A NEW CONTRIBUTION TO NONLINEAR ROBUST REGRESSION AND CLASSIFICATION WITH MARS AND ITS APPLICATIONS TO DATA MINING FOR QUALITY CONTROL IN MANUFACTURING

FATMA YERLİKAYA

SEPTEMBER 2008

A NEW CONTRIBUTION TO NONLINEAR ROBUST REGRESSION AND CLASSIFICATION WITH MARS AND ITS APPLICATIONS TO DATA MINING FOR QUALITY CONTROL IN MANUFACTURING

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

BY

FATMA YERLİKAYA

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

SEPTEMBER 2008

Approval of the Graduate School of Applied Mathematics

Prof.Dr. Ersan AKYILDIZ
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Bülent KARASÖZEN
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. İnci Batmaz
Co-supervisor

Prof. Dr. Gerhard Wilhelm Weber
Supervisor

Examining Committee Members

Prof. Dr. Gülser Köksal

Prof. Dr. Gerhard Wilhelm Weber

Assoc. Prof. Dr. İnci Batmaz

Assist. Prof. Dr. Hakan Öktem

Assist. Prof. Dr. Pakize Taylan

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: Fatma Yerlikaya

Signature:

ABSTRACT

A NEW CONTRIBUTION TO NONLINEAR ROBUST REGRESSION AND CLASSIFICATION WITH MARS AND ITS APPLICATIONS TO DATA MINING FOR QUALITY CONTROL IN MANUFACTURING

Fatma Yerlikaya
M.Sc., Department of Scientific Computing
Supervisor: Prof. Dr. Gerhard Wilhelm Weber
Co-Supervisor: Assoc. Prof. Dr. İnci Batmaz
September 2008, 102 pages

Multivariate adaptive regression spline (MARS) denotes a modern methodology from statistical learning which is very important in both classification and regression, with an increasing number of applications in many areas of science, economy and technology.

MARS is very useful for high dimensional problems and shows a great promise for fitting nonlinear multivariate functions. MARS technique does not impose any particular class of relationship between the predictor variables and outcome variable of interest. In other words, a special advantage of MARS lies in its ability to estimate the contribution of the basis functions so that both the additive and interaction effects of the predictors are allowed to determine the response variable.

The function fitted by MARS is continuous, whereas the one fitted by classical classification methods (CART) is not. Herewith, MARS becomes an alternative to CART. The MARS algorithm for estimating the model function consists of two complementary algorithms: the forward and backward stepwise algorithms. In the first step, the model is built by adding basis functions until a maximum level of complexity is reached. On the other hand, the backward stepwise algorithm is began by removing the least significant basis functions from the model.

In this study, we propose not to use the backward stepwise algorithm. Instead, we construct a penalized residual sum of squares (PRSS) for MARS as a Tikhonov regularization problem, which is also known as ridge regression. 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. In particular, we apply the elegant framework of conic quadratic programming which is an area of convex optimization that is very well-structured, herewith, resembling linear programming and, hence, permitting the use of interior point methods. The boundaries of this optimization problem are determined by the multiobjective optimization approach which provides us many alternative solutions.

Based on these theoretical and algorithmical studies, this MSc thesis work also contains applications on the data investigated in a TÜBİTAK project on quality control. By these applications, MARS and our new method are compared.

Keywords: Statistical Learning, MARS, Penalty Methods, Continuous Optimization, Conic Quadratic Programming, Well-Structured Convex Problems, Interior Point Methods, Multiobjective Optimization

ÖZ

DOĞRUSAL OLMAYAN SAĞLAM REGRESYON VE SINIFLANDIRMAYA MARS İLE YENİ BİR KATKI VE BU KATKININ ENDÜSTRİDE KALİTE KONTROLÜ AMAÇLI VERİ MADENCİLİĞİ UYGULAMALARI

Fatma Yerlikaya
Yüksek Lisans, Bilimsel Hesaplama Bölümü
Tez Yöneticisi: Prof. Dr. Gerhard Wilhelm Weber
Ortak Tez Yöneticisi: Doç. Dr. İnci Batmaz

Eylül 2008, 102 sayfa

Çok değişkenli uyarlanabilir regresyon eğrileri (MARS), istatistiksel öğrenmede modern bir teknoloji olarak görülmektedir. Hem sınıflandırma hem de regresyonda çok büyük bir öneme sahip olan MARS, ekonomi, bilim ve teknoloji alanında giderek artan bir şekilde uygulanmaktadır.

Çok boyutlu problemlerin çözümünde oldukça elverişli olan MARS, doğrusal olmayan çok değişkenli fonksiyonlara uygunluk bakımından da büyük bir olanak vaad etmektedir. MARS tekniği, bağımsız değişkenlerle bağımlı değişken arasında belirli bir ilişki biçimi öngörmez. Bir başka değişle, bağımlı değişkeni tanımlamak için bağımsız değişkenlerin eklemeli ve etkileşimsel katkılarına yer vermektedir. Bu ise MARS' ın önemli bir avantajı olan, temel fonksiyonların katkısını tahmin etme yeteneğini ortaya koymaktadr.

MARS' ın uygunluk sağladığı fonksiyon sürekli bir fonksiyon iken, klasik sınıflandırma yöntemlerinden biri olan CART' ın uygunluk sağladığı fonksiyon sürekli değildir. Bu nedenle MARS, sürekli fonksiyonlara uygunluk bakımından, CART' ın bir alternatifi olarak görülmektedir.

Model fonksiyonunu tahmin etmek için MARS iki aşamalı bir algoritmadan oluşmaktadır. Birinci aşamada, maksimum karmaşıklık düzeyine ulaşıncaya dek temel fonksiyonlar eklenerek model yapılandırılır. İkinci aşamada ise modele katkısı en az fonksiyonlar elenir.

Bu çalışmada biz, MARS' ın ikinci aşamasını oluşturan geriye doğru eleme yöntemi yerine penaltı yöntemini kullanmayı önermekteyiz. Bu amaçla, 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. Bu optimizasyon probleminin sınırlarının, çok amaçlı optimizasyon yaklaşımı ile belirlenmesi, bize pek çok alternatif çözüm sağlamaktadır.

Bu tez, yukarıda bahsi edilen teorik ve algoritmik çalışmaların yanısıra , kalite kontrolüne yönelik bir TÜBİTAK projesinin verileri üzerine uygulamaları da kapsamaktadr. Bu uygulamalarda, MARS ve geliştirdiğimiz yeni metod karşılaştırılmıştır.

Anahtar Kelimeler: İstatistiksel Öğrenme, MARS, Penaltı Metodu, Sürekli Optimizasyon, İkinci Dereceden Konik Karesel Programlama, İyi Yapılandırılmış Dışbükey Problemler, İç Nokta Yöntemleri, Çok Amaçlı Optimizasyon

To my family

ACKNOWLEDGMENT

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

I would like to give special thanks to Prof. Dr. Gülser Köksal and Assist. Prof. Dr. Pakize Taylan for their valuable contribution to this study.

I am thankful to my friend Neslihan Özmen for not letting me feel alone and motivating me. I am grateful to my friend Selime Gürol for her encouragement, understanding and guidance throughout this study.

I would like thank to Elçin Kartal and Efsun Kurum for their friedship and also for their help during my study.

It is very important for the continuation of the study to be in a peaceful place which is supported during all my study from department of Industrial Engineering (IE). So, thanks to all IE personel for making my stay in the department enjoyable.

This work was supported by The Scientific and Technical Research Council of Turkey MSc. Scholarship Program (TÜBİTAK Yurt İçi Yüksek Lisans Burs Programı).

I wish to thank the members of the TÜBİTAK project on Use and Development of Data Mining Methods for Quality Control in Manufacturing for support and encouragement of this study.

None of my studies would have been possible without the support of my family. Thanks each of them for their continuous support, endless love and patience. They always believed and encouraged me. Thanks for their existence.

TABLE OF CONTENTS

CHAPTER

APPENDICES

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Data mining (DM) is one of the most important techniques of scientific and technologic studies. It is a complicated, e.g., interdisciplinary, process dealing with outcomes of experiments, records, measurements, questionnaires, etc.. This process inevitably contains difficulties such as computational time, inaccurate predictions, interpretability and transfering results into other computational systems. Moreover, complex data sets are another challenging matter in DM process. This situation motivates to develope innovative DM tools.

In this thesis, we study multivariate adaptive regression splines (MARS) which is developed by Friedman in 1991 and used successfully in many areas of science, economy and technology. We also developed our C-MARS as a modification of MARS.

As a modern methodology from statistical learning MARS is very important in classification and regression. Its ability to estimate the contribution of basis functions and additive and interaction effects of the predictors is a special advantage of MARS. This makes MARS a useful tool for high-dimensional problems. In order to estimate the model function, MARS uses two step-wise algorithms, forward and backward. In the first step, the model is generated by adding basis functions until a maximum level of complexity is reached. In the backward step, the basis functions having least contribution to the overall fit are removed from the model.

Instead of this second step, we propose to construct a ***penalized*** *residual sum of squares (PRSS)* enabling us to control complexity and the accuracy of the model. Our *PRSS* transforms MARS in to a Tikhonov regularization problem. In order to solve this problem, we use conic quadratic programming (CQP) as a continuous optimization technique. The boundaries of this optimization problem are determined by the multiobjective optimization approach. This provides us many alternative solutions. In order to see the efficiency of C-MARS, our modificated version of MARS, we compare these two methods by using three different data sets, while two of these data sets are simulation data sets, the other one is a real-world data about metal casting industry obtained from a TÜBİTAK Project (the project number is 105M138). For comparing these two models, three different data sets are used and one of them is real-world data. This comparison is applied first of all according to method based measures, then general performance measures are used. For model-free measures, cross validation is used. Besides these comparisons, by using Tukey test, it is aimed at to determine whether there are statistically significant differences between the averaged values of employed measures. According to an ordinal semantic scale -“very poor”, “poor”, “good”, “very good”- the results are re-evaluated.

In this thesis, a literature review of regression models is given briefly in Chapter 2. This chapter also includes a comprehensive information about Tikhonov regularization, CQP and multiobjective optimization which constitute the background of our study. Chapter 3 contains a detailed description of MARS and its modified version C-MARS. In this chapter, a numerical example for C-MARS is also presented. The applications of the MARS and C-MARS take place in Chapter 4. The comparison of the methods with respect to the determined performance measures and evaluations are included in Chapter 5. Moreover, an outlook on further studies is given in this chapter.

CHAPTER 2

LITERATURE SURVEY AND BACKGROUND

2.1 Literature Survey

The data of real-world problems are finite, that is “discrete”. *Regression* models, which are also called as *discrete approximation* or *Gaussian approximation*, are used for analyzing data sets by disclosing the relationship between the predictors and response variable(s). Regression analysis is the most widely used statistical technique, in investigating and modeling the relationship between variables. There are many regression models. They are used for several purposes such as data description, parameter estimation for learning, prediction and control [44].

Almost in every field such as engineering, the physical and chemical sciences, economics and social sciences, scientists and engineers use regression models for summarizing or describing a set of data [2, 28, 44].

2.1.1 Linear Regression Model

If a regression model is linear in a fitted parameters, it is called as *linear regression model* (*LRM*) [2]. In general, the following equation

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k + \epsilon \qquad (2.1.1)$$

represents an *LRM* [44].

In equation (2.1.1), y is called the *response variable* (or the *dependent variable*) and x_j $(j = 1, 2, ..., k)$ are called the *regressor variables* (*predictor* or *independent variables*) [44]. Furthermore, ϵ is a random error component. The errors are assumed to have a normal distribution with a mean of zero and unknown constant variance, σ^2. It is also assumed that the errors are uncorrelated. In other words, the value of one error is independent from the value of any other error [44]. These are also known as "white noise assumption". The parameter β_0 means the intercept and the other parameters β_j $(j = 1, 2, ..., k)$ are the regression coefficients. The parameter β_l represents the expected change in the response y per unit change in x_l when all of the remaining regressor variables x_j $(j = 1, 2, ..., k;\ j \neq l)$ are held constant [44].

The word linear is used to indicate that the model is linear in the parameters $\beta_0, \beta_1, \beta_2, ..., \beta_k$. It does not mean that y is a linear function of the coordinates x_j. Even in case of a nonlinear fashion in which y is related to the x_j' s will be treated as a linear regression model when the equation is linear in the components β_j [44].

In most real-world problems, the values of the regression coefficients β_j and the error variance σ^2 are not known. These parameters and the error variance must be estimated from a sample data set. The fitted regression equation or the model enable to predict future observations of the response variable y. *Least squares estimation (LSE)* or *maximum likelihood estimation (MLE)* are two widely used optimization methods applied on the regression model for estimating the unknown regression parameters [44, 55].

Least Squares Estimation Method

The method of LS is used for estimating the regression coefficients [44] $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, ..., \beta_k)^T$ in $y = \beta_0 + \sum\limits_{j=1}^{k} \beta_j x_j$ to minimize the residual sum of squares (RSS) [25]. The below (RSS) formula is written in terms of the N pairs of data (x_i, y_i) $(i = 1, 2, ..., N)$ as follows:

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

Here, $RSS(\boldsymbol{\beta})$ is a quadratic function of the parameters. For minimizing RSS, it is the easiest expression to write it in matrix notation as follows [25]:

$$\begin{aligned} RSS(\boldsymbol{\beta}) &= (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}), \\ &= \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 . \end{aligned}$$

In this equation, $\boldsymbol{X}$ denotes the $N \times (k+1)$ matrix with each row input vector, a column of entries 1 standing in the first position in the matrix $\boldsymbol{X}$ and $\boldsymbol{y}$ is the N-vector of output in the data set. The Euclidean norm is denoted by $\|.\|_2^2$ [25]. Differentiating RSS with respect to $\boldsymbol{\beta}$ results in:

$$\nabla RSS(\boldsymbol{\beta}) = -2\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}), \tag{2.1.2}$$

$$\nabla^2 RSS(\boldsymbol{\beta}) = -2\boldsymbol{X}^T\boldsymbol{X}. \tag{2.1.3}$$

The second derivative matrix of RSS in equation (2.1.3) is a Hessian matrix. After setting the first derivative (2.1.2) to zero, we get

$$\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) = 0, \tag{2.1.4}$$

the normal equations are obtained [25].

If $\boldsymbol{X}^T\boldsymbol{X}$ is nonsingular, then the unique solution is given by

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}, \tag{2.1.5}$$

and the fitted values are defined by [25]

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

If $\boldsymbol{X}^T\boldsymbol{X}$ is singular, then a method of solving the LS problem is called as *singular value decomposition (SVD)*. By using the *SVD* method, the solution of the LS problem is obtained from the normal equations as shown in equation (2.1.4). By this, a particular (especially, also norm-minimal) solution of (2.1.5) is obtained [2].

Maximum Likelihood Estimation Method

Although the LS estimation method is generally very convenient, it does not make much sense in some cases. If the form of the distribution of the error is known, a more general principle for estimating regression coefficients is *MLE method* [25]. The model which is defined in the equation (2.1.1) can be written in matrix notation as follows:

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

Here, the error term $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, ..., \epsilon_N)^T$ is the residual vector and this vector components are normally and independently distributed with constant variance σ_i^2 like in *LSE* [25]. Given the set of data (x_i, y_i) $(i = 1, 2, ..., N)$, $\boldsymbol{y} = (y_1, y_2, ..., y_N)^T$, $\boldsymbol{X}$ is an $N \times (k+1)$ matrix and $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, ..., \beta_k)^T$. The probability density function $f(y_i, \boldsymbol{\beta})$ for y_i $(i = 1, 2, ..., N)$ is

$$f(y_i, \boldsymbol{\beta}) = \frac{1}{\sigma_i (2\pi)^{1/2}} \exp(-\frac{1}{2\sigma_i^2}(y_i - (\boldsymbol{X}\boldsymbol{\beta})_i)^2). \tag{2.1.6}$$

This expression corresponds to the general framework for the probability density function by taking $\boldsymbol{\sigma} = \mathrm{diag}(\sigma_1, \sigma_2, ..., \sigma_N)^T$. The likelihood function for the complete data set is

$$L(\boldsymbol{\beta}|\boldsymbol{y}) = \frac{1}{(2\pi)^{N/2} \prod\limits_{i=1}^{N} \sigma_i^2} \prod_{i=1}^{N} \exp(-\frac{1}{2\sigma_i^2}(y_i - (\boldsymbol{X}\boldsymbol{\beta})_i)^2). \tag{2.1.7}$$

The constant factor $\frac{1}{(2\pi)^{N/2} \prod\limits_{i=1}^{N} \sigma_i^2}$ does not affect the maximization of L, so it can be solved as

$$\max_{\boldsymbol{\beta}} \quad \prod_{i=1}^{N} \exp(-\frac{1}{2\sigma_i^2}(y_i - (\boldsymbol{X}\boldsymbol{\beta})_i)^2). \tag{2.1.8}$$

Since the logarithm is a monotonically increasing function, the equation (2.1.8) can be equivalently solved by

$$\max_{\boldsymbol{\beta}} \quad \log \prod_{i=1}^{N} \exp(-\frac{1}{2\sigma_i^2}(y_i - (\boldsymbol{X}\boldsymbol{\beta})_i)^2).$$

After making some calculations, the following equation is given by

$$\max_{\boldsymbol{\beta}} \quad -\sum_{i=1}^{N} \frac{\left(y_i - (\boldsymbol{X}\boldsymbol{\beta})_i\right)^2}{2\sigma_i^2}. \tag{2.1.9}$$

By changing sign and ignoring the constant factor of 1/2, the maximization problem is transformed into the following minimization problem:

$$\min_{\boldsymbol{\beta}} \quad \sum_{i=1}^{N} \frac{\left(y_i - (\boldsymbol{X}\boldsymbol{\beta})_i\right)^2}{\sigma_i^2}. \tag{2.1.10}$$

This minimization problem is identical to the LS problem of $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$. To combine the data points standard deviations into this solution, a diagonal weight matrix $\boldsymbol{W} = \text{diag}\ (1/\sigma_1, 1/\sigma_2, ..., 1/\sigma_N)$ is used. The weighted system of equations is

$$\boldsymbol{y}_w = \boldsymbol{X}_w\boldsymbol{\beta} + \boldsymbol{\epsilon}. \tag{2.1.11}$$

Here, $\boldsymbol{X}_w := \boldsymbol{W}\boldsymbol{X}$ and $\boldsymbol{y}_w := \boldsymbol{W}\boldsymbol{y}$. The solution of the above weighted system is

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

if $(\boldsymbol{X}_w^T\boldsymbol{X}_w)^{-1}$ exists. Thus, the *LS solution* of $\boldsymbol{y}_w = \boldsymbol{X}_w\boldsymbol{\beta} + \boldsymbol{\epsilon}$ is turned out to be the *ML solution* [2].

