equation (5.25), $\|\boldsymbol{L\theta}\|_2^2$ is the squared norm of $\boldsymbol{L\theta}$ which is:

$$\begin{aligned}\|\boldsymbol{L\theta}\|_2^2 &= (\theta_1 \cdot (3.9497))^2 + (\theta_2 \cdot (1.5875))^2 + (\theta_3 \cdot (1.1958))^2 + (\theta_4 \cdot (9.9015))^2 + \\ &\quad (\theta_5 \cdot (0.1975))^2 . \end{aligned} \tag{3.54}$$

We can calculate the objective function PRSS for the numerical example from the equations (3.16) and (3.44). As we mentioned before, PRSS is the Tikhonov Regularization Problem. To solve this problem, we can reformulate PRSS as a CQP problem as follows:

$$\begin{aligned} &\min_{t,\boldsymbol{\theta}} \quad t, \\ \text{subject to} \quad &\left\|\boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta} - \boldsymbol{y}\right\|_2 \leq t, \\ &\|\boldsymbol{L\theta}\|_2 \leq \sqrt{\tilde{M}}, \end{aligned} \tag{3.55}$$

PRSS and CQP have different notations, but they have the same solution for appropriate choice of the parameter λ and $\sqrt{\tilde{M}}$. When decreasing the values of λ and $\sqrt{\tilde{M}}$ a bit, the minimum value of $\|\boldsymbol{\psi}(\tilde{\boldsymbol{d}})\boldsymbol{\theta} - \boldsymbol{y}\|_2$ increases for both PRSS and CQP that are minimization problems.

Our previous CQP problem can be rewritten as follows:

$$\begin{aligned} &\min_{t,\boldsymbol{\theta}} \quad t, \\ \text{subject to} \quad &0.013 - \theta_0 - 1.63\theta_2 = \theta_6, \\ &0.016 - \theta_0 - 1.55\theta_2 - 0.07\theta_3 - 3.43\theta_4 - 0.0133\theta_5 = \theta_7, \\ &0.015 - \theta_0 - 1.55\theta_2 - 0.07\theta_3 - 3.43\theta_4 - 0.0133\theta_5 = \theta_8, \\ &0.016 - \theta_0 - 1.55\theta_2 - 0.07\theta_3 - 3.43\theta_4 - 0.0168\theta_5 = \theta_9, \\ &0.015 - \theta_0 - 1.55\theta_2 - 0.07\theta_3 - 0.0203\theta_5 = \theta_{10}, \\ &0.016 - \theta_0 - 1.55\theta_2 - 0.07\theta_3 - 0.0203\theta_5 = \theta_{11}, \\ &0.014 - \theta_0 - 1.21\theta_2 - 0.24\theta_3 - 11,76\theta_4 - 0.0216\theta_5 = \theta_{12}, \\ &0.021 - \theta_0 - 1.04\theta_2 - 0.32\theta_3 - 15,68\theta_4 - 0.0288\theta_5 = \theta_{13}, \\ &0.018 - \theta_0 - 1.04\theta_2 - 0.32\theta_3 - 15,68\theta_4 - 0.0288\theta_5 = \theta_{14}, \\ &0.019 - \theta_0 - 1.04\theta_2 - 0.32\theta_3 - 15,68\theta_4 - 0.0288\theta_5 = \theta_{15}, \\ &0.021 - \theta_0 - 1.04\theta_2 - 0.32\theta_3 - 15,68\theta_4 - 0.0608\theta_5 = \theta_{16}, \end{aligned}$$