2.1.2 Nonlinear Regression Model

Although linear regression is a powerful method for analyzing data when the model is linear in the parameters [6], there are many situations where the linear regression model is not appropriate [44]. Indeed, in general, life has various nonlinear features. Many processes in nature, technology and economy, especially, financial processes, involve stochastic fluctuations. Therefore, *stochastic differential equations (SDEs)* that have nonlinearly embedded parameters, are considered. Moreover, the true relationship between the response variable and regressors can be expressed by a differential equation or by a solution of a differential equation. In such cases, we may use *nonlinear regression model* (*NRM*) [2, 44, 45]. A *NRM* is nonlinear in the unknown parameters. In general, the NRM is defined by the following equation

$$y = f(\boldsymbol{x}, \boldsymbol{\theta}) + \epsilon, \tag{2.1.12}$$

where $\boldsymbol{\theta}$ is a $k \times 1$ vector of unknown parameters $\boldsymbol{\theta} = (\theta_1, \theta_2, ..., \theta_k)^T$; ϵ is an uncorrelated random error term with constant variance σ^2 and a zero of mean, as in the the LR case, and $f(\boldsymbol{x}, \boldsymbol{\theta})$ is the expectation function for the nonlinear regression model and $\boldsymbol{x} = (x_1, x_2, ..., x_k)^T$ is an input vector [44]. In nonlinear regression models, at least one of the derivatives of the expectation function with respect to the parameters depends on at least one of the parameters [44]. The expression in equation (2.1.12) can be written as a vector form in terms of the unknown parameters θ_j $(j = 1, 2, ..., k)$ by

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

where $\boldsymbol{\eta}(\boldsymbol{\theta}) := (f(\boldsymbol{x}_1, \boldsymbol{\theta}), f(\boldsymbol{x}_2, \boldsymbol{\theta}), ..., f(\boldsymbol{x}_N, \boldsymbol{\theta}))^T$ and $\boldsymbol{x}_i$ $(i = 1, 2, ..., N)$ is given input data.

Nonlinear Least Squares Estimation Method

In a sample of N observations, the response and the regressors are y_i and $\boldsymbol{x}_i = (x_{i1}, x_{i2}, ..., x_{ik})^T$ $(i = 1, 2, ..., N)$, respectively. The function in (2.1.12) can be written in a form of LS as follows:

$$S(\boldsymbol{\theta}) = \sum_{i=1}^{N} (y_i - f(\boldsymbol{x}_i, \boldsymbol{\theta}))^2. \tag{2.1.13}$$

To find the LS estimators, the equation (2.1.13) must be differentiated with respect to each element of $\boldsymbol{\theta}$. This provides a set of k normal equations for the nonlinear regression situation. The first order necessary optimality conditions are obtained by the following equations [44]:

$$\sum_{i=1}^{N} (y_i - f(\boldsymbol{x}_i, \boldsymbol{\theta})) \frac{\partial f(\boldsymbol{x}_i, \boldsymbol{\theta})}{\partial \theta_j} |_{\boldsymbol{\theta}=\boldsymbol{\theta}^*} = 0. \tag{2.1.14}$$

In a nonlinear regression model, solving the normal equations can be very difficult because the derivatives are functions of the unknown parameters and the expectation function has a nonlinear function form [44].

Maximum-Likelihood Estimation Method

The method of *MLE* will lead to LS when the error terms in the model are normally and independently distributed with the variances σ_i^2 per experiment [44]. The likelihood function for the complete data set $\boldsymbol{z}_i = (\boldsymbol{x}_i, y_i)$ $(i = 1, 2, ..., N)$ is expressed by

$$L(\boldsymbol{\theta}|\boldsymbol{y}) := \frac{1}{(2\pi)^{N/2}\prod_{i=1}^{N}\sigma_i^2}\prod_{i=1}^{N}\exp(-\frac{1}{2\sigma_i^2}\left[y_i - f(\boldsymbol{x}_i, \boldsymbol{\theta})\right]^2). \tag{2.1.15}$$

This expression corresponds to the general framework for the likelihood function by taking $\boldsymbol{\sigma} = \mathrm{diag}(\sigma_1, \sigma_2, ..., \sigma_N)^T$. Then, maximizing the likelihood function in equation (2.1.15) and minimizing the LS in equation (2.1.13) are equivalent kind of problems in the normal-theory case [44].

A nonlinear LS problem is an unconstraint minimization problem of the following form presented by Nash and Sofer (1996):

$$\min_{\boldsymbol{\theta}} \quad F(\boldsymbol{\theta}) = \frac{1}{2}\sum_{i=1}^{N} g(\boldsymbol{z}_i, \boldsymbol{\theta})^2.$$

The function $g(\boldsymbol{z}_i, \boldsymbol{\theta}) = y_i - f(\boldsymbol{x}_i, \boldsymbol{\theta})$ is called “least squares” because the sum of squares of this function is minimized. This minimization problem can be represented in vector notation as follows:

$$\min_{\boldsymbol{\theta}} \quad F(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{\eta}(\boldsymbol{\theta})^T\boldsymbol{\eta}(\boldsymbol{\theta}),$$

where $\boldsymbol{\eta}$ is the vector valued function $\boldsymbol{\eta}(\boldsymbol{\theta}) = (g(\boldsymbol{z}_1, \boldsymbol{\theta}), g(\boldsymbol{z}_2, \boldsymbol{\theta}), ..., g(\boldsymbol{z}_N, \boldsymbol{\theta}))^T$ and $\boldsymbol{z}_i = (x_{i1}, x_{i2}, ..., x_{ik}, y_i)^T$ are our data vectors. In fact, by the chain rule

$$\nabla F(\boldsymbol{\theta}) := \nabla\boldsymbol{\eta}(\boldsymbol{\theta})\boldsymbol{\eta}(\boldsymbol{\theta}), \tag{2.1.16}$$

is obtained, where $\nabla\boldsymbol{\eta}(\boldsymbol{\theta})$ is an $(k \times N)$-matrix valued function. By row-wise differentiation of $\nabla\boldsymbol{\eta}(\boldsymbol{\theta})$ and using this gradient representation, the Hessian matrix of F is obtained:

$$\nabla^2 F(\boldsymbol{\theta}) := \nabla\boldsymbol{\eta}(\boldsymbol{\theta})\boldsymbol{\eta}(\boldsymbol{\theta})^T + \sum_{i=1}^{N} g(\boldsymbol{z}_i, \boldsymbol{\theta})\nabla^2_{\boldsymbol{\theta}} g(\boldsymbol{z}_i, \boldsymbol{\theta}).$$

Let $\boldsymbol{\theta}^*$ be a solution of the nonlinear LS problem and suppose that $g(\boldsymbol{z}_i, \boldsymbol{\theta}^*) = 0$ $(i = 1, 2, ..., N)$. Then, all the residuals r_i vanishes and the model fits data without error. As a result, $\boldsymbol{\eta}(\boldsymbol{\theta}^*) = 0$ and, by (2.1.16), $\nabla F(\boldsymbol{\theta}^*) = 0$, which confirms the first-order necessary optimality condition. Then, the Hessian matrix of F is obtained by

$$\nabla^2 F(\boldsymbol{\theta}^*) := \nabla\boldsymbol{\eta}(\boldsymbol{\theta}^*)\nabla\boldsymbol{\eta}(\boldsymbol{\theta}^*)^T,$$

which is a positive semi-definite matrix, just as we expected by *second-order necessary optimality condition.* In case where $\nabla\boldsymbol{\eta}(\boldsymbol{\theta}^*)$ is a matrix of full rank, i.e., $\text{rank}(\nabla\boldsymbol{\eta}(\boldsymbol{\theta}^*)) = k$ $(k \leq N)$, then $\nabla^2_{\boldsymbol{\theta}} F(\boldsymbol{\theta}^*)$ is positive definite, i.e., the *second-order necessary optimality condition* is satisfied such that $\boldsymbol{\theta}^*$ is also a strict local minimizer.

The Gauss-Newton Method

There are a number of specialized *nonlinear least squares* methods. The simplest of these methods is the *Gauss-Newton method.* The Gauss-Newton method of parameter estimation corresponds to the Newton's method for nonlinear regression problems [2, 44]. The Gauss-Newton method uses the following approximation:

$$\nabla^2 F(\boldsymbol{\theta})\boldsymbol{\delta} = -\nabla F(\boldsymbol{\theta}).$$

It computes a search direction using the formula for Newton's method, but replaces the Hessian with the approximation. Therefore, it has the form

$$\nabla\boldsymbol{\eta}(\boldsymbol{\theta})\nabla\boldsymbol{\eta}(\boldsymbol{\theta})^T\boldsymbol{\delta} = -\nabla\boldsymbol{\eta}(\boldsymbol{\theta})\boldsymbol{\eta}(\boldsymbol{\theta}),$$

where $\boldsymbol{\delta}$ is Gauss-Newton increment $\boldsymbol{\delta} = \boldsymbol{\theta}_k - \boldsymbol{\theta}_{k-1}$.

If $\boldsymbol{\eta}(\boldsymbol{\theta}^*) \approx 0$ and $\text{rank}(\nabla\boldsymbol{\eta}(\boldsymbol{\theta}^*)) = k$, then, near the solution $\boldsymbol{\theta}^*$, Gauss-Newton behaves like Newton's method. Since the second derivatives creates computational costs, it is unnecessary to calculate them. Gauss-Newton's method sometimes behaves poorly if there is one or a number of outliers, i.e., if the model does not fit the data well, or if $\text{rank}(\nabla\boldsymbol{\eta}(\boldsymbol{\theta}^*))$ is not of full rank k. In these cases, there is a poor approximation to the Hessian of F.

Many other methods for nonlinear least-squares can be interpreted as using an approximation to the second additive form in the formula for the Hessian, i.e., and each of the functions $g(\boldsymbol{z}_i, \boldsymbol{\theta})$ corresponds to a residual in nonlinear problem which may arise in a mathematical modelling or an inverse problem.

$$\sum_{i=1}^{N} g(\boldsymbol{z}_i, \boldsymbol{\theta}) \nabla^2_{\boldsymbol{\theta}} g(\boldsymbol{z}_i, \boldsymbol{\theta}). \tag{2.1.17}$$

The Levenberg-Marquardt Method

Although the Gauss-Newton iterative method for nonlinear LS estimation is simple and easy to implement for finding $\boldsymbol{\theta}^*$, it may converge very slowly in some problems. It may also generate a move in the wrong direction. Even in some extreme cases, it may fail to converge at all [6, 44]. To overcome these shortcomings, same modifications and refinements have been developed [44]. One of the modification is the *Levenberg-Marquardt method* (*LM*).

The LM method uses a rank-improving approximation in equation (2.1.17):

$$\sum_{i=1}^{N} g(\boldsymbol{z}_i, \boldsymbol{\theta}) \nabla^2_{\boldsymbol{\theta}} g(\boldsymbol{z}_i, \boldsymbol{\theta}) \approx \lambda \boldsymbol{I}_k, \tag{2.1.18}$$

with some scalar $\lambda \geq 0$. This approximation yields the following linear system:

$$(\nabla\boldsymbol{\eta}(\boldsymbol{\theta})\nabla\boldsymbol{\eta}(\boldsymbol{\theta})^T + \lambda \boldsymbol{I}_k)\boldsymbol{\delta} = -\nabla\boldsymbol{\eta}(\boldsymbol{\theta})\boldsymbol{\eta}(\boldsymbol{\theta}). \tag{2.1.19}$$

The LM method is also implemented in the context of a *trust region* strategy. There, $\boldsymbol{\delta}$ is a search direction and it is obtained by minimizing a quadratic model of the objective function with Gauss-Newton approximation to the Hessian:

$$\begin{array}{ll} \min\limits_{\boldsymbol{\delta}} & Q(\boldsymbol{\delta}) := F(\boldsymbol{\theta}) + \boldsymbol{\delta}^T \nabla \boldsymbol{\eta}(\boldsymbol{\theta}) \boldsymbol{\eta}(\boldsymbol{\theta}) + \frac{1}{2} \boldsymbol{\delta}^T \nabla \boldsymbol{\eta}(\boldsymbol{\theta}) \nabla \boldsymbol{\eta}(\boldsymbol{\theta})^T \boldsymbol{\delta} \\ \text{subject to} & \|\boldsymbol{\delta}\|_2 \leq \Delta. \end{array} \tag{2.1.20}$$

Here, λ is indirectly determined by picking a value of Δ ($\Delta > 0$). The scalar Δ can be chosen based on the effectiveness of the Gauss-Newton.

LM method can approximately be interpreted as the Gauss-Newton method if $\lambda \approx 0$ and steepest-descent method if λ is very large. An adaptive and sequential way of choosing λ and, by this, of the adjustment of mixture between the methods the methods of Gauss-Newton and steepest-descent, is presented in [2, 45]. The term $\lambda \boldsymbol{I}_k$ in equation (2.1.19) can also be regarded as regularization term. Another way to solve the system (2.1.19) for given $\boldsymbol{\theta} = \boldsymbol{\theta}^{\nu}$, i.e., to find $(\nu + 1)$-st iterate $\boldsymbol{\delta} = \boldsymbol{\delta}^{\nu}$, consists of an application of LS estimation. The equation (2.1.19) can be represented by $\boldsymbol{G}\boldsymbol{\delta} = \boldsymbol{d}$, where $\boldsymbol{G} = \nabla \boldsymbol{\eta}(\boldsymbol{\theta}) \nabla \boldsymbol{\eta}(\boldsymbol{\theta})^T + \lambda \boldsymbol{I}_k$ and $\boldsymbol{d} = -\nabla \boldsymbol{\eta}(\boldsymbol{\theta}) \boldsymbol{\eta}(\boldsymbol{\theta})$. Then, the regularization form of the problem can be written by adding to the squared residual norm $\| \boldsymbol{G}\boldsymbol{\delta} - \boldsymbol{d} \|_2^2$, a penalty or regularization term of the form $\gamma^2 \|\boldsymbol{L}\boldsymbol{\delta}\|_2^2$, as follows:

$$\min_{\boldsymbol{\theta}} \quad \left\| (\nabla \boldsymbol{\eta}(\boldsymbol{\theta}) \nabla^T \boldsymbol{\eta}(\boldsymbol{\theta}) + \lambda \boldsymbol{I}_k) \boldsymbol{\delta} - (-\nabla \boldsymbol{\eta}(\boldsymbol{\theta}) \boldsymbol{\eta}(\boldsymbol{\theta})) \right\|_2^2 + \gamma^2 \|\boldsymbol{L}\boldsymbol{\delta}\|_2^2, \tag{2.1.21}$$

where $\boldsymbol{L}$ may be the unit matrix, but it can also be a discrete differentiation of first or second order. This minimization problem bases on the ***tradeoff*** between both accuracy, i.e., a small sum of square error, and not too high in complexity. This tradeoff is established through the penalty parameters γ^2. If $\gamma^2 \geq 0$, then the set of feasible solutions becomes smaller, and the minimum value of $\| \boldsymbol{G}\boldsymbol{\delta} - \boldsymbol{d} \|_2^2$ increases. If $\gamma^2 \approx 0$, then the set of feasible models and the minimum value of $\|\boldsymbol{L}\boldsymbol{\delta}\|_2^2$ decreases. This regularization reduces the complexity of the model. The minimization problem above is called a *Tikhonov regularization* problem. As an alternative to our penalization approach, the regularization term $\|\boldsymbol{L}\boldsymbol{\delta}\|_2^2$ can be bounded by an inequality constraint. This optimization problem can be turned to a *conic quadratic programming* (*CQP*) problem for finding step $\boldsymbol{\delta}^{\nu}$ and also next iterate $\boldsymbol{\theta}^{\nu+1} = \boldsymbol{\theta}^{\nu} + \boldsymbol{\delta}^{\nu}$. In order to determine step $\boldsymbol{\delta}$, with a suitable and adaptive choice of an upper bound

M, the CQP can be written as [25, 51, 52]:

$$\min_{\boldsymbol{\theta}} \quad \left\|(\nabla\boldsymbol{\eta}(\boldsymbol{\theta})\nabla^T\boldsymbol{\eta}(\boldsymbol{\theta}) + \lambda\boldsymbol{I}_k)\boldsymbol{\delta} - (-\nabla\boldsymbol{\eta}(\boldsymbol{\theta})\boldsymbol{\eta}(\boldsymbol{\theta}))\right\|_2^2, \quad \text{subject to} \quad \|\boldsymbol{L\delta}\|_2^2 \leq M, \tag{2.1.22}$$

This minimization problem can be written as a CQP with linear objective function t and two ice-cream cones [47]:

$$\min_{t,\boldsymbol{\theta}} \quad t, \tag{2.1.23}$$

$$\text{subject to} \quad \left\|(\nabla\boldsymbol{\eta}(\boldsymbol{\theta})\nabla^T\boldsymbol{\eta}(\boldsymbol{\theta}) + \lambda\boldsymbol{I}_k)\boldsymbol{\delta} - (-\nabla\boldsymbol{\eta}(\boldsymbol{\theta})\boldsymbol{\eta}(\boldsymbol{\theta}))\right\|_2^2 \leq t^2, \;\; t \geq 0,$$

$$\|\boldsymbol{L\delta}\|_2^2 \leq M.$$

The general problem form for CQP is

$$\min_{\boldsymbol{x}} \;\; \boldsymbol{c}^T\boldsymbol{x}, \quad \text{subject to} \;\; \|\boldsymbol{D}_i\boldsymbol{x} - \boldsymbol{d}_i\| \leq \boldsymbol{p}_i^T\boldsymbol{x} - \boldsymbol{q}_i \;\; (i = 1, 2, ..., k). \tag{2.1.24}$$

The optimization problem (2.1.22) is such a CQP with

$$\boldsymbol{c} = (1, \boldsymbol{0}_k^T)^T, \; \boldsymbol{x} = (t, \; \boldsymbol{\delta}^T)^T, \; \boldsymbol{D}_1 = (\boldsymbol{0}_k, \bar{\boldsymbol{A}}), \; \boldsymbol{d}_1 = -\nabla\boldsymbol{\eta}(\boldsymbol{\theta})\boldsymbol{\eta}(\boldsymbol{\theta}), \; \boldsymbol{p}_1 = (1, 0, ..., 0)^T,$$

$$\boldsymbol{\delta}_1 = 0, \; \boldsymbol{D}_2 = (\boldsymbol{0}_k, \boldsymbol{L}_{k\times k}), \; \boldsymbol{d}_2 = \boldsymbol{0}_k, \; \boldsymbol{k}_2 = \boldsymbol{0}_{k+1} \text{ and } \; \boldsymbol{\delta}_2 = -\sqrt{M_1}.$$

CQP and Tikhonov regularization will be introduced in detail in the next sections of this study. There are also other approaches for solving NLR models such as the methods of steepest descent, fractional increments, Marquardt's compromise. These are modification and refinements of the Gauss-Newton iteration method [2].

2.1.3 Generalized Linear Model

Both linear and nonlinear regression models are unified under the framework of *generalized linear models* (*GLMs*). This approach is used when the assumptions of normality and constant variance are not satisfied. It enables the incorporation of nonnormal response distributions [44]. It allows the mean of a dependent variable, Y, to depend on a linear predictor through a nonlinear link function and also allows the probability distribution of Y, to be any member of an exponential family of distributions. Many widely used statistical models belong to GLMs. These include classical linear models with normal errors, logistic and probit models for binary data, and log-linear models for multinomial data and many other useful statistical models such as the Poisson, binomial, Gamma, and normal distribution can be formulated as GLMs by the selection of an appropriate link function and response probability distribution.

A GLM has the following basic structure

$$h(\mu_i) = \boldsymbol{x}_i^T \boldsymbol{\beta},$$

where $\mu_i = E(Y_i)$, h is a smooth monotonic "link function", $\boldsymbol{x}_i$ is the input variable of predictors, and $\boldsymbol{\beta}$ is a vector of an unknown parameters. In addition, a GLM usually makes the distributional assumption that the response variables Y_i are independent and can have any distribution from *exponential family density* of the form

$$Y_i \sim f_{Y_i}(y_i, \theta_i, \phi) = \exp\left\{ \frac{\theta_i y_i - b_i(\theta_i)}{a_i(\phi)} + c_i(y_i; \phi) \right\} \ (i = 1, 2, ..., N), \tag{2.1.25}$$

where a_i, b_i and c_i are arbitrary functions, ϕ is an arbitrary "scale" parameter and θ_i is called a natural parameter. It can also be obtained a general expression for the mean and variance of dependent variable Y_i using log likelihood of θ_i, $\mu_i = E(Y_i) = b_i^{'}(\theta_i)$ and $Var(Y_i) = b_i^{''}(\theta_i) \cdot a_i(\phi)$. Generally, $a_i(\phi)$ is defined as $a_i(\phi) := \phi / w_i$, and $Var(Y_i) = Var(\mu_i) \cdot \phi$, where $Var(\mu_i) := b_i^{''} / w_i$. Here, the symbol " $'$ " is used for differentiation [57].

2.1.4 Nonparametric Regression Models

The general nonparametric regression model is of the form

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

where $\boldsymbol{x} = (x_1, x_2, ..., x_k)^T$. The object of traditional regression analysis is to estimate parameters of the model, while the aim of nonparametric regression is to estimate the regression function f directly [18, 19]. In nonparametric regression, it is implicitly assumed that f is a generally smooth, continuous function and in the model the error term ϵ has zero mean and constant variance σ^2. However, in some cases, it can be nonsmooth [18, 44].

The *additive regression model*,

$$y = \beta_0 + f_1(x_1) + f_2(x_2) + ... + f_k(x_k) + \epsilon,$$

where β_0 is the unknown bias (intercept) and the partial regression functions f_j $(j = 1, 2, ..., k)$ are assumed to be smooth. Both β_0 and the functions f_j $(j = 1, 2, ..., k)$ are to be estimated from the data.

Variations of the additive regression models are the *semiparametric regression model*, in which predictor variables are "additively" separated by the unknown functions like:

$$y = \beta_0 + \beta_1 x_1 + f_2(x_2) + ... + f_k(x_k) + \epsilon,$$

or interactions of some predictor variables are expressed in unknown functions that appear as higher-dimensional terms such as:

$$y = \beta_0 + f_{12}(x_1, x_2) + f_3(x_3) + ... + f_k(x_k) + \epsilon.$$

These models are also extended to generalized nonparametric regression [18]. In addition to these nonparametric regression models, there are same other models such as *projection-pursuit regression*, *Classification and Regression Trees* (*CART*) and *Multivariate Adaptive Regression Splines* (*MARS*) [18]. In MARS, functions are multiplicative nature and nonsmooth.

The nonparametric regression models mentioned above are estimated by using three common methods of nonparametric regression. These are: (i) *kernel estimation*, (ii) *local-polynomial regression* being a generalization of kernel estimation, and (iii) *smoothing splines* [18].

2.1.5 Additive Models

Regression models, especially linear ones, are very important in many application areas. However, the traditional linear models often fail in real life since many effects are generally *nonlinear*. To characterize these effects, flexible statistical methods like *nonparametric regression* must be used (Fox, 2002) [18]. However, if the number of independent variables is large in the models, many forms of nonparametric regression do not perform well. It is also difficult to interpret nonparametric regression depending on smoothing spline estimates. To overcome these difficulties, *additive models* are used. These models estimate an additive approximation of the multivariate regression function.

If the data consist of N realizations $(\boldsymbol{x}_i, y_i)$ $(i = 1, 2, ..., N)$ of random variable y at k design values, then the *additive model (AM)* takes the following form:

$$y = \beta_0 + \sum_{j=1}^{k} f_j(x_j) + \epsilon.$$

Here, β_0 is the intercept, input data values are represented by x_j $(j = 1, 2, ..., k)$ and $\boldsymbol{x} = (x_1, x_2, ..., x_k)^T$. The functions f_j $(j = 1, 2, ..., k)$ are mostly considered to be splines, i.e., piecewise polynomial. Since they can have too strong or early asympototic towards ±∞, they do not satisfy for data fitting.

Additive models have a strong motivation as a useful data analytic tool. Each function is estimated by an algorithm proposed by Friedman and Stuetzle (1981) [9] and called *backfitting* (or *Gauss-Seidel*) algorithm. As the estimator for β_0, the arithmetic mean (average) of the output data is used:

$$\hat{\beta}_0 = \mathrm{ave}(y_i | i = 1, ..., N) := (1/N) \sum_{i=1}^{N} y_i.$$

This procedure depends on the partial residual against x_{ij}:

$$r_{ij} = y_i - \hat{\beta}_0 - \sum_{j=1}^{k} \hat{f}_\kappa(x_{i\kappa}) \ \ (j \neq \kappa),$$

and consists of estimating each smooth function by holding all other ones fixed [11].

2.1.6 Generalized Additive Models

Generalized Additive Models (GAMs) are extended forms of the additive models. They belong to modern techniques from statistical learning and they are applicable in many areas of predictions. For identifying and characterizing nonlinear regression effects, GAMs provide more flexible statistical methods. Having k covariates $x_1, x_2, ..., x_k$, comprised by the k-tuple $\boldsymbol{x} = (x_1, x_2, ..., x_k)^T$, and a response y to the input $\boldsymbol{x}$ is assumed to have exponential family density $h_y(y, \alpha, \varpi)$ with the mean $\mu = E(y|x_1, x_2, ..., x_k)$ linked to the predictors through a *link function* $G = \frac{P_r(y-1|x)}{P_r(y-1|x)}$. Examples of link functions are *logit* link function, the *probit* link function and *identity* link function. In addition, α is called the natural parameter and ϖ is the dispersion parameter.

In a regression setting, GAMs have the following form:

$$\eta(\boldsymbol{x}) = G(\mu) = \beta_0 + \sum_{j=1}^{k} f_j(x_j),$$

where the functions f_j are unspecified ("nonparametric") and $\boldsymbol{\chi} := (\beta_0, f_1, ..., f_k)^T$ is the unknown entire parameter vector to be estimated. The incorporation of β_0 as an average outcome allows to assume $E(f_j(x_j)) = 0$ $(j = 1, 2, ..., k)$ [25].

2.2 Background

2.2.1 Tikhonov Regularization

Problems whose solution do not exist, or which is not unique or not stable under perturbations on data are called *ill-posed* [2]. *Tikhonov regularization* belongs to the most commonly used methods of making these problems well-posed (regular or stable) in some fields, it is also known as *ridge regression.* The Tikhonov solution can be expressed quite easily in terms of *singular value decomposition* (SVD) of the coefficient matrix $\boldsymbol{X}$ of a regarded linear systems of equations

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

For a general linear LS problem there may be infinitely many solutions. If we take into account that the data contain noise, in that situation, generally, noisy data points cannot be fitted exactly. Then, it becomes evident that there may be many solutions which can adequately fit the data in the sense that the Euclidean distance $\|\boldsymbol{X\beta} - \boldsymbol{y}\|_2$ is the smallest. The *discrepancy principle* [2] can be used to regularize the solution of a discrete ill-posed problem based on the assumption that a reasonable level for $\delta = \|\boldsymbol{X\beta} - \boldsymbol{y}\|_2$ is known. If the norm of the error in the data or the norm of the solution of the error-free problem is available, a suitable value of the parameter for Tikhonov regularization is considered and computed. Under the discrepancy principle, all solutions with $\|\boldsymbol{X\beta} - \boldsymbol{y}\|_2 \leq \ \delta$ are considered, and we select the one from these solutions such that it minimizes the norm of $\boldsymbol{\beta}$,

$$\min_{\boldsymbol{\beta}} \ \ \|\boldsymbol{\beta}\|_2 \ \ \text{such that} \ \ \|\boldsymbol{X\beta} - \boldsymbol{y}\|_2 \leq \delta. \tag{2.2.26}$$

In this minimization problem, any nonzero feature that appears in the regularized solution increases the norm of $\boldsymbol{\beta}$. These features exist in the solution because they are necessary to fit the data. Therefore the mimimum of $\|\boldsymbol{\beta}\|_2$ should ensure that unneeded features do not appear in the regularized solution. While δ increases, the set of feasible models expands, and the minimum value of $\|\boldsymbol{\beta}\|_2$ decreases. It is possible to trace out this minimization problem by considering problems of the form

$$\min_{\boldsymbol{\beta}} \quad \|\boldsymbol{X\beta} - \boldsymbol{y}\|_2 \quad \text{such that} \quad \|\boldsymbol{\beta}\|_2 \leq \epsilon. \tag{2.2.27}$$

As ϵ decreases, the set of all feasible solutions becomes smaller, and the minimum value of $\|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2$ increases. In the second optimization problem, it is desired to select the minimum norm solution among those parameter vectors which adequately fit the data, because any important nonzero feature that appears in the regularized solution must not be neglected to fit the data and unimportant data must be removed by the regularization. Yet, there is also a third option in which we consider a dampened LS problem of the form

$$\min_{\boldsymbol{\beta}} \quad \|\boldsymbol{X\beta} - \boldsymbol{y}\|_2^2 + \varphi^2 \|\boldsymbol{\beta}\|_2^2 \,, \tag{2.2.28}$$

arising when we apply the method of Lagrange multipliers to problem (2.2.27). Here, $\lambda = \varphi^2$ is the tradeoff parameter between the first and the second part.

These three problems have the same solution for some appropriate choice of the values $\delta, \epsilon, \varphi$ [2].

When plotted on a log-log scale, the curve of optimal values of $\|\boldsymbol{\beta}\|_2^2$ versus $\|\boldsymbol{X\beta} - \boldsymbol{y}\|_2^2$ often has a characteristic L shape. This occurs because $\|\boldsymbol{\beta}\|_2^2$ is a strictly decreasing function of φ and $\|\boldsymbol{X\beta} - \boldsymbol{y}\|_2^2$ is a strictly increasing function of φ. The sharpness of the "corner" varies from problem to problem but it is often well-defined. Because of this, the curve is called an ***L-curve*** [24].

Above, different kinds of Tikhonov regularization represented by minimization problems are discussed. These problems can be solved using the SVD [1]. However, in many situation, it is preferred to obtain a solution which minimizes some other measure of $\boldsymbol{\beta}$, such as the norm of some 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 discrete points enumerated by i and i+1. These difference quotients approximate first- and second- order derivates; altogether they are comprised by products $\boldsymbol{L\beta}$ of $\boldsymbol{\beta}$ with matrices $\boldsymbol{L}$ that represent the discrete differential operators of first- and second- order, respectively. These matrices are of a band structure with values -1, 1 and 1, -2, 1 on the band [1]. Herewith, the optimization problem takes the following from

$$\min \ \|\boldsymbol{X\beta} - \boldsymbol{y}\|_2^2 + \varphi^2 \|\boldsymbol{L\beta}\|_2^2 . \tag{2.2.29}$$

The optimization problem given in (2.2.28) is a special realization of optimization problem of (2.2.29), namely, with $\boldsymbol{L} = \boldsymbol{I}$. Generally, (2.2.29) comprises higher-order Tikhonov regularization problem which can be solved using *generalized singular value decomposition* (*GSVD*) [1]. In many situations, to reach a solution that minimizes some other measures of $\boldsymbol{\beta}$, such as the norm of the first or second derivative, is preferred. In the *first-order Tikhonov regularization*, for solving the dampened LS problem, the following $\boldsymbol{L}$ matrix is used:

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

In the *second-order Tikhonov regularization*,

$$\boldsymbol{L} = \begin{bmatrix} 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & \mathbf{0} & \\ & & & \cdots & & & \\ & \mathbf{0} & & & 1 & -2 & 1 \\ & & & & & 1 & -2 & 1 \end{bmatrix}$$

is used. Here, $\boldsymbol{L\beta}$ is a finite-difference approximation proportional to the second derivative of $\boldsymbol{\beta}$, and inclusion of $\|\boldsymbol{L\beta}\|_2^2$ into the joint minimization penalizes solutions that are like a second derivative sense. In our study, the matrix $\boldsymbol{L}$ has a different type (cf. Section 3). For all of these matrices and problems, *MATLAB Regularization Toolbox* can be used [2].

2.2.2 Conic Quadratic Programming

Some “generic” group of conic problems are of special interest both for theory and applications. The cones in these problems are simple enough; therefore, it can be described explicitly the dual cone, due to the general duality machinery becoming algorithmic as in the linear programming (LP) case. In addition, in many cases, this algorithmic duality machinery facilitates to understand the original model better [47]. The well-known examples of generic conic problem are LP, semidefinite programs and conic quadratic programming.

A generic *conic problem* can be written as follows:

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

associated with a cone $\boldsymbol{K}$ given as a direct product of m cones, each of them being either a semidefinite or a second-order (Lorentz) cone:

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

A *conic quadratic (CQ)* problem is a conic problem which can be shown as follows [47]:

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

for which the cone $\boldsymbol{K}$ is the direct product of several ice-cream cones. In case of *CQP*, there are no semidefinite factors $\boldsymbol{S}_{+}^{k_i}$; therefore, $\boldsymbol{K}$ can be represented in the following way:

$$\boldsymbol{K} := \boldsymbol{L}^{k_1} \times ... \times \boldsymbol{L}^{k_r} \ \subseteq \ \boldsymbol{E},$$

and the k-dimensional ice-cream (second-order, Lorenz) cone $\boldsymbol{L}^k$ is as follows:

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

In general, a CQ problem is an optimization problem with linear objective function and finitely many “ice-cream constraints”

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

Therefore, a CQ problem can be written as [47]:

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

If we subdivide the data matrix, $[\boldsymbol{A}_i, \boldsymbol{b}_i]$, as follows:

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

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

$$\min_{\boldsymbol{x}} \ \boldsymbol{c}^T \boldsymbol{x}, \quad \text{subject to} \quad \|\boldsymbol{D}_i \boldsymbol{x} - \boldsymbol{d}_i\|_2 \leq \boldsymbol{p}_i^T \boldsymbol{x} - \boldsymbol{q}_i \quad (i = 1, 2, ..., r); \tag{2.2.30}$$

This is the most explicit form that is used. 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 $\boldsymbol{q}_i$ are real numbers [47].

On Solution Methods for Conic Quadratic Programming

For solving convex optimization problems like LP, semidefinite programming, geometric programming and, in particular, CQ problems, all of them being very important in DM, classical *polynomial time algorithms* can be applied. However, these algorithms have some disadvantages since they use local information on the objective function and the constraints. For this reason, to solve “well-structured” convex problems such as the aforementioned ones and, in particular, CQ problems, there are *interior point methods* (*IPM*s) [48] which were firstly introduced by Karmakar (1984) [29]. Let us consider an optimization problem given by

$$\min_{\boldsymbol{x}} \ \boldsymbol{c}^T \boldsymbol{x}, \quad \text{where} \quad \boldsymbol{x} \in \boldsymbol{\Omega} \subseteq \mathbb{R}^n. \tag{2.2.31}$$

*IPM*s classically base on the interior points of the feasible set Ω, which is assumed to be closed and convex. Then, an *interior penalty function (barrier)* $\boldsymbol{F}(\boldsymbol{x})$ is chosen, well defined (smooth and strongly convex) in the interior of $\boldsymbol{\Omega}$ and "blowing up" as a sequences from the interior int $\boldsymbol{\Omega}$ approches a boundary point of $\boldsymbol{\Omega}$:

$$\boldsymbol{x}_k \in \text{int } \boldsymbol{\Omega} \; (k \in \mathbb{N}_0), \lim_{k\to\infty} \boldsymbol{x}_k \in \partial\boldsymbol{\Omega} \;\Rightarrow\; \boldsymbol{F}(\boldsymbol{x}_k) \to \infty \; (k \to \infty). \tag{2.2.32}$$

Now, we consider one parametric family of functions generated by our objective an interior *penalty function* $\boldsymbol{F}_t(\boldsymbol{x}) := t\boldsymbol{c}^T\boldsymbol{x} + \boldsymbol{F}(x) : \text{int } \boldsymbol{\Omega} \to \mathbb{R}$. Here, the *penalty parameter* t is assumed to be nonnegative. Under mild regularity assumptions,

- every function $\boldsymbol{F}_t(\cdot)$ attains 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.