$$
\begin{aligned}
0.019 - \theta_0 - 1.04\theta_2 - 0.32\theta_3 - 15,68\theta_4 - 0.0608\theta_5 &= \theta_{17},\\
0.021 - \theta_0 - 1.04\theta_2 - 0.32\theta_3 - 15,68\theta_4 - 0.0608\theta_5 &= \theta_{18},\\
0.025 - \theta_0 - 1.01\theta_2 - 0.84\theta_3 - 41,16\theta_4 - 0.0756\theta_5 &= \theta_{19},\\
0.025 - \theta_0 - 0.21\theta_2 - 0.74\theta_3 - 36,26\theta_4 - 0.0666\theta_5 &= \theta_{20},\\
0.026 - \theta_0 - 0.21\theta_2 - 0.84\theta_3 - 41,16\theta_4 - 0.0756\theta_5 &= \theta_{21},\\
0.024 - \theta_0 - 0.01\theta_2 - 0.84\theta_3 - 41,16\theta_4 - 0.0756\theta_5 &= \theta_{22},\\
0.025 - \theta_0 - 0.01\theta_2 - 0.84\theta_3 - 41,16\theta_4 - 0.0756\theta_5 &= \theta_{23},\\
0.024 - \theta_0 - 0.01\theta_2 - 0.84\theta_3 - 41,16\theta_4 - 0.0756\theta_5 &= \theta_{24},\\
0.025 - \theta_0 - 0.01\theta_2 - 0.84\theta_3 - 41,16\theta_4 - 0.1596\theta_5 &= \theta_{25},\\
0.027 - \theta_0 - 0.01\theta_2 - 0.84\theta_3 - 41,16\theta_4 - 0.1596\theta_5 &= \theta_{26},\\
0.026 - \theta_0 - 0.01\theta_2 - 1.24\theta_3 - 60,76\theta_4 - 0.2356\theta_5 &= \theta_{27},\\
0.029 - \theta_0 - 0.79\theta_1 - 1.24\theta_3 - 60,76\theta_4 - 0.1116\theta_5 &= \theta_{28},\\
0.03 - \theta_0 - 0.79\theta_1 - 1.24\theta_3 - 60,76\theta_4 &= \theta_{29},\\
0.028 - \theta_0 - 0.79\theta_1 - 1.24\theta_3 - 60,76\theta_4 - 0.0496\theta_5 &= \theta_{30},\\
0.032 - \theta_0 - 0.79\theta_1 - 1.4\theta_3 - 68,6\theta_4 - 0.196\theta_5 &= \theta_{31},\\
0.033 - \theta_0 - 1.12\theta_1 - 1.24\theta_3 - 60,76\theta_4 - 0.1116\theta_5 &= \theta_{32},\\
0.039 - \theta_0 - 1.79\theta_1 - 1.24\theta_3 - 122,76\theta_4 &= \theta_{33},\\
0.04 - \theta_0 - 1.79\theta_1 - 1.24\theta_3 - 60,76\theta_4 &= \theta_{34},\\
0.035 - \theta_0 - 1.79\theta_1 - 1.24\theta_3 - 60,76\theta_4 - 0.1736\theta_5 &= \theta_{35},\\
0.056 - \theta_0 - 10.29\theta_1 - 1.24\theta_3 - 122,76\theta_4 &= \theta_{36},\\
0.068 - \theta_0 - 16.29\theta_1 - 1.24\theta_3 - 122,76\theta_4 &= \theta_{37},\\
\left(\sum_{i=6}^{37}\theta_i^2\right)^{1/2} &\leq t,\\
\left(\sum_{i=38}^{43}\theta_i^2\right)^{1/2} &\leq \sqrt{\tilde{M}},
\end{aligned}
$$

where $\theta_{38} = 0\theta_1$, $\theta_{39} = 3.9497\theta_1$, $\theta_{40} = 1.5875\theta_2$, $\theta_{41} = 1.1958\theta_3$, $\theta_{42} = 9.9015\theta_4$, $\theta_{43} = 0.1975\theta_5$. As can be seen from the equation (3.18), our problem involves 32 linear constraints and two quadratic cones. For solving our numerical problem, we

transform it into the *MOSEK* format. For this transformation, we introduce new unknown variables (θ_6,...,θ_{43}), to the linear notations in these two quadratic cones. Therefore, the notations in the cones are simplified and we write them as constraints. MOSEK uses an interior-point optimizer to solve CQP problem. It is a well-recognized implementation of the homogeneous and self-dual algorithm. We use model-free (train and error) method for different $\sqrt{\tilde{M}}$ values in our example. By using different $\sqrt{\tilde{M}}$ values when solving our CMARS model in MOSEK, we reach several solutions that are based on five BFs.

In our optimization problem, the values $\sqrt{\tilde{M}}$ can be regarded as a *model-free method*. We note that our family of optimization problems, indexed by $\tilde{M}$, can be considered as a problem of *parametric programming*. If the $\sqrt{\tilde{M}}$ values are accessed in our CMARS code, CMARS provides us several solutions, each of them based on 5 BFs.