These algorithms have the advantage of employing the structure of the problem, of allowing better complexity bounds for the indicated generic problems and exhibiting a much better practical performance. For closer details about these IPMs, we refer to [8]. In the so-called *primal-dual IPMs*, both the primal and the dual problems and their variables are regarded (cf. Section 3), the joint optimality conditions perturbed, parametrically solved and followed towards a solution along a *central path.*

Complexity of Conic Quadratic Programming

A program from conic quadratic optimization:

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

where the matrices $\boldsymbol{D}_i$ are of the type $k_i \times k$, $\boldsymbol{p}_i$, $\boldsymbol{x} \in \mathbb{R}^k$ and $\boldsymbol{d}_i \in \mathbb{R}^{k_i}$. The data of (2.2.30) can be presented in the way [47, 52]

$$\text{Data}((2.2.30)) := [r; k; k_1; ...; k_r; \boldsymbol{c}; \boldsymbol{D}_1, \boldsymbol{d}_1, \boldsymbol{p}_1, \boldsymbol{q}_1; ..., \boldsymbol{D}_k, \boldsymbol{d}_k, \boldsymbol{p}_k, \boldsymbol{q}_k; t]$$

and

$$\text{Size}((2.2.30)) \; := \text{Data}((2.2.30)) \; := \left(r + \sum_{i=1}^{r} k_i\right)(k+1) + k + 3.$$

The arithmetic complexity of ϵ-solution is as follows:

$$\text{Compl}((2.2.30), \epsilon) \; := O(1)(r+1)^{1/2} k \left(k^2 + r + \sum_{i=1}^{r} k_i^2\right) \text{Digits}((2.2.30), \epsilon),$$

where

$$\text{Digits}((2.2.30), \epsilon) := \ln\left(\left(\text{Size}((2.2.30)) + \|\text{Data}((2.2.30))\|_1 \epsilon^2\right)/\epsilon\right),$$

is defined as the number of accuracy digits in an ϵ-solution to (2.2.30), referring to the sum (or l_1) norm [52]. Please note that complexity is often and differently is used in this thesis. Here, definition of complexity is given by Arkadi Nemirovski [47].

2.2.3 MOSEK

The MOSEK, which is a MATLAB add-on, is an optimization tool for solving large-scale mathematical optimization problems [41]. MOSEK provides solvers for optimization problems of the following types:

- linear problems,
- CQ problems,
- convex quadratic problems,
- general convex problems,
- mixed integer problems.

MOSEK has technical advantages [41]. For example, it can solve large-scale problems. The problem size is only limited by the available memory. MOSEK has an interior-point optimizer with basis identification. For its excellent *speed* and *stability*, the MOSEK interior-point optimizer is well known. Fine tuning of algorithmic parameters to obtain good performance is needed. The software exploits problem sparsity and structure automatically to obtain the best possible efficiency. MOSEK also has both primal and dual simplex optimizers for LP. It has an efficient presolver for reducing problem size before optimization. Moreover, MOSEK can also deal with primal and dual infeasible problems in a systematic way. It can read and write industry standard formats such as MPS, LP and XML, and includes tools for infeasibility diagnosis and repair. Finally, it corrects sensitivity analysis for linear problems [41].

MOSEK optimization tools also consist of interfaces that makes it easy to deploy the functionality of MOSEK from programming languages such as C, C++, MATLAB Toolbox, Java, NET, and Python [41].

2.2.4 Multiobjective Optimization

In classical optimization problems, there is a single objective function and the goal is to find a solution that optimizes the objective function value. However, many real life problems have many objectives and decisions should be made by considering these objective functions simultaneously. Normally different objectives are conflicting with each other and a solution that fulfills well in one objective will not fulfill as well in other objectives. There are many solutions that do not perform well each other in all objectives. It does not become clear which of these solutions are better until the *decision maker (DM)* evaluates them.

A *multiobjective problem (MOP)* can be stated as follows

$$\min \quad \boldsymbol{C}\boldsymbol{x} = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), ..., f_p(\boldsymbol{x}))^T \quad \text{subject to} \quad \boldsymbol{x} \in \boldsymbol{X},$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is a feasible solution and $\boldsymbol{X}$ is the set of all feasible solutions. In this problem, there are p objective functions to be minimized and $\boldsymbol{C}$ is a $(p \times n)$-matrix. The qth row of $\boldsymbol{C}$ corresponds to the qth objective function, $f_q(\boldsymbol{x})$.

The point $\boldsymbol{y} = (y_1, y_2, ..., y_p)^T \in \mathbb{R}^p$ such that $\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x}$ is the outcome of the solution $\boldsymbol{x} \in \boldsymbol{X}$. The sets $\boldsymbol{X}$ is called *decision space* and $\boldsymbol{Y} = \{\boldsymbol{y} \in \mathbb{R}^p | \boldsymbol{y} = \boldsymbol{C}\boldsymbol{x}, \boldsymbol{x} \in \boldsymbol{X}\}$

is called the *objective (criterion) space*.

A point $\boldsymbol{y}$ is called to *dominate* point $\boldsymbol{y}^{'}$ if and only if $y_q \leq y_q^{'}$ for all q and $y_q < y_q^{'}$ for at least one q. If $y_q < y_q^{'}$ for all q then $\boldsymbol{y}$ is called to *strictly dominate* $\boldsymbol{y}^{'}$. If there exists no $\boldsymbol{y}^{'} \in \boldsymbol{Y}$ such that $\boldsymbol{y}^{'}$ dominates $\boldsymbol{y}$, then $\boldsymbol{y}$ is called *nondominated*. A point $\boldsymbol{y}$ is said to be *weakly nondominated* if and only if there is no point $\boldsymbol{y}^{'} \in Y$ such that $y_q > y_q^{'}$ for all q. The set of weakly nondominated points consists of all nondominated points and some special dominated points.

The terms *dominance* and *efficiency* are equivalent of each other in the objective and decision spaces, respectively. A solution $\boldsymbol{x}$ is said to be *efficient* (nondominated) if and only if $\boldsymbol{y} = \boldsymbol{Cx}$. In other words, a feasible solution to an MOP is efficient (nondominated) if no other feasible solution is at least as good for every objective and strictly better in one. A solution $\boldsymbol{x} \in \boldsymbol{X}$ is *inefficient* (dominated) if and only if $\boldsymbol{y} = \boldsymbol{Cx}$. A solution $\boldsymbol{x} \in \boldsymbol{X}$ is *weakly efficient* if and only if $\boldsymbol{y} = \boldsymbol{Cx}$ is weakly nondominated. We refer to Steuer (1986) [50] for an overview of the multiple criteria optimization theory, methodology and applications. In Figure 2.1, while y_1, y_2, y_3, y_4, y_5, y_6 and y_7 are nondominated points, y_8, y_9 and y_{10} are dominated points.

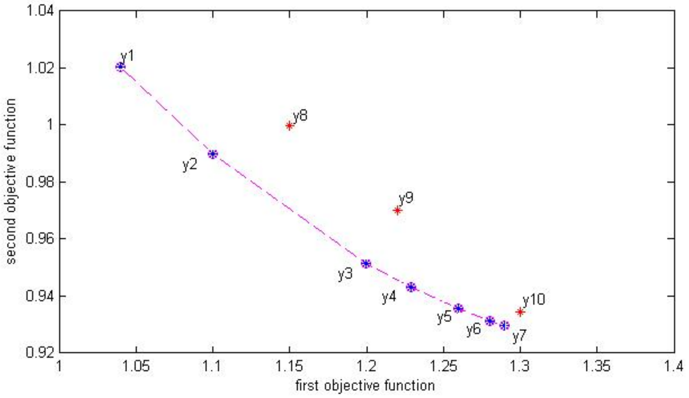

Figure 2.1: Efficient frontier with dominated and nondominated points.

CHAPTER 3

METHODS

3.1 Multivariate Adaptive Regression Splines

3.1.1 Introduction into MARS

Multivariate Adaptive Regression Splines (MARS), developed in 1991 by the well-known physicist and statistician Jerome Friedman (Friedman 1991) [20], is a novel and powerful adaptive regression method from statistical learning. MARS essentially constructs flexible models by introducing piecewise linear regressions. The 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. In other words, the slope of each regression line is allowed to change from one interval to another one with the condition that there is a "knot" defined in between. Therefore, splines are used rather than normal straight lines if there is a need. Predictors which are included in the final model together with their respective knots are found via a fast but intensive search procedure. Other than examining each individual predictor, MARS also automatically searches for interactions between them in any degree [15].

The MARS method generates a model in a two-stage process: forward and backward. In the first stage, MARS constructs an extra large number of basis functions (BFs), which deliberately overfit the data. These BFs represent distinct intervals of every predictor divided by knots, and in an intensive search, every possible knot location is tested. The MARS model is actually, in each dimension, a linear summation of certain BFs, and interactions among them if needed. Then, some of the BFs are removed as they contribute least to the overall performance. Therefore, the forward construction initially includes many incorrect terms. In the backward pruning step, these erroneous terms are eventually excluded. Thus, the backward step reduces the "complexity" of the model without degrading the fit to the data. By allowing arbitrary shapes of BFs and their interactions, MARS has the capacity of reliably tracking very complex data structures that often hide in high dimensions [15].

In recent years, MARS has been successfully applied in many areas of science and technology such as predicting object-oriented software maintainability [65], species distribuions from presence-only data [17], gastro-intestinal absorption of drugs [13], wastewater treatment [54], and predicting Acute Myocardial Infaction (AMI) mortality [3]. In addition, MARS has applications in speech modeling [23], mobile radio channels prediction [33], intrusion detection in information systems securty [42], global optimum in structural design [11], determining the relationship between biological activities and HIV reverse transcriptase inhibitors [59], and detecting disease risk relationship differences among subgroups [62]. MARS has also been employed to simulate soil temparature [61], and pesticide transport in soils [63], to detect genotype-environment interaction [62], to examine the impact of information technology investment an productivity [31], to model the relationship between retention indices and molecular descriptors of alkanes [60]. Moreover, MARS is used for DM on breast cancer pattern [8], credit scoring [36], and foreign exchange rate prediction. In Chapter 5, we will also indicate application in the financial sector.

3.1.2 MARS Word by Word

The first word "*multivariate*" expresses that MARS is able to deal with multidimensional data, examine individual features and possible interactions among them.

The second word "*adaptive*" simply means selective. MARS automatically deletes certain number of predictors if they do not contribute enough to the performance of the final model. Sometimes, this is also called *feature selection* [15].

The next word "*regression*" refers to the normally used statistical term, which is often represented as a general prediction function:

$$y = \beta_0 + \sum_{j=1}^{p} \beta_j x_j,$$

where y is the target value, β_0 is the constant term, β_j are the set of coefficients, x_j are the predictor values.

The last word "*splines*" indicates a wide class of piecewise defined functions that are used in applications requiring data interpolation and/or smoothing. In order to develop a spline, the original space is divided into a conventional number of regions. The boundary between regions is known as a knot. By obtaining a sufficient number of knots, any shape can be well approximated [15].

3.1.3 The Approach

MARS is a nonparametric modeling approach versus the well-known global parametric modeling methods such as linear regression. In global and parametric approaches, a global parametric function which is fitted to the available data is used to approximate the underlying relationship between a target variable and a set of explanatory variables. While global parametric modeling methods are relatively easy to improve and interpret, they have a limited flexibility and work well only in the case where the true underlying relationship is close to the pre-specified approximated function in the model. To overcome the shortcomings of global parametric approaches, nonparametric models are developed locally over specific subregions of the data; the data is searched for optimum number of subregions and a simple function is optimally fit to the realizations in each subregion [64].

Let y be the dependent response, which can be continuous or binary, and let $\boldsymbol{x} = (x_1, x_2, ..., x_p)^T$ be a vector of predictor variables. The true relationship between y and $\boldsymbol{x}$ can be described as

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

where f is an unknown function, and the error term ϵ is a white noise. The most fundamental elements of MARS are BFs, as they are used to build the most essential piecewise linear regression. The following two functions are *truncated functions*, where $x \in \mathbb{R}$ [9]:

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

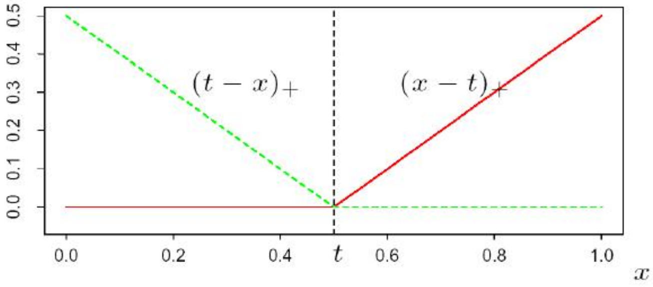

Figure 3.1: The BFs $(x-t)_+$ and $(t-x)_+$ used by MARS [25].

For both forms, let us consider a functional value x^*. In the first form, x^* is set to 0 for all values of x up to some threshold value t and x^* is equal to $x-t$ for all values of x greater than t. In the second form, x^* is set to 0 for all values of x greater than some threshold value t and x^* is equal to $t-x$ for all values of x less than t [1]. Each function is piecewise linear, with a knot at the value t. These trancated functions are linear nonsmooth splines. The two functions are called a *reflected pair*. The idea is to form reflected pairs for each input x_j with knots at each observed value $x_{i,j}$ $(i = 1, 2, ..., N \; ; \; j = 1, 2, ..., p)$ of that input. Therefore, *the collection of BFs* can be written as:

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

If all of the input values are distinct, there are $2Np$ BFs altogether. It should be noted that each BF depends only on a single x_j [25].

The usual method for generalizing spline fitting in higher dimensions is to employ BF that are the tensor products of univariate spline functions. Therefore, multivariate spline BFs take the following form:

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

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 $s_{km} \in \{\pm 1\}$.

The model-building strategy is similar to a forward stepwise linear regression; however, instead of using the original inputs, it is allowed to use functions from the set C and their products. Thus, the model has the form

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

where M is the set of BFs in the current model and c_0 is the intercept [15].

Given some choices for the B_m, the coefficients c_m are estimated by minimizing the RSS, that is also made in standard linear regression. To generate the model, the most important issue is the construction of the functions B_m. The construction of the model starts with only the constant function $B_0(\boldsymbol{x}) = 1$, and all functions in the set C are candidate functions.

The following functions are possible forms of BFs $B_m(\boldsymbol{x})$ [32]:

- 1,

- x_j,

- $(x_j - t_k)_+$,

- $x_l x_j$,

- $(x_j - t_k)_+ x_l$,

- $(x_j - t_k)_+ (x_l - t_h)_+$.

In the MARS algorithm, each BF can not include the same input variables. Thus, above BFs which are obtained from two multiplied BFs use different input variables such as x_j, x_l and t_k, t_h are their corresponding knots. At each stage, a new BF pair is all products of a function $B_m(\boldsymbol{x})$, in the model set $\mathcal{M}$ with one of the reflected pairs in C. Then, the term below is added to the model set $\mathcal{M}$:

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

this produces the largest decrease in training error. Here, $\hat{C}_{M+1}$ and $\hat{C}_{M+2}$ are coefficients estimated by LS, along with all the other $M+1$ coefficients in the model. The process is continued until the model set $\mathcal{M}$ contains some preset maximum number of terms. This process means that the model set $\mathcal{M}$ is *iteratively (recursively)* built up [25].

For example, the following BFs are possible candidates [32]:

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

At the end of this process, a large model is obtained. This model typically overfits the data, and then a backward deletion procedure is started. In this pruning step, the term whose removal causes the smallest increase in residual squared error is deleted from the model at each stage. This process produces an estimated best model $\hat{f}_M$ of each size (number of terms) M. In order to estimate the optimal value of M, for computational savings, the MARS procedure uses *generalized cross-validation.* This criterion, also known as lock of fit criterion, is defined as [20]

$$\begin{aligned} LOF(\hat{f}_M) &= GCV_{Friedman} := \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{f}_M(x_i))^2/(1 - \mathbf{C}(M)/N)^2, \\ \mathbf{C}(M) &= \mathrm{trace}(B(B^TB)^{-1}B^T) + 1, \end{aligned}$$

where there are N number of data samples, $\mathbf{C}(M)$ is the cost penalty measures of a model containing M BFs, and B is an $(M \times N)$-matrix. Indeed, $\mathbf{C}(M)$ is the number of parameters being fit. The numerator is the common RSS, which is penalized by the denominator. This denominator accounts for the increasing variance in the case of increasing the model complexity.

There are different representations of $\mathbf{C}(M)$. One of them is: $\mathbf{C}(M) = r + cK$. Here, r is linearly independent BFs in the model, and K represents the number of knots which are selected in the forward process and, the quantity c shows a cost for each BF optimization and generally, $c = 3$ [25]. If the model is additive, a penalty of $c = 2$ is used. A smaller $\mathbf{C}(M)$ generates a larger model with more BFs, a larger $\mathbf{C}(M)$ creates a smaller model with less BFs. Using lack of fit criteria, the best model along the backward sequence that minimizes $GCV_{Friedman}$ is reached [15, 25].

MARS uses piecewise linear BFs, and it has a particular model strategy. A key property of the piecewise linear BFs is their ability to operate locally; they are zero over part of their range. When they are multiplied together, as in Figure 3.2, the result is nonzero only over the small part of the factor space where both component functions are nonzero. As a result, the regression surface is built up by using nonzero components locally - only where they are needed. On the other hand, the use of other BFs such as polynomials, would produce a nonzero product everywhere, and would not work as well.

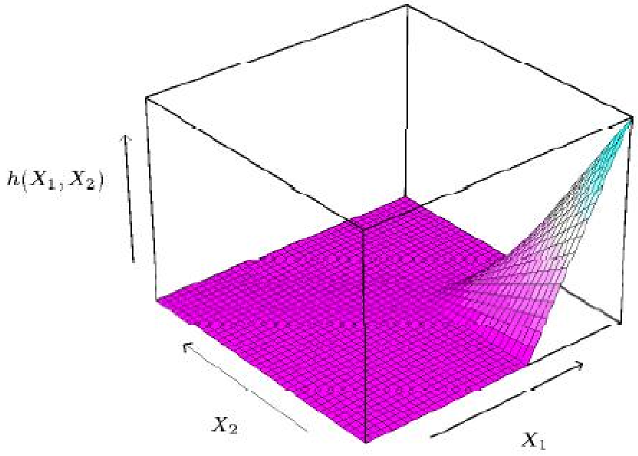

Figure 3.2: Two-way interaction basis fnctions [25].