In the next section, we apply CMARS to different sizes and types of data sets. The results obtained from the algorithms with solving Conic Quadratic Programming for the "without interaction" and also "with interaction" data sets, are also compared with Tikhonov Regularization Problem whose results are obtained from the thesis [49], which is on progress, according to various different general performance comparison criteria.

3.5 Validation Approach and Comparison Measures

3.5.1 Introduction

In our applications, to compare the Tikhonov Regularization Problem, whose results are obtained from the thesis [49], which is on progress, with Conic Quadratic Programming methods, we use two different data sets that are "continuous" (real-valued). The first data set has no interaction and the other has interaction between variables. We wanted to see how CQP or Tikhonov estimates the response variable.

For the comparison, the *Linear Regression Models (LRMs)* are also developed for no interaction and interaction training and test data sets by using the stepwise regression algorithm [74]. While developing LRMs, assumptions related to *Least Square Error*

(LSE) are all tested. If any one is not validated, corrective measures such as transformation of the response or predictor(s) are taken. In addition to LRMs, MARS models are built by using Salford MARS software program [60]. Then, CQP models are constructed by running the MATLAB code developed by the authors [62] and MOSEK code developed by the authors [95].

In order to evaluate the CQP and Tikhonov methods' performance, we employed several measures [106]. The performance measures we employed in our applications and their general notations are as follows;

General Notation

- 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 $(\boldsymbol{X}) = p$ $(p \leq N)$.

3.5.2 Comparison Measures

i. r

This value is a correlation coefficient that is a measure of how well linear association between the actual and the predicted response values [97]. The formula is

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

where y is the actual response variables, $\hat{y}$ is the predicted response variables and $\bar{y}$ is the mean of actual values. Here, $s(y)$ is the standard deviations of actual and $s(\hat{y})$ is the standard deviations of predicted response variable. If r closes to -1, there is a strong but negative relationship; and if r closes to 1, there is a strong positive relationship between the actual and the predicted response variables. The degree of relationship decreases as it approaches zero [97].

ii. Prediction Error Sum of Squares (PRESS)

PRESS shows that predictive ability of our model. It is actually the sum of squares of the prediction error. The smaller the *PRESS* the better it is [97]. The formula is

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

iii. $\boldsymbol{R^2}$

R^2 is a coefficient of determination that provides a measure of how well future outcomes are likely to be predicted by the model. The higher the R^2, the better the model fits your data [97]. Its formula is

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

iv. **Adjusted** $\boldsymbol{R^2}$

This value is a modification of R^2 that adjusts for the number of explanatory terms in a model. Unlike R^2, the *Adjusted* R^2 increases only if the new term improves the model more than would be expected by chance. So, it is useful for comparing models with different numbers of predictors. The higher the *Adjusted* R^2, the better the model fits your data [97]. 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** $\boldsymbol{R^2}$

The *predicted*R^2 shows that how well the model predicts responses for our new observations. The higher *Predicted* R^2 value suggests that our model has a greater

predictive ability. The higher the *Predicted* R^2, the better it is [97]. Its formula is

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

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

MSE of an estimator is one of many ways to quantify the difference between an estimator and the true value of the quantity being estimated. The smaller MSE, the better it is [97]. The formula is

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

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

RMSE is a frequently-used measure of the differences between values predicted by a model or an estimator and the values actually observed from the thing being modeled or estimated. RMSE is a good measure of precision. The smaller RMSE, the better it is [97]. 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)**

This value AAE measures the average magnitude of error. The smaller AAE, the better it is [97]. The formula is

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

***ix.* Average Absolute Percentage Error (AAPE)**

$AAPE$ measures the scale relative error. The smaller AAPE, the better it is [97]. The formula is

$$AAPE := \frac{100}{N} \sum_{i=1}^{N} \left| \frac{y_i - \hat{y}_i}{y_i} \right|.$$

3.6 Numerical Results of the Conic Quadratic Problems

We used Matlab programming language (Matlab R2007a) and MOSEK which is introduces in Subsection 3.1.1 for our CQP method, in our numerical examples.

At first, we employ the command *Generalized Singular Value Decomposition* ($GSVD$) with the matrices BF and L as inputs of this command, in our application. After that, we use MOSEK program to get the unknown regression coefficients. The M value, which is our regularization parameter, plays a key role in the estimation of coefficients and it should be chosen with care. The L-curve criterion can be used to decide the M value. We choose the value which corresponds to the *corner* of the L-curve, the point with maximum curvature.

In this study, we run the program many times, each time with a different M values ($M > 0$), and observe how the result changes. Then, we calculated the RSS and $\|\boldsymbol{L\theta}\|_2$ values, for each solution. Therefore, the range of M can be decided, where its end points (the first value and the last value) are stabilized.

In this thesis, we examined two different data sets which are with and without interaction, respectively. We compared results of the data sets and saw which data sets have better results when CQP is employed. While doing comparisons, we use some statistical tools, such as: ${R^2}_{adj}$, r, $RMSE$ and AAE. The results are illustrated in the following Table 3.1:

Table 3.1: Conic Quadratic Programming for the two data sets

	No Interaction			Interaction		
Measure	M_{first}	M_{corner}	M_{last}	M_{first}	M_{corner}	M_{last}
AAE	3.6566	0.9581	0.9571	0.0080	0.0015	0.0014
${R^2}_{adj}$	-0.2631	0.9258	0.9264	-0.2399	0.9602	0.9663
$RMSE$	5.3269	1.2910	1.2861	0.0132	0.0024	0.0022
r	0.8635	0.9703	0.9704	0.9346	0.9839	0.9863

From Table 3.1 we can easily observe that CQP gives better solutions for the data set with interaction than for the ones with no interaction for different M values. For all three points from with interaction data, the AAE and $RMSE$ results are smaller than from without interaction data. Lower AAE and $RMSE$ indicates that CQP solution provides more accurate results for data sets with interaction. Also, r for the data with interaction is closer to 1 than with use of the data without interaction. This means that there is a better linear association between the actual and the predicted response values at the data with interaction for all three values: M_{first}, M_{corner} and M_{last} : $M_{first} < M_{corner} < M_{last}$.

For both data sets, M_{last} gives better results than M_{first} and M_{corner}. Since both data set have higher R^2_{adj} criteria values, which shows that the model fit is better at M_{last}; lower AAE and $RMSE$ indicate that the CQP solution provides more accurate results for data sets at the point M_{last} and r closer to 1. This shows us that there is a better linear association between the actual and the predicted response values at M_{last} for the with and the without interaction data sets. We can obviously see that as the M value decreases, the error rates AAE and $RMSE$ gets bigger, while R^2_{adj} decreases. This means that our model gets worse to explain the variation in the response variable.

Moreover, we get the following L-curves, obtained by plotting values of RSS versus $\|\boldsymbol{L\theta}\|_2$, for the two data sets :

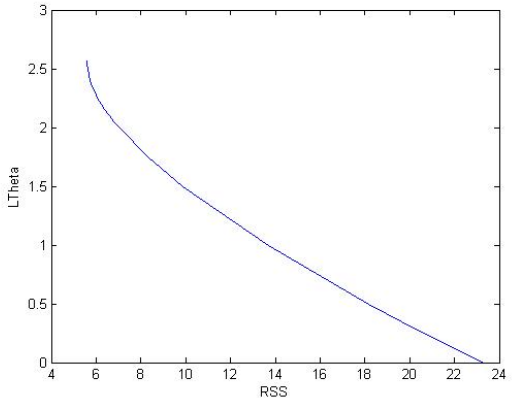

Figure 3.1: L-curve, RSS vs. norm of $\boldsymbol{L\theta}$ for the data with no interaction.

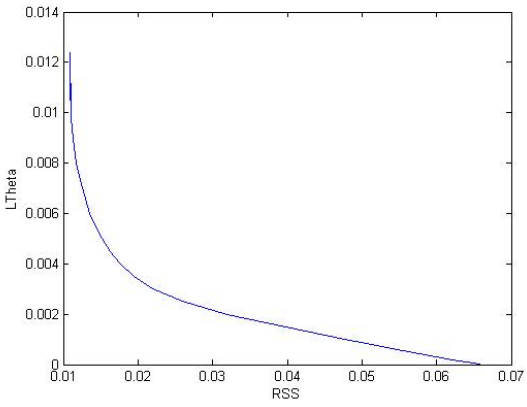

Figure 3.2: L-curve, *RSS* vs. norm of $\boldsymbol{L\theta}$ for the data with interaction.

In the next section, we will compare the CQP results with Tikhonov Regularization results obtained from the thesis [49], which is on progress, on the two data sets with interaction and without interaction, for the end points and corner points.