In Figure 3.2, the function $h(X_1, X_2) = (X_1 - x_{51})_+(x_{72} - X_2)_+$ is resulting from multiplication of two piecewise linear MARS BFs. The forward modeling strategy in MARS is hierarchical. Since multiway products are built up from products involving terms already in the model. A high-order interaction only exists if some of its lower order components exist as well. For example, a four-way product can only be added to the model if one of its three-way components is already in the model. This is a reasonable working assumption and avoids the search over an exponentially growing space of alternatives [25].

There is one limitation put on the formation of MARS model terms: each input can appear at most once in a product. This prevents the formation of higher-order powers of an input, which increases or decreases too sharply near the boundaries of the factor space. Higher order powers can be approximated in a more stable way with piecewise linear functions.

Enabling to set an upper limit on the order of interaction is a useful option in the MARS procedure. For example, choosing two as a limit allows pairwise products of piecewise linear functions but not three- or higherway products. This can be helpful to interpret the final model. An upper limit of one results in an additive model [25].

3.1.4 MARS Software Overview

The MARS models in this study are fitted using *MARS (Version 2, Salford Systems, San Diego, Calif., USA)*. 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; the maximum number of interactions can be more than the degree of two (2-way interaction). The MARS package developed by *Salford Systems* is avalible at [9]. It is a well designed piece of software that implements MARS technique with *friendly graphical user interface.*

Penalty on added variables results in MARS to prefer to reusing of variables already in the model over adding new variables. As the penalty increases, MARS automatically generates new knots in existing variables of generates interaction terms in involving existing variables [40]. Although the minimum number of observations between knots is very useful for continuous variables, it is not useful for discrete variables with only a few distinct values. By default, MARS allows to create a knot at every distinct observed data value, which allows the MARS regression to change slope or direction anywhere and as often as the data dictate [40]. The search speed parameter can be set one to five and its default value is 4. It is suggested by MARS (2001) that the search speed parameter be set to four for real-world problems, and the use of search speed parameter of three, four, or five do not change the models [58].

After setting all the parameters correctly, MARS will yield the final model in a rather short time. There are lots of result evaluations provided: R^2, Mean Square Error (MSE), ANOVA, f-value, t-value, p-value, RSS, variable importance measurement assessed by observing the decrease in performance when one is removed from the model, etc.. Moreover, various result illustrations are also available: the final model consists of a number of specific BFs, gain and lift charts, curve and surface plots, etc.. In addition, a previously yielded model can be applied to a new dataset. Therefore, the MARS package is considered as very powerful as it takes in various preferences, criteria, constraints, and control parameters for the user [15].

3.1.5 MARS vs. Other Algorithms

The explanation given till now provides a complete picture of how MARS works. Although it is an extension of *Classification and Regression Trees CART*, MARS is normally not presented in decision tree (DT) format. The similarity is mainly on the partitioning of intervals, where two symmetric BFs are created at the knot location. However, MARS differs from decision tree techniques such as *CART* and *CHAID* since it assigns a coefficient (a slope) to each part. In other words, while DT techniques use step functions to model the dependent variable and this causes a discontinuous models, MARS uses piecewise linear functions which are continuous. This produces continuous models which provides a more effective way to model nonlinearities (De Veaux et al., 1993) [12].

MARS is a flexible regression technique that uses a modified recursive partitioning strategy for simplifying high-dimensional problems. Although *recursive partitioning regression (RPR)* is a powerful method, it has some shortcomings such as discontinuity at the subregion boundaries. MARS overcomes these limitations [64].

When compared with other typical modeling techniques such as *multivariate linear regression* models, *regression tree* models, *support vector* results, MARS has a better prediction accuracy. Moreover, the *artificial neural network* has limitations like a long training process, interpretation difficulties of the model and application in some problems. MARS has also the capability to overcome these problems [65].

Conventional statistical methods such as regression can handle interactions terms, but this is not easy in practice since it requires trying many combinations of the variables in the data set. In fact, it can be computationally infeasible. MARS automatically looks for suitable interactions between independent variables, which makes it in particular preferable whenever there is a large number of interacting variables.

The MARS methodology has a risk of overfitting because of very exhaustive search that is conducted to identify nonlinearities and interactions. There are protections against overfitting such as setting a lower maximum number of BFs and a higher "cost" per knot [20].

In conclusion, although MARS has this limitation, it offers a number of advantages. For example, MARS is capable of identifying a relatively small number of predictor variables which are complex transformations of initial variables. It also enables to discover nonlinearities that may exist in the relationship between response and predictor variables. Another advantage of MARS is that it identifies interactions, and also produces graphs that help visualize and understand interactions [14].

In the next section, we will present an own contribution to the theory of MARS by the use of modern continuous optimization. In fact, while in the explanations given in this section different elements of a *model-free* approach were used, especially, via GCV in the backward stepwise algorithm, we are going now to turn to an integrated *model-based* approch. For this one, continuous optimization will serve us, in the form of a penalized optimization problem and, then, a conic quadratic optimization problem. By this we will arrive at a new, alternative version of MARS, called *C-MARS* ("***C***" standing for ***conic***, but also reminding us of ***continuous*** and ***convex***).

3.2 Conic Multivariate Adaptive Regression Splines

3.2.1 Multivariate Adaptive Regression Splines Method Revisited by Tikhonov Regularization

Multivariate Adaptive Regression Splines (*MARS*) is a method to estimate general functions of high dimensional arguments given sparse data [20]; it has an increasing number of applications in many areas of science, economy and technology. At the same time it is a research challenge, to which this present thesis wishes to contribute, especially, by means of using continuous optimization theory. We shall mostly refer to a regression formulation, but also classification will become addressed. The finitely many data underlying may base on different types of experiments, questionnaires, records or a preprocessing of information by clustering, etc.; they can also be obtained with different kinds of technologies.

MARS is an *adaptive* procedure because the selection of BFs is data-based and specific to the problem at hand. This algorithm is a nonparametric regression procedure that makes no specific assumption about the underlying functional relationship between the dependent and indepentent variables. It is very useful for high dimensional problems and shows a great promise for fitting nonlinear multivariate functions. A special advantage of MARS lies in its ability to estimate the contributions of the BFs so that both the additive and the interactive effects of the predictors are allowed to determine the response variable.

For this model an algorithm was proposed by Friedman (1991) [20] as a flexible approach to high dimensional nonparametric regression, based on a modified recursive partitioning methodology. The above explanations are given in detail in the previous section. In this section, we introduce a modified version of MARS called *Conic Multivariate Adaptive Regression Splines (C-MARS)*. Here, "***C***" means not only the word ***conic*** but also ***convex*** and **continuous**. For our explanations on C-MARS, we prefer the following notation for the piecewise linear BFs:

$$c^{+}(x,\tau) = [+(x-\tau)]_{+}, \quad c^{-}(x,\tau) = [-(x-\tau)]_{+}, \tag{3.2.1}$$

where $[q]_+ := \max\{0, q\}$ and τ is an univariate knot. Each function is piecewise linear, with a knot at the value τ, and it is called a *reflected pair*. For a visualization see Figure 3.3:

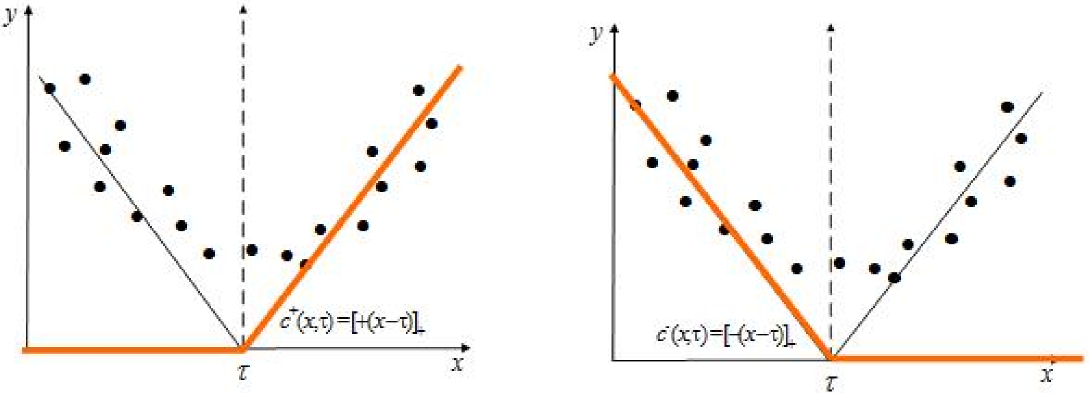

Figure 3.3: Basic elements in the regression with MARS [56].

The points in this figure illustrate the data $(\bar{\boldsymbol{x}}_i, \bar{y}_i)$ $(i = 1, 2, ..., N)$ composed of a p-dimensional input specification of the variable $\boldsymbol{x}$ and the corresponding one-dimensional response which specify the variable y.

Let us consider the following general model on the relation between input and response that we introduced in the Subsection 3.1.3:

$$Y = f(\boldsymbol{X}) + \epsilon, \tag{3.2.2}$$

where Y is a response variable, $\boldsymbol{X} = (X_1, X_2, ..., X_p)^T$ is a vector of predictor variables and ϵ is an additive stochastic component which is assumed to have zero mean and finite variance. The goal is to construct reflected pairs for each input X_j $(j = 1, 2, ..., p)$ with p-dimensional knots $\boldsymbol{\tau}_i = (\tau_{i,1}, \tau_{i,2}, ..., \tau_{i,p})^T$ at or just nearby each input data vectors $\bar{\boldsymbol{x}}_i = (\bar{x}_{i,1}, \bar{x}_{i,2}, ..., \bar{x}_{i,p})^T$ of that input $(i = 1, 2, ..., N)$. Such a nearby placement means a slight modification made in this study. In the previous section, the knots' values are presented as equal to input values. Indeed, we may without loss of generality assume that $\tau_{i,j} \neq \bar{x}_{i,j}$ for all i and j, in order to prevent from nondifferentiability in our optimization problem later on. This assumption is also implied into Figure 3.3. Actually, we could even choose the knots $\tau_{i,j}$ more far away from the input values $\bar{x}_{i,j}$ if any such a position promises a better data fitting.

In compact matrix notation, $\tau_{i,j}$ and $\bar{x}_{i,j}$ can be comprised as follows:

$$(\tau_{i,j})_{\substack{i=1,2,\ldots,N\\ j=1,2,\ldots,p}} = \begin{bmatrix} \tau_{1,1} & \tau_{1,2} & \cdots & \tau_{1,p} \\ \tau_{2,1} & \tau_{2,2} & \cdots & \tau_{2,p} \\ \vdots & \ddots & \ddots & \vdots \\ \tau_{N,1} & \tau_{N,2} & \cdots & \tau_{N,p} \end{bmatrix} \quad , \quad (\bar{x}_{i,j})_{\substack{i=1,2,\ldots,N\\ j=1,2,\ldots,p}} = \begin{bmatrix} \tau_{1,1} & \tau_{1,2} & \cdots & \tau_{1,p} \\ \tau_{2,1} & \tau_{2,2} & \cdots & \tau_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ \tau_{N,1} & \tau_{N,2} & \cdots & \tau_{N,p} \end{bmatrix} .$$

After these preparations for C-MARS, let use below formulation for the set of BFs:

$$\wp := \left\{ (X_j - \tau)_+, (\tau - X_j)_+ \mid \tau \in \{x_{1,j}, x_{2,j}, ..., x_{N,j}\}, j \in \{1, 2, ..., p\} \right\}. \tag{3.2.3}$$

If all of the input values are distinct, there are $2Np$ BFs altogether. Thus, we can represent $f(\boldsymbol{X})$ by a linear combination which is successively built up by the set $\wp$ and with the intercept θ_0 such that (3.2.2) takes the form

$$Y = \theta_0 + \sum_{m=1}^{M} \theta_m \boldsymbol{\psi}_m(\boldsymbol{X}) + \epsilon. \tag{3.2.4}$$

Here, $\boldsymbol{\psi}_m$ $(m = 1, 2, .., M)$ represents a BF from $\wp$ or products of two or more such functions, ψ_m is taken from a set of M linearly independent basis elements, and θ_m is the unknown coefficient for the mth BF $(m = 1, 2, .., M)$ for the constant 1, m equals to zero. A set of eligible knots $\tau_{i,j}$ is assigned separately for each input variable dimension and is chosen to approximately coincide with the input levels represented in the data. *Interaction BFs* are created by multiplying an existing BF with a truncated linear function involving a new variable. In this case, both the existing BF and the newly created interaction BF are used in the MARS approximation. Provided the observations represented by the data $(\bar{\boldsymbol{x}}_i, \bar{y}_i)$ $(i = 1, 2, ..., N)$ the form of the mth BF is as follows:

$$\boldsymbol{\psi}_m(\boldsymbol{x}) := \prod_{j=1}^{K_m} [s_{\kappa_j^m} \cdot (x_{\kappa_j^m} - \tau_{\kappa_j^m})]_+, \tag{3.2.5}$$

where K_m is the number of truncated linear functions multiplied in the mth BF,

$x_{\kappa_j^m}$ is the input variable corresponding to the jth truncated linear function in the mth BF, $\boldsymbol{\tau}_{\kappa_j^m}$ is the knot value corresponding to the variable $x_{\kappa_j^m}$ and $s_{\kappa_j^m}$ is the selected sign +1 or -1. A lack of fit criterion is used to compare the possible BFs. The search of new BFs can be restricted to interactions of a maximum order. For example, if only up to two-factor interactions are permitted, then $K_m \leq 2$ would be restricted in.

The first fundamental drawback of recursive partitioning strategies like CART [7] which uses indicator functions, is the lack of continuity, which affects the model accuracy. Secondly, 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. For this reason, the MARS algorithm is a modifed recursive partitioning algorithm which has important advantages compared to other recursive partitioning algorithms.

The MARS algorithm for estimating the model function $f(\boldsymbol{x})$ consists of two algorithms (Friedman 1991) [20]:

(i) The *forward stepwise algorithm:* Here, forward stepwise search for the BF starts with the constant BF, the only one presents initially. At each step, the split that minimizes some "lack of fit" from all the possible splits on each BF is chosen. The process stops when a user-specified value M_{max} is reached. At the end of this process, we have a large expression given in (3.2.4). This model typically overfits the data and so a *backward* deletion procedure is applied.

(ii) The *backward stepwise algorithm*: The purpose of this algorithm is to prevent from overfitting by decreasing the complexity of the model without degrading the fit to the data. Therefore, the backward stepwise algorithm involves removing from the model such BFs that contribute to the smallest increase in the RSS error at each stage, producing an optimally estimated model $\hat{f}_\alpha$ with respect to each number of terms, called α. Note here that α 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. For our explanations on C-MARS, we prefer to use the following notation for this criterion which is also mentioned in the previous section:

$$GCV := \frac{1}{N} \frac{\sum_{i=1}^{N} (y_i - \hat{f}_\alpha(x_i))^2}{(1 - \mathbf{M}(\alpha)/N)^2}, \tag{3.2.6}$$

where $\mathbf{M}(\alpha) := u + dK$, α depending on (u, d, K) [10]. Here, N is the number of sample observations, u is the number of linearly independent BFs, K is the number of knots selected in the forward process, and d is a cost for BF optimization as well as a smoothing parameter for the procedure. We do not employ the backward stepwise algorithm to estimate the function $f(\boldsymbol{x})$. At its place, as an alternative, we propose to use penalty terms in addition to the LSE to control the lack of fit from the viewpoint of the *complexity* of the estimation. We shall explain this below. Because of this new treatment offered, we do not need to run the backward stepwise algorithm.

3.2.2 The Penalized Residual Sum of Squares Problem

Let us use the penalized residual sum of squares (PRSS) with M_{max} BFs having been accumulated in the *forward stepwise algorithm.* For the MARS model, PRSS has the following form:

$$PRSS := \sum_{i=1}^{N} (y_i - f(\bar{\boldsymbol{x}}_i))^2 + \sum_{m=1}^{M_{max}} \lambda_m \sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s \in V_m}} \int \theta_m^2 \left[D_{r,s}^{\boldsymbol{\alpha}} \boldsymbol{\psi}_m(\boldsymbol{t}^m) \right]^2 d\boldsymbol{t}^m, \tag{3.2.7}$$

where $V_m := \left\{ \kappa_j^m | j = 1, 2, ..., K_m \right\}$ is the variable set associated with the mth basis function $\boldsymbol{\psi}_m$, $\boldsymbol{t}^m = \left(t_{m_1}, t_{m_2}, ..., t_{m_{K_m}} \right)^T$ represents the vector of variables which contribute to the mth basis function ψ_m. The parameter λ_m are nonnegative ($\lambda_m \geq 0$), and in the role of *penalty parameters* ($m = 1, 2, ..., M_{max}$). While the integrals of the second-order derivatives measure the *energy (unstability, complexity)* inscribed into the model (via the model functions) [25, 51], the integral of the first-order derivatives measure the *flatness* of the model functions. Furthermore, we refer to

$$D_{r,s}^{\boldsymbol{\alpha}} \boldsymbol{\psi}_m(\boldsymbol{t}^m) := \frac{\partial^{\boldsymbol{\alpha}} \boldsymbol{\psi}_m}{\partial^{\alpha_1} t_r^m \, \partial^{\alpha_2} t_s^m} (\boldsymbol{t}^m)$$

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 $D^{\boldsymbol{\alpha}}_{r,s}\boldsymbol{\psi}_m(\boldsymbol{t}^m)$ vanishes, and by addressing indices $r < s$, we have applied Schwarz's Theorem. In order not to overload the exposition, we still accept a slight flaw in the notation since in case of $|\boldsymbol{\alpha}| = 1$ and $K_m > 1$ the integral terms become mentioned K_m times by the pair $r < s$. By redefining λ_m by λ_m / K_m, this little deficiency could be easily corrected. The reader may choose a notation of his or her preference. Furthermore, for convenience, we use the integral symbol "$\int$" as a dummy in the sense of $\int\limits_{Q^m}$, where Q^m is some appropriately large K_m-dimensional parallelpipe where the integration takes place. We shall come back to this below. Finally, since all the regarded derivatives of any function $\boldsymbol{\psi}_m$ exist except on a set of measure zero, the integrals and entire optimization problems are well defined [53].

Our 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. The goal on a small complexity encompasses two parts.