3.7 Comparison of the Results for the Two Methods (Tikhonov Regularization and CQP)

3.7.1 Data with No Interaction

1. *At Initial Values* ($M_{first} - \lambda_{first}$)

The results for M_{first} (for CQP) and λ_{first} (for Tikhonov) are as follows:

Table 3.2: Comparison of CQP and Tikhonov Regularization ($M_{first} - \lambda_{first}$)

Measure	CQP	Tikhonov
AAE	3.6566	0.9571
$R^2{}_{adj}$	-0.2631	0.9264
$RMSE$	5.3269	1.2861
r	0.8635	0.9704

From Table 3.2, it is obviously seen that Tikhonov solver is better at the initial points according to all the statistical measures for this data. For Tikhonov regularization, a

higher ${R^2}_{adj}$ and a lower $RMSE$ indicate a good model fit and high accuracy rate, respectively. Besides, for both method, the r value is high meaning that there is a good linear relationship between the actual and predicted values. However, Tikhonov is a bit much better.

2. *At Last Values ($M_{last} - \lambda_{last}$)*

The results for M_{last} (for CQP) and λ_{last} (for Tikhonov), where $M_{last} > M_{first}$ and $\lambda_{last} > \lambda_{first}$, are as follows:

Table 3.3: Comparison of CQP and Tikhonov Regularization ($M_{last} - \lambda_{last}$)

Measure	CQP	Tikhonov
AAE	0.9571	3.6566
${R^2}_{adj}$	0.9264	-0.2631
$RMSE$	1.2861	5.3269
r	0.9704	0.9703

From Table 3.3, it is obviously seen that CQP results are much better than the ones from Tikhonov Regularization at the last points, according to all statistical measures for this data set. While for CQP, a higher ${R^2}_{adj}$ and a lower $RMSE$ indicates a good model fit and high accuracy rate, respectively, for Tikhonov the situation is reverse. Moreover, again, the r value is high, even approximately equal, meaning that there is a good linear relationship between the actual and predicted values for both method.

3. *At Corner Values ($M_{corner} - \lambda_{corner}$)*

The results for M_{corner} (for CQP) and λ_{corner} (for Tikhonov), where $M_{last} > M_{corner} > M_{first}$ and $\lambda_{last} > \lambda_{corner} > \lambda_{first}$, are as follows:

Table 3.4: Comparison of CQP and Tikhonov Regularization at corner point ($M_{corner} - \lambda_{corner}$)

Measure	CQP	Tikhonov
AAE	0.9581	0.9548
${R^2}_{adj}$	0.9258	0.9260
$RMSE$	1.2910	1.2895
r	0.9703	0.9704

From Table 3.4, the performance measure results are nearly same for two methods. However, AAE and $RMSE$ is slightly lower than that of CQP which shows that Tikhonov solver gives more accurate results than CQP for this data. Moreover, ${R^2}_{adj}$ result are higher for Tikhonov regularization. Besides, for both two methods, the r value is high meaning that there is a good linear relationship between the actual and predicted values. Although Tikhonov is a bit better, both methods give similar results for the without interaction data set.

In conclusion, when we look at Tables 3.2 and 3.3, we see that the performance measures give the same results for λ_{first} and M_{last}, and the results are, again, approximately the same for λ_{last} and M_{first}. This can be due to the fact that CQP uses *interior point* while Tikhonov uses *exterior point* method [94, 39, 72].

3.7.2 Data with Interaction

1. *At Initial Values ($M_{first} - \lambda_{first}$)*

The results for M_{first} (for CQP) and λ_{first} (for Tikhonov) are as follows:

Table 3.5: Comparison of CQP and Tikhonov Regularization ($M_{first} - \lambda_{first}$)

Measure	CQP	Tikhonov
AAE	0.0080	0.0014
${R^2}_{adj}$	-0.2399	0.9663
$RMSE$	0.0132	0.0022
r	0.9346	0.9863

As can be seen from Table 3.5, all the performance measures show that Tikhonov solver is better at the initial points for this data. For Tikhonov regularization, a higher ${R^2}_{adj}$ and lower $RMSE$ and AAE indicate a better model fit and higher accuracy rate, respectively. Besides, for both method, r value is high meaning that there is a good linear relationship between the actual and predicted values.