Firstly, the areas where the base functions contribute to an explanation of the observations, should be large. In the case of classification, this means that the classes should be big rather than small. This aim is achieved by a *"flat"* model which is the linear combination of the BFs, together with our wish to have small residual errors; i.e., the model being "lifted" from the coordinate axes towards the data points $(\bar{\boldsymbol{x}}_i, \bar{y}_i)$ $(i = 1, 2, ..., N)$. Here, the basic idea is to dampen the slope of the linear parts of the BFs via the parameters θ_m, while still guaranteeing a quite satisfactory goodness of data fitting. Secondly, we aim at *stability* of the estimation, by taking care that the curvatures of the model function with its compartments according to (3.2.4)-(3.2.5), are not so high and, hence, their oscillations is not so frequent and intense. For closer information we refer to the paper of Taylan, Weber and Beck (2007) [52]. Motivated in this way, both first- and second-order partial derivatives of the model function f, better to say: of its additive components, enter our penalty terms in order to keep the complexity of the LS estimation appropriately low.

In this study, we tackle that tradeoff by means of penalty methods, such as regularization techniques [2], and by CQP [5, 16, 47].

If we take into account the representations (3.2.4) and (3.2.5) in (3.2.7), then the objectice function (3.2.7) will be of the following form [53]:

$$
\begin{aligned}
PRSS &= \sum_{i=1}^{N}\left(\bar{y}_i - \theta_0 - \sum_{m=1}^{M}\theta_m \boldsymbol{\psi}_m(\bar{\boldsymbol{x}}_i^m) - \sum_{m=M+1}^{M_{max}} \theta_m \boldsymbol{\psi}_m(\bar{\boldsymbol{x}}_i^m)\right)^2 \\
&+ \sum_{m=1}^{M_{max}} \lambda_m \sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_m}} \int \theta_m^2 \left[D_{r,s}^{\boldsymbol{\alpha}} \boldsymbol{\psi}_m(\boldsymbol{t}^m)\right]^2 d\boldsymbol{t}^m,
\end{aligned}
\qquad (3.2.8)
$$

where $\bar{\boldsymbol{x}}_i = (\bar{x}_{i,1}, \bar{x}_{i,2}, ..., \bar{x}_{i,p})^T$ denotes any of the input vectors and $\bar{\boldsymbol{x}}_i^m = \left(\bar{x}_{i,\kappa_1}, \bar{x}_{i,\kappa_2}, ..., \bar{x}_{i,\kappa_{K_m}}\right)^T$ stands for the corresponding projection vectors of $\bar{\boldsymbol{x}}_i$ onto those coordinates which contribute to the mth BF $\boldsymbol{\psi}_m$, they are related with the ith output $\bar{y}_i$. In matrix notation, the vectors $\bar{\boldsymbol{x}}_i^m$ $(i = 1, 2, ..., N)$ for the mth BF could also be compactly comprised as follows:

$$
\left(\bar{\boldsymbol{x}}_{i,\kappa_j^m}^m\right)_{\substack{i=1,2,...,N \\ j=1,2,...,K_m}} = \begin{bmatrix} \bar{x}_{1,\kappa_1^m}^m & \bar{x}_{1,\kappa_2^m}^m & \cdots & \bar{x}_{1,\kappa_{K_m}^m}^m \\ \bar{x}_{2,\kappa_1^m}^m & \bar{x}_{2,\kappa_2^m}^m & \cdots & \bar{x}_{2,\kappa_{K_m}^m}^m \\ \vdots & \vdots & \cdots & \vdots \\ \bar{x}_{N,\kappa_1^m}^m & \bar{x}_{N,\kappa_2^m}^m & \cdots & \bar{x}_{N,\kappa_{K_m}^m}^m \end{bmatrix}.
$$

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

$$
\begin{aligned}
PRSS &= \sum_{i=1}^{N} \big(y_i - \boldsymbol{\theta}^T \boldsymbol{\psi}(\bar{\boldsymbol{d}}_i)\big)^2 \\
&+ \sum_{m=1}^{M_{max}} \lambda_m \sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_m}} \int \theta_m^2 \left[D_{r,s}^{\boldsymbol{\alpha}} \boldsymbol{\psi}(\boldsymbol{t}^m)\right]^2 d\boldsymbol{t}^m, \qquad (3.2.9)
\end{aligned}
$$

where $\boldsymbol{\psi}(\bar{\boldsymbol{d}}_i) := \big(1, \psi_1(\bar{\boldsymbol{x}}_i^1), ..., \psi_M(\bar{\boldsymbol{x}}_i^M), \psi_{M+1}(\bar{\boldsymbol{x}}_i^{M+1}), ..., \psi_{M_{max}}(\bar{\boldsymbol{x}}_i^{M_{max}})\big)^T$, $\boldsymbol{\theta} := (\theta_0, \theta_1, ..., \theta_{M_{max}})^T$ with the point $\bar{\boldsymbol{d}}_i := \big(\bar{\boldsymbol{x}}_i^1, \bar{\boldsymbol{x}}_i^2, ..., \bar{\boldsymbol{x}}_i^M, \bar{\boldsymbol{x}}_i^{M+1}, ..., \bar{\boldsymbol{x}}_i^{M_{max}}\big)^T$ in the argument. In matrix notation, the vectors $\boldsymbol{\psi}(\bar{\boldsymbol{d}}_i)$ $(i = 1, 2, ..., N)$ can be compactly comprised as follows:

$$
\boldsymbol{\psi}(\bar{\boldsymbol{d}}_i) := \begin{bmatrix}
1 & \psi_1(\bar{\boldsymbol{x}}_1^1) & \cdots & \psi_M(\bar{\boldsymbol{x}}_1^M) & \cdots & \psi_{M_{max}}(\bar{\boldsymbol{x}}_1^{M_{max}}) \\
1 & \psi_1(\bar{\boldsymbol{x}}_2^1) & \cdots & \psi_M(\bar{\boldsymbol{x}}_2^M) & \cdots & \psi_{M_{max}}(\bar{\boldsymbol{x}}_2^{M_{max}}) \\
\vdots & \vdots & \cdots & & \cdots & \vdots \\
1 & \psi_1(\bar{\boldsymbol{x}}_N^1) & \cdots & \psi_M(\bar{\boldsymbol{x}}_2^M) & \cdots & \psi_{M_{max}}(\bar{\boldsymbol{x}}_N^{M_{max}})
\end{bmatrix}.
$$

On the other hand, in matrix notation, the vector $\bar{\boldsymbol{d}}_i$ $(i = 1, 2, ..., N)$ could be compactly comprised as a matrix in the following way,

$$
\bar{\boldsymbol{d}} := \begin{bmatrix}
\bar{\boldsymbol{x}}_1^1 & \bar{\boldsymbol{x}}_1^2 & \cdots & \bar{\boldsymbol{x}}_1^{M_{max}} \\
\bar{\boldsymbol{x}}_2^1 & \bar{\boldsymbol{x}}_2^2 & \cdots & \bar{\boldsymbol{x}}_2^{M_{max}} \\
\vdots & \vdots & \cdots & \vdots \\
\bar{\boldsymbol{x}}_N^1 & \bar{\boldsymbol{x}}_N^2 & \cdots & \bar{\boldsymbol{x}}_N^{M_{max}}
\end{bmatrix}.
$$

To approximate the multi-dimensional integrals

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

we use discretized forms of them instead [53]. For this purpose, our data point $(\bar{\boldsymbol{x}}_l, \bar{y}_l)$ $(l = 1, 2, ..., N)$ with $\bar{\boldsymbol{x}}_l \in \mathbb{R}^n$ are given. In a natural way, these input data $\bar{\boldsymbol{x}}_l = (\bar{x}_{l,1}, \bar{x}_{l,2}, ..., \bar{x}_{l,p})^T$ $(l = 1, 2, ..., N)$ generate a subdivision of any sufficiently large parallelpipe Q of $\mathbb{R}^n$ which contains each of them as elements. Let Q be a parallelpipe which encompasses all our input data; we represent it by

$$Q = [a_1, b_1] \times [a_2, b_2] \times ... \times [a_p, b_p] = \prod_{j=1}^{p} Q_j,$$

where $Q_j =: [a_j, b_j]$, $a_j < \bar{x}_{l,j} < b_j$ $(j = 1, 2, ..., p)$ $(l = 1, 2, ..., N)$. Without loss of generality, we may assume $a_j < \bar{x}_{l,j} < b_j$. For all j we reorder the coordinates of the input data points: $\bar{x}_{l_1^j,j} \leq \bar{x}_{l_2^j,j} \leq ... \leq \bar{x}_{l_N^j,j}$, where $l_\sigma^j = 1, 2, ..., N$ $(\sigma = 1, 2, ..., N; j = 1, 2, ..., p)$, and $\bar{x}_{l_\sigma^j,j}$ is the jth component of $\bar{x}_{l_\sigma^j}$, the l_σ^j input vector after reordering. Without loss of generality we may assume $\bar{x}_{l_\sigma^j,j} \neq \bar{x}_{l_\varphi^j,j}$ for all $\sigma, \varphi = 1, 2, ..., N$ with $\sigma \neq \varphi$; i.e., $\bar{x}_{l_1^j,j} < \bar{x}_{l_2^j,j} < ... < \bar{x}_{l_N^j,j}$ $(j = 1, 2, ..., p)$. The symbol "×" and "$\prod$" are used for Cartesian product, and and "$\prod$"is also used for the multiplication of numbers [53].

Indeed, whenever "=" is attained for some coordinate, we would obtain subparallelpipes of a lower dimension in the following integration process and its approximation, i.e., zero sets [53]. Let us denote

$$\bar{x}_{l_0^j,j} := a_j,\ l_0^j := 0;\ \ \bar{x}_{l_{N+1}^j,j} := b_j,\ l_{N+1}^j := N + 1.$$

Then,

$$Q = \bigcup_{\sigma^j=0}^{N} \prod_{j=1}^{p} \left[\bar{x}_{l_{\sigma^j}^j,j}, \bar{x}_{l_{\sigma^j+1}^j,j} \right].$$

Based on the aforementioned notation, we discretize our integrals according to the following approximate relations:

$$\int_Q f(\boldsymbol{t}) d\boldsymbol{t} \approx \sum_{(\sigma^j)_{j\in\{1,2,...,p\}} \in \{0,1,2,...,N+1\}^p} f\left(\bar{x}_{l_{\sigma^1}^1,1}, \bar{x}_{l_{\sigma^2}^2,2}, ..., \bar{x}_{l_{\sigma^p}^p,p} \right) \prod_{j=1}^{p} \left(\bar{x}_{l_{\sigma^j+1}^j,j} - \bar{x}_{l_{\sigma^j}^j,j} \right).$$

In our study, that notation, subdivision and approximation needs to be done for all

$$\left[D_{r,s}^{\boldsymbol{\alpha}} \boldsymbol{\psi}_m(\boldsymbol{t}^m) \right]^2,$$

with the corresponding variables and lower dimensions of $\boldsymbol{t}^m$ also. For this purpose, we look at the projection of Q into $\mathbb{R}^{K_m}$ related with the special coordinates of $\boldsymbol{t}^m$ and we can take the subdivision of the corresponding Q^m according to the subdivision obtained for Q.

Then, if we apply this idea to our case, we write discretization form as

$$\int\limits_{Q^m} \theta_m^2 \left[D^{\boldsymbol{\alpha}}_{r,s}\boldsymbol{\psi}_m(\boldsymbol{t}^m)\right]^2 d\boldsymbol{t}^m \approx \sum_{(\sigma^j)_{j\in\{1,2,\ldots,p\}}\in\{0,1,2,\ldots,N+1\}^{K_m}} \theta_m^2 \cdot$$

$$\left[D^{\boldsymbol{\alpha}}_{r,s}\boldsymbol{\psi}_m(\bar{x}_{l^{\kappa_1^m}_{\sigma^{\kappa_1^m}},\kappa_1^m}, \ldots, \bar{x}_{l^{\kappa^m_{K_m}}_{\sigma^{\kappa^m_{K_m}}},\kappa^m_{K_m}})\right]^2 \cdot \prod_{j=1}^{K_m}\left(\bar{x}_{l^{\kappa_j^m}_{\sigma^{\kappa_j^m}+1},\kappa_j^m} - \bar{x}_{l^{\kappa_j^m}_{\sigma^{\kappa_j^m}},\kappa_j^m}\right).$$

Then, we can rearrange PRSS in the following form [53]:

$$PRSS \approx \sum_{i=1}^{N}\left(y_i - \boldsymbol{\theta}^T\boldsymbol{\psi}(\bar{\boldsymbol{d}}_i)\right)^2 + \sum_{m=1}^{M_{max}}\lambda_m \sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_m}} \sum_{(\sigma^{\kappa_j})} \theta_m^2 \cdot$$

$$\left[D^{\boldsymbol{\alpha}}_{r,s}\boldsymbol{\psi}_m(\bar{x}_{l^{\kappa_1^m}_{\sigma^{\kappa_1^m}},\kappa_1^m}, \ldots, \bar{x}_{l^{\kappa^m_{K_m}}_{\sigma^{\kappa^m_{K_m}}},\kappa^m_{K_m}})\right]^2 \cdot \prod_{j=1}^{K_m}\left(\bar{x}_{l^{\kappa_j^m}_{\sigma^{\kappa_j}+1},\kappa_j^m} - \bar{x}_{l^{\kappa_j^m}_{\sigma^{\kappa_j}},\kappa_j^m}\right), \qquad (3.2.10)$$

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}) [53]:

$$\hat{\boldsymbol{x}}_i^m = \left(\bar{x}_{l^{\kappa_1^m}_{\sigma^{\kappa_1^m}},\kappa_1^m}, \ldots, \bar{x}_{l^{\kappa^m_{K_m}}_{\sigma^{\kappa^m_{K_m}}},\kappa^m_{K_m}}\right), \quad \Delta\hat{\boldsymbol{x}}_i^m := \prod_{j=1}^{K_m}\left(\bar{x}_{l^{\kappa_j^m}_{\sigma^{\kappa_j}+1},\kappa_j^m} - \bar{x}_{l^{\kappa_j^m}_{\sigma^{\kappa_j}},\kappa_j^m}\right). \qquad (3.2.11)$$

By (3.2.11), we can approximate $PRSS$ as follows:

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

(3.2.12)

For a short representation, we can rewrite the approximate relation (3.2.10) as

$$PRSS \approx \left\| \boldsymbol{y} - \psi(\bar{\boldsymbol{d}})\boldsymbol{\theta} \right\|_2^2 + \sum_{m=1}^{M_{max}} \lambda_m \sum_{i=1}^{(N+1)^{K_m}} L_{im}^2 \theta_m^2, \tag{3.2.13}$$

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

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

The first parts of PRSS equations in (3.2.12) and (3.2.13) are equal. We can show as follows how the first part of the equation in (3.2.12) turns into the first part of the PRSS equation in (3.2.13):

$$\begin{aligned} \sum_{i=1}^{N} \big(y_i - \boldsymbol{\theta}^T \boldsymbol{\psi}(\bar{\boldsymbol{d}}_i)\big)^2 \quad = \quad & \big(y_1 - \boldsymbol{\theta}^T \boldsymbol{\psi}(\bar{\boldsymbol{d}}_1)\big)^2 + \\ & \big(y_2 - \boldsymbol{\theta}^T \boldsymbol{\psi}(\bar{\boldsymbol{d}}_2)\big)^2 + \\ & \qquad \vdots \\ & \big(y_N - \boldsymbol{\theta}^T \boldsymbol{\psi}(\bar{\boldsymbol{d}}_N)\big)^2 \end{aligned}$$

$$= \quad \Big[y_1 - \boldsymbol{\theta}^T \boldsymbol{\psi}(\bar{\mathbf{d}}_1), \quad y_2 - \boldsymbol{\theta}^T \boldsymbol{\psi}(\bar{\boldsymbol{d}}_2), \quad \cdots \quad , y_N - \boldsymbol{\theta}^T \boldsymbol{\psi}(\bar{\boldsymbol{d}}_N) \Big] \begin{bmatrix} y_1 - \boldsymbol{\theta}^T \boldsymbol{\psi}(\bar{\boldsymbol{d}}_1) \\ y_2 - \boldsymbol{\theta}^T \boldsymbol{\psi}(\bar{\boldsymbol{d}}_2) \\ \vdots \\ y_N - \boldsymbol{\theta}^T \boldsymbol{\psi}(\bar{\boldsymbol{d}}_N) \end{bmatrix}.$$

If we write the above equation in vector notation, we can get the following equation:

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

3.2.3 Tikhonov Regularization Applied

Now, we approach our problem *PRSS* as a *Tikhonov regularization problem* [2]. For this purpose we consider formula (3.2.13) again, arranging it as follows [53]:

$$
\begin{aligned}
PRSS &\approx \left\|\boldsymbol{y}-\psi(\bar{\boldsymbol{d}})\boldsymbol{\theta}\right\|_2^2+\sum_{m=1}^{M_{max}}\lambda_m\sum_{i=1}^{(N+1)^{K_m}}L_{im}^2\theta_m^2\\
&= \left\|\boldsymbol{y}-\psi(\bar{\boldsymbol{d}})\boldsymbol{\theta}\right\|_2^2+\sum_{m=1}^{M_{max}}\lambda_m\left[(L_{1m}\theta_m)^2+(L_{2m}\theta_m)^2+...+(L_{(N+1)^{K_m}m}\theta_m)^2\right]\\
&= \left\|\boldsymbol{y}-\psi(\bar{\boldsymbol{d}})\boldsymbol{\theta}\right\|_2^2+\\
&\quad \sum_{m=1}^{M_{max}}\lambda_m\left(\left[L_{1m}\theta_m,\quad L_{2m}\theta_m,\cdots,L_{(N+1)^{K_m}m}\theta_m\right]\begin{bmatrix}L_{1m}\theta_m\\ L_{2m}\theta_m\\ \vdots\\ L_{(N+1)^{K_m}m}\theta_m\end{bmatrix}\right)\\
&= \left\|\boldsymbol{y}-\boldsymbol{\psi}(\bar{\boldsymbol{d}})\boldsymbol{\theta}\right\|_2^2+\sum_{m=1}^{M_{max}}\lambda_m\left\|\boldsymbol{L}_m\theta_m\right\|_2^2\\
&= \left\|\boldsymbol{y}-\boldsymbol{\psi}(\bar{\boldsymbol{d}})\boldsymbol{\theta}\right\|_2^2+\lambda_1\left\|\boldsymbol{L}_1\theta_1\right\|_2^2+\lambda_2\left\|\boldsymbol{L}_2\theta_2\right\|_2^2+...+\\
&\quad \lambda_{M_{max}}\left\|\boldsymbol{L}_{M_{max}}\theta_{M_{max}}\right\|_2^2, \qquad (3.2.14)
\end{aligned}
$$

where $\boldsymbol{L}_m := (L_{1m}, L_{2m}, ..., L_{(N+1)^{K_m},m})^T$ $(m = 1, 2, ..., M_{max})$. But, rather than a singleton, there is a finite sequence of the *tradeoff* or *penalty* parameters $\lambda_1, \lambda_2, ..., \lambda_{M_{max}}$ such that this equation is *not* yet a *Tikhonov regularization problem* with a single such parameter. For this reason, let us make a uniform penalization by taking the same λ for each derivative term, i.e., $\lambda_1 = \lambda_2 = ... = \lambda_{M_{max}} =: \lambda$, where $\lambda_m \geq 0$ $(m = 1, 2, ..., M_{max})$. Then, our approximation of *PRSS* can be rearranged as

$$PRSS \approx \left\|\boldsymbol{y}-\boldsymbol{\psi}(\bar{\boldsymbol{d}})\boldsymbol{\theta}\right\|_2^2+\lambda\left\|\boldsymbol{L}\theta\right\|_2^2, \qquad (3.2.15)$$

where $\boldsymbol{L}$ 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. Furthermore, $\boldsymbol{\theta}$ is an $((M_{max} + 1) \times 1)$-parameter vector to be estimated through the data points. Let us state explicitly [53]:

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

Then, our *PRSS* problem looks as a *Tikhonov regularization problem* (2.2.29) with $\varphi > 0$, i.e., $\lambda = \varphi^2$ for some $\varphi \in \mathbb{R}$ [2].

Tikhonov regularization problem has multiple objective functions through a linear combination of $\left\|\boldsymbol{y} - \boldsymbol{\psi}(\bar{\boldsymbol{d}})\boldsymbol{\theta}\right\|_2^2$ and $\|\boldsymbol{L\theta}\|_2^2$. We select the solutions such that it minimizes both first objective function ($\left\|\boldsymbol{y} - \boldsymbol{\psi}(\bar{\boldsymbol{d}})\boldsymbol{\theta}\right\|_2^2$) and second objective function ($\|\boldsymbol{L\theta}\|_2^2$). Therefore, we can consider Tikhonov regularization problem as a multiobjective problem. Indeed, our Tikhonov regularization problem combines these two objective functions into a single functional form. This combination is a weighted linear sum of the objectives. We set a weight by a penalty parameter λ. The solutions are obtained by referring to such a given weighted sum. However, coming up with meaningful combinations of weights can be challenging.

3.2.4 An Alternative for Tikhonov Regularization Problem with Conic Quadratic Programming

Construction of the Conic Quadratic Programming Problem

Let us tackle the Tikhonov regularization problem (3.2.15) with *CQP* which is a continuous optimization program with its corresponding technique. We can easily formulate *PRSS* as a *CQP* problem (please revisit Section 2). Indeed, based on an appropriate choice of a bound $\bar{M}$ we state the following optimization problem [53]:

$$\min_{\boldsymbol{\theta}} \quad \left\| \boldsymbol{\psi}(\bar{\boldsymbol{d}})\boldsymbol{\theta} - \boldsymbol{y} \right\|_2^2 \tag{3.2.16}$$

$$\text{subject to} \quad \| \boldsymbol{L\theta} \|_2^2 \leq \bar{M}.$$

Let us underline that this choice of $\bar{M}$ should be the outcome of a careful learning process, with the help of model-free or model-based methods [25]. In (3.2.16), we have the LS objective function $\left\| \boldsymbol{\psi}(\bar{\boldsymbol{d}})\boldsymbol{\theta} - \boldsymbol{y} \right\|_2^2$ and the inequality constraint function $- \| \boldsymbol{L\theta} \|_2^2 + \bar{M}$ which is requested to be nonnegative for feasibility. Now, by a classical epigraph argument, we equivalently write our optimization problems as follows [53]:

$$\min_{t,\boldsymbol{\theta}} \quad t, \tag{3.2.17}$$

$$\text{subject to} \quad \left\| \boldsymbol{\psi}(\bar{\boldsymbol{d}})\boldsymbol{\theta} - \boldsymbol{y} \right\|_2^2 \leq t^2, \quad t \geq 0,$$

$$\| \boldsymbol{L\theta} \|_2^2 \leq \bar{M}.$$

Please note that we have introduced a new variable, the hight variable [53]. Now, equivalently again, our problem looks so:

$$\min_{t,\boldsymbol{\theta}} \quad t, \tag{3.2.18}$$

$$\text{subject to} \quad \left\| \boldsymbol{\psi}(\bar{\boldsymbol{d}})\boldsymbol{\theta} - \boldsymbol{y} \right\|_2 \leq t,$$

$$\| \boldsymbol{L\theta} \|_2 \leq \sqrt{\bar{M}}.$$

Let us use modern methods of *continuous optimization techniques*, especially, from CQP where we use the basic notation [52]:

$$\min_{\boldsymbol{x}} \ \boldsymbol{c}^T\boldsymbol{x}, \quad \text{subject to} \ \ \|\boldsymbol{D}_i\boldsymbol{x} - \boldsymbol{d}_i\| \leq \boldsymbol{p}_i^T\boldsymbol{x} - \boldsymbol{q}_i \ \ (i = 1, 2, ..., k). \tag{3.2.19}$$

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

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

$$\boldsymbol{q}_1 = 0, \ \boldsymbol{D} = (\boldsymbol{0}_{M_{max}+1}, \boldsymbol{L}), \ \boldsymbol{d}_2 = \boldsymbol{0}_{M_{max}+1}, \ \boldsymbol{p}_1 = \boldsymbol{0}_{M_{max}+2} \text{ and } \ \boldsymbol{q}_2 = -\sqrt{\bar{M}}.$$

In order to write the optimality condition for this problem, we firstly reformulate the problem (3.2.18) as follows [53]:

$$\min_{t,\boldsymbol{\theta}} \quad t, \tag{3.2.20}$$

$$\begin{aligned} \text{such that} \quad \boldsymbol{\chi} \ &:= \begin{bmatrix} \boldsymbol{0}_N & \boldsymbol{\psi}(\bar{\boldsymbol{d}}) \\ 1 & \boldsymbol{0}^T_{M_{max}+1} \end{bmatrix} \begin{bmatrix} t \\ \boldsymbol{\theta} \end{bmatrix} + \begin{bmatrix} -\boldsymbol{y} \\ 0 \end{bmatrix} \\ \boldsymbol{\eta} \ &:= \begin{bmatrix} \boldsymbol{0}_{M_{max}+1} & \boldsymbol{L} \\ 0 & \boldsymbol{0}^T_{M_{max}+1} \end{bmatrix} \begin{bmatrix} t \\ \boldsymbol{\theta} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0}_{M_{max}+1} \\ \sqrt{\bar{M}} \end{bmatrix}, \end{aligned}$$

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

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

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

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

$$\max \quad (\boldsymbol{y}^T, 0)\omega_1 + \left(\boldsymbol{0}^T_{M_{max}+1}, -\sqrt{M}\right)\omega_2 \tag{3.2.21}$$

$$\text{such that} \quad \boldsymbol{\chi} \;:=\; \begin{bmatrix} \boldsymbol{0}_N^T & 1 \\ \boldsymbol{\psi}(\bar{\boldsymbol{d}}) & \boldsymbol{0}^T_{M_{max}+1} \end{bmatrix} \boldsymbol{\omega}_1 + \begin{bmatrix} \boldsymbol{0}^T_{M_{max}+1} & 0 \\ \boldsymbol{L}^T & \boldsymbol{0}_{M_{max}+1} \end{bmatrix} \boldsymbol{\omega}_2 = \begin{bmatrix} 1 \\ 0_{M_{max}+1} \end{bmatrix},$$

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

Moreover, $(t, \boldsymbol{\theta}, \boldsymbol{\chi}, \boldsymbol{\eta}, \boldsymbol{\omega}_1, \boldsymbol{\omega}_2)$ is a *primal dual optimal solution* if and only if [53]

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

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

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

$$\begin{aligned} \boldsymbol{\omega}_1^T \boldsymbol{\chi} = 0, \quad \boldsymbol{\omega}_2^T \boldsymbol{\eta} = 0, \\ \boldsymbol{\omega}_1 \in L^{N+1}, \quad \boldsymbol{\omega}_2 \in L^{M_{max}+2}, \\ \boldsymbol{\chi} \in L^{N+1}, \quad \boldsymbol{\eta} \in L^{M_{max}+2}. \end{aligned}$$

In order to provide with some fundamental facts on the solution methods for *CQP* and convex problem classes beyond [53], we state the Subsection 2.2.2 of this thesis.

3.3 Numerical Example for C-MARS

The data set that we used for our numerical example of C-MARS has five predictor variables $(x_1, x_2, x_3, x_4, x_5)$ and contains 32 observations (taken from Myers and Montgomery (2002) [44] p. 71). Here, we write $\boldsymbol{x}$ as a *generic* variable in the corresponding space $\mathbb{R}^l$ $(l \in \{1, 2, ..., 5\})$. Later, we will write $\boldsymbol{x}$ $\boldsymbol{t}^1, \boldsymbol{t}^2, ...$ or $\boldsymbol{t}^5$. In order to build the MARS model by trial and error we set the maximum number of BFs allowed to five, i.e., $M_{max} = 5$ and set the highest degree of interaction allowed to be two. Then the number of maximum basis functions and interactions which are constructed by using MARS version 2 developed by Salford Systems are as follows:

$$\begin{aligned}
\psi_1(\boldsymbol{x}) &= \max\{0, x_2 + 0.159\}, \\
\psi_2(\boldsymbol{x}) &= \max\{0, -0.159 - x_2\}, \\
\psi_3(\boldsymbol{x}) &= \max\{0, x_4 + 1.517\}, \\
\psi_4(\boldsymbol{x}) &= \max\{0, x_1 + 2.576\} * \max\{0, x_4 + 1.517\}, \\
\psi_5(\boldsymbol{x}) &= \max\{0, x_5 + 1.562\} * \max\{0, x_4 + 1.517\}.
\end{aligned}$$

The BFs ψ_1 and ψ_2 are the standard BFs and mirror image (reflected) BFs for the predictor x_2, respectively. The graphical representation of ψ_1 and ψ_2 is given in Figure 3.4.

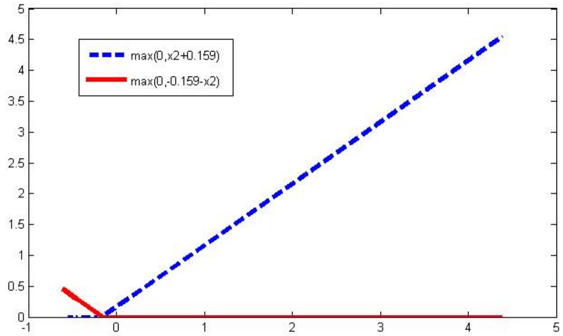

Figure 3.4: The graphical representation of BFs 1 and 2.

The x_2 value of -0.159 is found to be the knot point for the predictor x_2. This knot point is a value both for BFs ψ_1 and ψ_2.

For x_4, there is only one standard BF ψ_3, where the knot location has the value -1.517. Figure 3.5 shows the BF ψ_3:

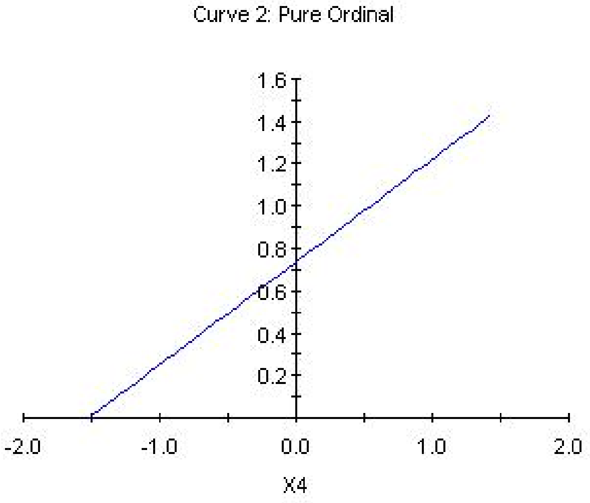

Figure 3.5: The graphical representation of basis function 3 [39].

While the BF ψ_4 uses the BF ψ_3 to express the interaction between the variables x_1 and x_4, the BF ψ_5 uses the BF ψ_3 to express the interaction between the input variables x_5 and x_4. The interactions between the predictor variables are presented in Figure 3.6 and Figure 3.7.

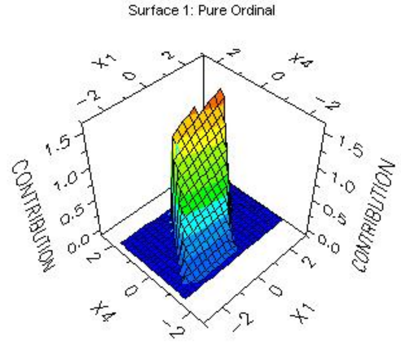

Figure 3.6: The graphical representation of interactions between the predictor variables x_1 and x_4 [39].

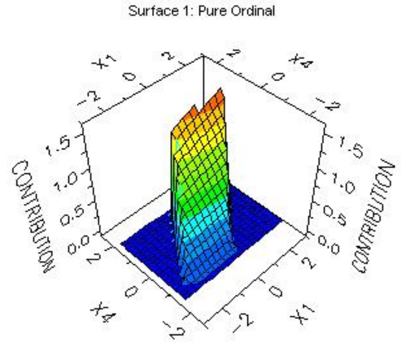

Figure 3.7: The graphical representation of interactions between the predictor variables x_4 and x_5 [39].

In order to prevent our optimization problem from nondifferentiability, we choose the knot values very near to the input values of the data point. Below we select knot values for corresponding BFs:
For ψ_1:

$$\tau_{16,2} = -0.159 \ , \ \bar{x}_{16,2} = -0.1589 \implies \tau_{16,2} \neq \bar{x}_{16,2}.$$

For ψ_2:

$$\tau_{16,2} = -0.159 \ , \ \bar{x}_{16,2} = -0.1589 \implies \tau_{16,2} \neq \bar{x}_{16,2}.$$

For ψ_3:

$$\tau_{1,4} = -1.517 \ , \ \bar{x}_{1,4} = -1.5172 \implies \tau_{1,4} \neq \bar{x}_{1,4}.$$

For ψ_4:

$$\tau_{5,1} = -2.576 \ , \ \bar{x}_{5,1} = -2.5759 \implies \tau_{5,1} \neq \bar{x}_{5,1},$$
$$\tau_{1,4} = -1.517 \ , \ \bar{x}_{1,4} = -1.5172 \implies \tau_{1,4} \neq \bar{x}_{1,4}.$$

For ψ_5:

$$\tau_{28,5} = -1.5624 \ , \ \bar{x}_{28,5} = -1.562 \implies \tau_{28,5} \neq \bar{x}_{28,5},$$
$$\tau_{1,4} = -1.517 \ , \ \bar{x}_{1,4} = -1.5172 \implies \tau_{1,4} \neq \bar{x}_{1,4}.$$

The BFs given in (3.2.5), which are constructed for the numerical example, are looking as follows:

$$\begin{aligned}
\psi_1 : K_1 &= 1, \\
x_{\kappa_1^1} &= x_2, \\
\tau_{\kappa_1^1} &= -0.159, \\
s_{\kappa_1^1} &= +1, \\
\psi_1(\boldsymbol{t}^1) &= \prod_{j=1}^{K_1} \left[s_{\kappa_1^1} \cdot (x_{\kappa_1^1} - \tau_{\kappa_1^1}) \right]_+ \\
&= \left[s_{\kappa_1^1} \cdot (x_{\kappa_1^1} - \tau_{\kappa_1^1}) \right]_+ ,
\end{aligned}$$

$$\begin{aligned}
\psi_2 : K_2 &= 1, \\
x_{\kappa_1^2} &= x_2, \\
\tau_{\kappa_1^2} &= -0.159, \\
s_{\kappa_1^2} &= -1, \\
\psi_2(\boldsymbol{t}^2) &= \prod_{j=1}^{K_2} \left[s_{\kappa_1^2} \cdot (x_{\kappa_1^2} - \tau_{\kappa_1^2}) \right]_+ \\
&= \left[s_{\kappa_1^2} \cdot (x_{\kappa_1^2} - \tau_{\kappa_1^2}) \right]_+ ,
\end{aligned}$$

$$\begin{aligned}
\psi_3 : K_3 &= 1, \\
x_{\kappa_1^3} &= x_4, \\
\tau_{\kappa_1^3} &= -1.517, \\
s_{\kappa_1^3} &= +1, \\
\psi_2(\boldsymbol{t}^3) &= \prod_{j=1}^{K_3} \left[s_{\kappa_1^3} \cdot (x_{\kappa_1^3} - \tau_{\kappa_1^3}) \right]_+ \\
&= \left[s_{\kappa_1^3} \cdot (x_{\kappa_1^3} - \tau_{\kappa_1^3}) \right]_+ ,
\end{aligned}$$

$$\begin{aligned}
\psi_4 : K_4 &= 2, \\
x_{\kappa_1^4} &= x_1, \quad x_{\kappa_2^4} = x_4, \\
\tau_{\kappa_1^4} &= -2.576, \quad \tau_{\kappa_2^4} = -1.517, \\
s_{\kappa_1^4} &= +1, \quad s_{\kappa_2^4} = +1, \\
\psi_4(\boldsymbol{t}^4) &= \prod_{j=1}^{K_4} \left[s_{\kappa_j^4} \cdot (x_{\kappa_j^4} - \tau_{\kappa_j^4}) \right]_+ \\
&= \left[s_{\kappa_1^4} \cdot (x_{\kappa_1^4} - \tau_{\kappa_1^4}) \right]_+ \cdot \left[s_{\kappa_2^4} . (x_{\kappa_2^4} - \tau_{\kappa_2^4}) \right]_+ ,
\end{aligned}$$

$$\begin{aligned}
\psi_5 : K_5 &= 2, \\
x_{\kappa_1^5} &= x_5, \quad x_{\kappa_2^5} = x_4, \\
\tau_{\kappa_1^5} &= -1.562, \quad \tau_{\kappa_2^5} = -1.517, \\
s_{\kappa_1^5} &= +1, \quad s_{\kappa_2^5} = +1, \\
\psi_5(\boldsymbol{t}^5) &= \prod_{j=1}^{K_5} \left[s_{\kappa_j^5} \cdot (x_{\kappa_j^5} - \tau_{\kappa_j^5}) \right]_+ \\
&= \left[s_{\kappa_1^5} \cdot (x_{\kappa_1^5} - \tau_{\kappa_1^5}) \right]_+ \cdot \left[s_{\kappa_2^5} \cdot (x_{\kappa_2^5} - \tau_{\kappa_2^5}) \right]_+ .
\end{aligned}$$