2. *At Last Values ($M_{last} - \lambda_{last}$)*

The results for M_{last} (for CQP) and λ_{last} (for Tikhonov) where $M_{last} > M_{first}$ and $\lambda_{last} > \lambda_{first}$, are as follows:

Table 3.6: Comparison of CQP and Tikhonov Regularization ($M_{last} - \lambda_{last}$)

Measure	CQP	Tikhonov
AAE	0.0014	0.0079
${R^2}_{adj}$	0.9663	-0.2072
$RMSE$	0.0022	0.0130
r	0.9863	0.9318

As can be seen from Table 3.6, at the last points, all the performance measures show that CQP results are better than the ones from Tikhonov regularization for this data set. While, for CQP, a higher ${R^2}_{adj}$ and a lower $RMSE$ indicates a better model fit and higher accuracy rate, respectively, for Tikhonov regularization method, the situation is reverse. Moreover, again, r value is high, even approximately equal, meaning that there is a good linear relationship between the actual and predicted values for both method.

3. *At Corner Values ($M_{corner} - \lambda_{corner}$)*

The results for M_{corner} (for CQP) and λ_{corner} (for Tikhonov), where $M_{last} > M_{corner} > M_{first}$ and $\lambda_{last} > \lambda_{corner} > \lambda_{first}$, are as follows:

Table 3.7: Comparison of CQP and Tikhonov Regularization at corner points ($M_{corner} - \lambda_{corner}$)

Measure	CQP	Tikhonov
AAE	0.0015	0.0017
${R^2}_{adj}$	0.9602	0.9404
$RMSE$	0.0024	0.0029
r	0.9839	0.9762

From Table 3.7, the results show that both methods are approximately as good as each other at the corner point. However, CQP is a bit better than Tikhonov solver as it has smaller error rate AAE and $RMSE$ as well as higher R^2_{adj} and r for this data.

In conclusion, looking at Tables 3.5 and 3.6, again we see the same results at initial λ and last M values, while they are approximately equal for last λ and first M values. This can be due to the fact that CQP uses *interior point* while Tikhonov uses *exterior point* method [39, 72, 94].

3.8 Comparison of the Results for the three Methods (CMARS, Tikhonov and IKL)

In this part, we want to see the performance of CMARS method according to the two other classification methods, Tikhonov Regularization and IKL, whose results are obtained from the thesis [49], which is on progress. We compared the three method by using the homogeneous data set, Votes and heterogeneous data set, Hepatitis. Data descriptions are given in Table 3.8:

Table 3.8: Data set description

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

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

Normalized data sets are used and a 5-fold cross validation is applied, in all techniques. As the performance measures of IKL toolbox are Mean Error rate, Std Dev Error and Mean AUC, we calculate these values for CMARS and Tikhonov Regularization is calculated in the thesis [49], which is on progress, to make a comparison and reach the following Table 3.9:

Table 3.9: Comparison of the methods IKL, Tikhonov and CMARS

	Votes			Hepatitis		
Measure	IKL	Tikhonov	CMARS	IKL	Tikhonov	CMARS
Mean Error	0.2091	0.0020	0.1145	0.1936	0.0019	0.1935
Std Dev Error	0.1469	0.0010	0.1109	0.0456	0.0003	0.0510
Mean AUC	0.81	0.65	0.68	0.64	0.98	0.91

Here, the Std Dev Error is the standard deviation of errors over 5-fold cross-validation. From table above, for $Votes$ data, it is seen that Tikhonov performs the lowest mean error with the lowest standard deviation of errors. However, prediction accuracy is smaller than CMARS and IKL. Although IKL does not have the smallest error rates,

this method's prediction accuracy rate is the biggest. On the other hand, for the *Hepatitis* data set, the performance of Tikhonov is better than that of CMARS and IKL. Specifically, even the Mean Error and Std Dev Error is the smallest and the AUC shows that Tikhonov gave more accurate results than CMARS and IKL for this data. Note that, as the AUC tends to 1, the better the prediction accuracy [25].

As all three methods aim to help for the classification of *heterogeneous data*, performance measures are closer in each method. However, Table 3.9 shows that CMARS provides a better accuracy for the heterogeneous data Hepatitis from IKL, but Tikhonov is the best; while IKL shows a better performance for the homogenous data Votes.

CHAPTER 4

CONCLUSION AND FUTURE RESEARCH

In this thesis, we worked on a further introduction of modern continuous optimization into statistical learning. We presented *Generalized Partial Linear Models* (*GPLMs*), which have a great advantage that consists in some *grouping* that could be done for the input dimensions or features to assign appropriate submodels specifically [94], with B-splines and the parameter estimation for them.

In this study, we combined GPLM with a modified form of *Multivariate Adaptive Regression Splines* (*MARS*). The MARS algorithm is modified by constructing *Penalized Residual Sum of Squares* (*PRSS*), instead of the usual backward stepwise algorithm of MARS, as a *Tikhonov Regularization Problem*. This problem is solved by using continuous optimization, *Conic Quadratic Programming* (*CQP*). This provides us an alternative modeling technique for MARS, which is called as *Conic Multivariate Adaptive Regression Splines* (*CMARS*).

After solving our numerical example for the two data sets, which are with and without interaction, with an optimization method, CQP; we compared the results of them, according to some statistical measures. For both data sets, we discovered that the last value of M gives better results than the first value of M and the corner value of M.

Moreover, we made comparison our CQP results with Tikhonov Regularization results that are obtained from the thesis [49], which is on progress, at different parameter values by using the two different data sets. For both data sets, we observed that Tikhonov solver gives better results than CQP at initial parameter values, while CQP is better at last points. Actually, they give almost the same results at the initial

parameter of Tikhonov Regularization and at last parameter of CQP. Likewise, they are approximately equal at the last parameter of Tikhonov Regularization and at the initial parameter of CQP. This can be due to the fact that CQP uses interior point while Tikhonov Regularization uses exterior point method.

We also compared the two methods at corner points and observe that they are approximately equal for both data sets. However, with a slight difference, Tikhonov gives better results for the data with no interaction. As well, CMARS is slightly better for the data with interaction, as we expect.

In this study, in order to see the performance of CMARS method for the heterogeneous data Hepatitis and for the homogenous data Votes, according to two other classification methods that are *Infinite Kernel Learning* (*IKL*) and Tikhonov Regularization whose results are obtained from the thesis [49], which is on progress, we make a comparison by the help of some statistical performance measures; Mean Error, Std Dev Error and Mean AUC. We observed that IKL has the biggest prediction accuracy rate for the homogenous data set, Votes, when we look at the Mean AUC values. However, the Mean and Std Dev Error rate of IKL is higher than that of CMARS and Tikhonov Regularization. On the other hand, for the nonhomogeneous data set, that is Hepatitis, Tikhonov has better results for all measures.

In this thesis, we focused on GPLM and the optimization methods. We analyzed data sets and represented comparisons. 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 such as, e.g., prediction of credit default in financial mathematics is possible. Besides, identification and investigation of further important model subclasses of GPLMs can be searched. Moreover, future analysis, comparison and, if possible, partial combination of GPLMs and IKL can be studied.

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, A. Nemirovskiy and C. Roos, *Robust Solutions of Uncertain Quadratic and Conic-Quadratic Problems*, August 6, 2001

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

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

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

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

[12] 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.

[13] 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.

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

[15] 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.

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

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

[18] 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.

[19] 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.

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

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

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

[23] 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.

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

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

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

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

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

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

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

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

[32] 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.

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

[34] 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.

[35] 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.

[36] 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.

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

[38] 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).

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

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

[41] 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.

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

[43] 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.

[44] 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.

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

[46] 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.

[47] 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.

[48] E. Kartal, *Metamodelling Complex systems Using Liner and Nonlinear Regression Methods*, Master Thesis, METU, Ankara, 2007.

[49] B. Kayhan, *Parameter Estimation in Generalized Partial Linear Models with Tikhonov Regularization*, MSc. Thesis at the Institute of Applied Mathematics of METU, Ankara, 2010.

[50] 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.

[51] 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.

[52] 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.

[53] 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.

[54] M. Kriner, *Survival Analysis with Multivariate adaptive Regression Splines*, 2007, Dissertation, LMU Mnchen: Faculty of Mathematics, Computer Science and Statistics.

[55] 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).

[56] C. Lanczos, *Linear Differential Operators*, Dover, Mineola, NewYork, 1997.

[57] C.L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, Engle wood Cliffs, NJ:Prentice-Hall (1974).

[58] 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.

[59] W. Mandehall and T. Sincich, *Statistics for Engineering and The Sciences*, New Jersey: Prentice Hall, 1995.

[60] MARS from Salford Systems,
http://www.salfordsystems.com/mars/phb (accessed 25 Aug. 2009).

[61] W.L. Martinez and A.R. Martinez, *Computational Statistics Handbook with MATLAB*, London: Chapman and Hall, CRC, 2002.

[62] MATLAB Version 7.5 (R2007b)

[63] P. McCullagh and J.A. Nelder, *Generalized Linear Models*, Chapman and Hall, London, 1989.

[64] G.G. Moisen, and T.S. Frescino, Comparing five modelling techniques for predicting forest characteristics, Ecological Modelling, (2002) 209-225.

[65] D.C. Montgomery, *Design and Analysis of Experiments*, Fifth, John Wiley & Sons Inc., New York, NY, 2001.

[66] 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.

[67] M. Müller, *Generalized Linear Models*, Statistics and Computing, Stat. Comput., 11, 2004, 4, 591–619, 0960-3174, STACE3, Database Expansion Item.

[68] 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.

[69] 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).

[70] 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).

[71] A. Nemirovski, *Five Lectures On Modern Convex Optimization (2002)*, http://iew3.technion.ac.il/Labs/Opt/opt/LN/Final.pdf, accessed 15 Oct. 2009.

[72] Y. E. Nesterov and A.S. Nemirovskii, *Interior Point Methods in Convex Programming*, SIAM, 1993.

[73] Y.E Nesterov and A.S Nemirovski, *Interior Point Polynomial Algorithms in Convex Programming*, SIAM, 1994.

[74] J. Neter, M. Kutner, W. Wasserman and C. Nachtsheim, *Applied Liear Statistical Models*, Boston, MA: WCB/McGrawHill, 1996.

[75] 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.

[76] J.M. Ortega and W.C. Rheinboldt, *Iterative solution of nonlinear equations in several variables*, Academic Press, New York, 1970.

[77] 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.

[78] 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.

[79] 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.

[80] 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.

[81] S. Özöğür-Akyüz and G.-W. Weber *On numerical optimization theory of infinite kernel learning*, Journal of Global Optimization, 2009.

[82] D.J. Poirier, *The Econometrics of Structural Change with Special Emphasis on Spline Functions*, New York: North-Holland Publishing Co. (1976).

[83] J. Renegar, *Mathematical View of Interior Point Methods in Convex Programming*, Society for Industrial and Applied Mathematics (SIAM), 2000.

[84] P.J. Rousseeuw and A.M. Leroy, *Robust regression and outlier detection*, Hoboken, N.J: Wiley-Interscience, 2003.

[85] S. Sonnenburg, G. Rätsch, C. Schafer and B. Schölkopf. *Large scale multiple kernel learning*, J. Machine Learning Research, 2006.

[86] *SPSS 16.0 GPL Reference Guide*, Chicago, IL: SPSS Inc, 2007. http://support.spss.com/ProductsExt/SPSS/Documentation/SPSSforWindows/ accessed 26 Ags. 2009.

[87] R.E. Steuer, *Multiple Criteria Optimisation: Theory, Computation and Application*, New York: John Wiley and Sons, NY, 1986.

[88] G. Still, *Semi-infinite programming: An introduction, preliminary version*, Technical report, University of Twente Department of Applied Mathematics, Enschede, The Netherlands, 2004.

[89] P. Taylan, F. Yerlikaya-Özkurt and G.-W. Weber, *Parameter Estimation For Semiparametric Models with CMARS and its applications*, working paper, IAM, METU, 2010.

[90] 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.

[91] 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.

[92] 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).

[93] 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).

[94] 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.

[95] The MOSEK optimization toolbox for MATLAB manual. Version 5.0 (Revision 105). www.mosek.com accessed 29 March 2010

[96] 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.

[97] G. Upton and I. Cook, *The Dictionary of Statistics*, Oxford University Press Inc., New York, 2008.

[98] 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.

[99] 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).

[100] 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.

[101] 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).

[102] 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.

[103] G.-W. Weber, *Generalized Semi-Infinite Optimization and Related Topics*, volume 29 of Research and Exposition in Mathematics. Heldermann Verlag, Germany, 2003.

[104] 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.

[105] S.N. Wood, *Generalized additive models*, Texts in Statistical Science Series, An Introduction with *R*, Chapman & Hall/CRC, Boca Raton, FL, 2006.

[106] 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.

[107] 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.

[108] 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