The large model (3.2.4) for this numerical example is then obtained as follows:

$$\begin{aligned}
Y &= \theta_0 + \sum_{m=1}^{M} \theta_m \boldsymbol{\psi}_m(\boldsymbol{x}) + \epsilon \\
&= \theta_0 + \theta_1 \psi_1(\boldsymbol{x}) + \theta_2 \psi_2(\boldsymbol{x}) + \theta_3 \psi_3(\boldsymbol{x}) + \theta_4 \psi_4(\boldsymbol{x}) + \theta_5 \psi_5(\boldsymbol{x}) + \epsilon \\
&= \theta_0 + \theta_1 \max\{0, x_2 + 0.159\} + \theta_2 \max\{0, -0.159 - x_2\} + \theta_3 \max\{0, x_4 + 1.517\} \\
&\quad + \theta_4 \max\{0, x_1 + 2.576\} * \max\{0, x_4 + 1.517\} \\
&\quad + \theta_5 \max\{0, x_5 + 1.562\} * \max\{0, x_4 + 1.517\} + \epsilon.
\end{aligned}$$

For this numeric example, we can write the PRSS objective function in (3.2.7) as follows:

$$\begin{aligned}
PRSS &= \sum_{i=1}^{32}(y_i - f(\bar{\boldsymbol{x}}_i))^2 + \sum_{m=1}^{5}\lambda_m \sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_m}} \int \theta_m^2 \left[D_{r,s}^{\boldsymbol{\alpha}} \boldsymbol{\psi}_m(\boldsymbol{t}^m)\right]^2 d\boldsymbol{t}^m \\
&= \sum_{i=1}^{32}(y_i - f(\bar{\boldsymbol{x}}_i))^2 + \lambda_1 \left(\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_1}} \int \theta_1^2 \left[D_{r,s}^{\boldsymbol{\alpha}} \boldsymbol{\psi}_1(\boldsymbol{t}^1)\right]^2 d\boldsymbol{t}^1 \right) \\
&\quad + \lambda_2 \left(\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_2}} \int \theta_2^2 \left[D_{r,s}^{\boldsymbol{\alpha}} \boldsymbol{\psi}_2(\boldsymbol{t}^2)\right]^2 d\boldsymbol{t}^2 \right) \\
&\quad + \lambda_3 \left(\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_3}} \int \theta_3^2 \left[D_{r,s}^{\boldsymbol{\alpha}} \boldsymbol{\psi}_3(\boldsymbol{t}^3)\right]^2 d\boldsymbol{t}^3 \right) \\
&\quad + \lambda_4 \left(\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_4}} \int \theta_4^2 \left[D_{r,s}^{\boldsymbol{\alpha}} \boldsymbol{\psi}_4(\boldsymbol{t}^4)\right]^2 d\boldsymbol{t}^4 \right) \\
&\quad + \lambda_5 \left(\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_5}} \int \theta_5^2 \left[D_{r,s}^{\boldsymbol{\alpha}} \boldsymbol{\psi}_5(\boldsymbol{t}^5)\right]^2 d\boldsymbol{t}^5 \right).
\end{aligned}$$

For the above numerical example, all evaluations of the notations V_m and $\boldsymbol{t}^m$ $(m = 1, ..., 5)$ in the above equation are given below:

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

The corresponding derivatives for the BFs $D_{r,s}^{\alpha}\boldsymbol{\psi}_m(\boldsymbol{t}^m)$ $(m = 1, 2, ..., 5)$ are stated below. For the BF $\psi_1(\boldsymbol{t}^1) = \max\{0, x_2 + 0.159\}$, there is no interaction; so $r = s = 2$. The sum of selected first- and second-order derivatives of ψ_1 is

$$
\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_1}} \left[D_{r,s}^{\boldsymbol{\alpha}}\boldsymbol{\psi}_1(\boldsymbol{t}^1)\right]^2 d\boldsymbol{t}^1,
$$

where

$$
\begin{aligned}
|\boldsymbol{\alpha}| = 1: \qquad & D_2^1\psi_1(\boldsymbol{t}^1) := \frac{\partial\psi_1}{\partial t_1^1}(\boldsymbol{t}^1) = \frac{\partial\psi_1}{\partial x_2}(x_2) = 1 \ \text{ if } \ x_2 > -0.159 \ , \\
& D_2^1\psi_1(\boldsymbol{t}^1) := \frac{\partial\psi_1}{\partial t_1^1}(\boldsymbol{t}^1) = \frac{\partial\psi_1}{\partial x_2}(x_2) = 0 \ \text{ if } \ x_2 \leq -0.159 \ , \\
|\boldsymbol{\alpha}| = 2: \qquad & D_2^2\psi_1(\boldsymbol{t}^1) := \frac{\partial^2\psi_1}{\partial t_1^1 \partial t_1^1}(\boldsymbol{t}^1) = \frac{\partial^2\psi_1}{\partial x_2 \partial x_2}(x_2) = 0 \ \text{ for all } \ x_2.
\end{aligned}
$$

For the BF $\psi_2(\boldsymbol{t}^2) = \max\{0, -0.159 - x_2\}$, there is no interaction; so $r = s = 2$. The sum of selected first- and second-order derivatives of ψ_2 is

$$
\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_2}} \left[D_{r,s}^{\boldsymbol{\alpha}}\boldsymbol{\psi}_2(\boldsymbol{t}^2)\right]^2 d\boldsymbol{t}^2,
$$

where

$$
\begin{aligned}
|\boldsymbol{\alpha}| = 1: \qquad & D_2^1\psi_2(\boldsymbol{t}^2) := \frac{\partial \psi_2}{\partial t_1^2}(\boldsymbol{t}^2) = \frac{\partial \psi_2}{\partial x_2}(x_2) = -1 \ \text{ if } \ x_2 < -0.159 \ , \\
& D_2^1\psi_1(\boldsymbol{t}^1) := \frac{\partial \psi_1}{\partial t_1^2}(\boldsymbol{t}^2) = \frac{\partial \psi_2}{\partial x_2}(x_2) = 0 \ \text{ if } \ x_2 \geq -0.159 \ , \\
|\boldsymbol{\alpha}| = 2: \qquad & D_2^2\psi_2(\boldsymbol{t}^2) := \frac{\partial^2 \psi_2}{\partial t_1^2 \partial t_1^2}(\boldsymbol{t}^2) = \frac{\partial^2 \psi_2}{\partial x_2 \partial x_2}(x_2) = 0 \ \text{ for all } \ x_2.
\end{aligned}
$$

For the BF $\psi_3(\boldsymbol{t}^3) = \max\{0, x_4 + 1.517\}$, there is no interaction; so $r = s = 4$. The sum of selected first- and second-order derivatives of ψ_3 is

$$
\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \ \sum_{\substack{r<s \\ r,s \in V_3}} \left[D_{r,s}^{\boldsymbol{\alpha}} \boldsymbol{\psi}_3(\boldsymbol{t}^3) \right]^2 d\boldsymbol{t}^3,
$$

where

$$
\begin{aligned}
|\boldsymbol{\alpha}| = 1: \qquad & D_4^1\psi_3(\boldsymbol{t}^3) := \frac{\partial \psi_3}{\partial t_1^3}(\boldsymbol{t}^3) = \frac{\partial \psi_3}{\partial x_4}(x_4) = 1 \ \text{ if } \ x_4 > -1.517 \ , \\
& D_4^1\psi_3(\boldsymbol{t}^3) := \frac{\partial \psi_3}{\partial t_1^3}(\boldsymbol{t}^3) = \frac{\partial \psi_3}{\partial x_4}(x_4) = 0 \ \text{ if } \ x_4 \leq -1.517 \ , \\
|\boldsymbol{\alpha}| = 2: \qquad & D_4^2\psi_3(\boldsymbol{t}^3) := \frac{\partial^2 \psi_3}{\partial t_1^3 \partial t_1^3}(\boldsymbol{t}^3) = \frac{\partial^2 \psi_3}{\partial x_4 \partial x_4}(x_4) = 0 \ \text{ for all } \ x_4.
\end{aligned}
$$

For the BF $\psi_4(\boldsymbol{t}^4) = \max\{0, x_1 + 2.576\} * \max\{0, x_4 + 1.517\}$, there is an interaction between the predictors x_1 and x_4; so $r < s \Rightarrow r = 1$ and $s = 4$. The sum of selected first- and second-order derivatives of ψ_4 is then:

$$
\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \ \sum_{\substack{r<s \\ r,s \in V_4}} \left[D_{r,s}^{\boldsymbol{\alpha}} \boldsymbol{\psi}_4(\boldsymbol{t}^4) \right]^2 d\boldsymbol{t}^4,
$$

where

$$
\begin{aligned}
|\boldsymbol{\alpha}| = 1: \qquad & D^1_{1,4}\psi_4(\boldsymbol{t}^4) := \frac{\partial \psi_4}{\partial t^4_1}(\boldsymbol{t}^4) = \frac{\partial \psi_4}{\partial x_1}(x_1, x_4) = \max\{0, x_4 + 1.517\} \\
& \text{if } \ x_1 > -2.576 \ , \\
& D^1_{1,4}\psi_4(\boldsymbol{t}^4) := \frac{\partial \psi_4}{\partial t^4_1}(\boldsymbol{t}^4) = \frac{\partial \psi_4}{\partial x_1}(x_1, x_4) = 0 \ \text{ if } \ x_1 \leq -2.576 \ , \\
& D^1_{1,4}\psi_4(\boldsymbol{t}^4) := \frac{\partial \psi_4}{\partial t^4_2}(\boldsymbol{t}^4) = \frac{\partial \psi_4}{\partial x_4}(x_1, x_4) = \max\{0, x_1 + 2.576\} \\
& \text{if } \ x_4 > -1.517 \ , \\
& D^1_{1,4}\psi_4(\boldsymbol{t}^4) := \frac{\partial \psi_4}{\partial t^4_2}(\boldsymbol{t}^4) = \frac{\partial \psi_4}{\partial x_4}(x_1, x_4) = 0 \ \text{ if } \ x_4 \leq -1.517 \ , \\
|\boldsymbol{\alpha}| = 2: \qquad & D^2_{1,4}\psi_4(\boldsymbol{t}^4) := \frac{\partial^2 \psi_4}{\partial t^4_1 \partial t^4_2}(\boldsymbol{t}^4) = \frac{\partial^2 \psi_4}{\partial x_1 \partial x_4}(x_1, x_4) = 1 \ \text{ for all } \ x_4 > -1.517 \ , \\
& D^2_{1,4}\psi_4(\boldsymbol{t}^4) := \frac{\partial^2 \psi_4}{\partial t^4_1 \partial t^4_2}(\boldsymbol{t}^4) = \frac{\partial^2 \psi_4}{\partial x_1 \partial x_4}(x_1, x_4) = 0 \ \text{ for all } \ x_4 \leq -1.517 \ .
\end{aligned}
$$

For the BF $\psi_5(\boldsymbol{t}^5) = \max\{0, x_5 + 1.562\} {*} \max\{0, x_4 + 1.517\}$, there is an interaction between the predictors x_4 and x_5; so $r < s \Rightarrow r = 4$ and $s = 5$. The sum of selected first- and second-order derivatives of ψ_5 is obtained as:

$$
\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \ \sum_{\substack{r<s \\ r,s \in V_5}} \left[D^{\boldsymbol{\alpha}}_{r,s} \boldsymbol{\psi}_5(\boldsymbol{t}^5) \right]^2 d\boldsymbol{t}^5,
$$

where

$$
\begin{aligned}
|\boldsymbol{\alpha}| = 1: \qquad & D^1_{4,5}\psi_5(\boldsymbol{t}^5) := \frac{\partial \psi_5}{\partial t^5_1}(\boldsymbol{t}^5) = \frac{\partial \psi_5}{\partial x_4}(x_4, x_5) = \max\{0, x_5 + 1.562\} \\
& \text{if } \ x_4 > -1.517, \\
& D^1_{4,5}\psi_5(\boldsymbol{t}^5) := \frac{\partial \psi_5}{\partial t^5_1}(\boldsymbol{t}^5) = \frac{\partial \psi_5}{\partial x_4}(x_4, x_5) = 0 \ \text{ if } \ x_4 \leq -1.517, \\
& D^1_{4,5}\psi_5(\boldsymbol{t}^5) := \frac{\partial \psi_5}{\partial t^5_2}(\boldsymbol{t}^5) = \frac{\partial \psi_5}{\partial x_5}(x_4, x_5) = \max\{0, x_4 + 1.517\} \\
& \text{if } \ x_5 > -1.562, \\
& D^1_{4,5}\psi_5(\mathbf{t}^5) := \frac{\partial \psi_5}{\partial t^5_2}(\boldsymbol{t}^5) = \frac{\partial \psi_5}{\partial x_5}(x_4, x_5) = 0 \ \text{ if } \ x_5 \leq -1.562, \\
|\boldsymbol{\alpha}| = 2: \qquad & D^2_{4,5}\psi_5(\boldsymbol{t}^5) := \frac{\partial^2 \psi_5}{\partial t^5_1 \partial t^5_2}(\boldsymbol{t}^5) = \frac{\partial^2 \psi_5}{\partial x_4 \partial x_5}(x_4, x_5) = 1 \\
& \text{for all } \ x_5 > -1.562, \\
& D^2_{4,5}\psi_5(\boldsymbol{t}^5) := \frac{\partial^2 \psi_5}{\partial t^5_1 \partial t^5_2}(\boldsymbol{t}^5) = \frac{\partial^2 \psi_5}{\partial x_4 \partial x_5}(x_4, x_5) = 0 \\
& \text{for all } \ x_5 \leq -1.562.
\end{aligned}
$$

As a result, the PRSS objective function in (3.2.8) has the following form:

$$
\begin{aligned}
PRSS \ &= \ \underbrace{\sum_{i=1}^{N} \left(y_i - \boldsymbol{\theta}^T \boldsymbol{\psi}(\bar{\boldsymbol{d}}_i) \right)^2}_{=: I \ (RSS)} \\
&+ \ \underbrace{\sum_{m=1}^{M_{max}} \lambda_m \sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \ \sum_{\substack{r<s \\ r,s \in V_m}} \int \theta_m^2 \left[D^{\boldsymbol{\alpha}}_{r,s} \boldsymbol{\psi}_m(\boldsymbol{t}^m) \right]^2 d\boldsymbol{t}^m}_{=: II} .
\end{aligned}
$$

If $\lambda_1 = \lambda_2 = ... = \lambda_{M_{max}} =: \lambda$, then the Tikhonov regularization problem form of the function PRSS equation look as follows:

$$
PRSS \approx \underbrace{\left\| \boldsymbol{y} - \boldsymbol{\psi}(\bar{\mathbf{d}})\boldsymbol{\theta} \right\|_2^2}_{=I} + \underbrace{\lambda \left\| \boldsymbol{L}\boldsymbol{\theta} \right\|_2^2}_{\approx II} ,
$$

The first part of the PRSS objective function and of the Tikhonov regularization problem are equal as it is seen below. Note here that the second part is approximately equal:

$$\begin{aligned}
I \quad &: \sum_{i=1}^{N} \left(y_i - \boldsymbol{\theta}^T \boldsymbol{\psi}(\bar{\boldsymbol{d}}_i)\right)^2 = \left\| \boldsymbol{y} - \boldsymbol{\psi}(\bar{\boldsymbol{d}})\boldsymbol{\theta} \right\|_2^2 . \\
II \quad &: \sum_{m=1}^{M_{max}} \lambda_m \sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s \in V_m}} \int \theta_m^2 \left[D_{r,s}^{\boldsymbol{\alpha}} \boldsymbol{\psi}_m(\boldsymbol{t}^m) \right]^2 d\boldsymbol{t}^m \approx \lambda \left\| \boldsymbol{L}\boldsymbol{\theta} \right\|_2^2 .
\end{aligned}$$

The combination and approximation of the parts I and II are displayed next in our numerical example. The following values are such ones of *RSS*. For some illustration, a part of it is presented below. The whole RSS can be seen in Appendix A.

On I:

$$\begin{aligned}
\sum_{i=1}^{N} \left(y_i - \boldsymbol{\theta}^T \boldsymbol{\psi}(\bar{\boldsymbol{d}}_i)\right)^2 = &(-1.1224 - \theta_0 - (\max\{0, -0.6109 + 0.159\})\,\theta_1 - \\
&(\max\{0, -0.159 + 0.6109\})\,\theta_2 - \\
&(\max\{0, -0.5172 + 1.517\})\,\theta_3 - \\
&(\max\{0, -0.0781 + 2.576\} * \max\{0, -1.5172 + 1.517\})\,\theta_4 - \\
&(\max\{0, -0.8184 + 1.562\} * \max\{0, -1.5172 + 1.517\})\,\theta_5)^2 + \\
&(-0.8703 - \theta_0 - (\max\{0, -0.5885 + 0.159\})\,\theta_1 - \\
&(\max\{0, -0.159 + 0.5885\})\,\theta_2 - \\
&(\max\{0, -1.3501 + 1.517\})\,\theta_3 - \\
&(\max\{0, -0.0781 + 2.576\} * \max\{0, -1.3501 + 1.517\})\,\theta_4 - \\
&(\max\{0, -0.8184 + 1.562\} * \max\{0, -1.3501 + 1.517\})\,\theta_5)^2 + \\
&\vdots \\
&(3.5314 - \theta_0 - (\max\{0, 4.3884 + 0.159\})\,\theta_1 - \\
&(\max\{0, -0.159 - 4.3884\})\,\theta_2 - \\
&(\max\{0, 1.0942 + 1.517\})\,\theta_3 - \\
&(\max\{0, 2.4197 + 2.576\} * \max\{0, 1.0942 + 1.517\})\,\theta_4 - \\
&(\max\{0, -1.5624 + 1.562\} * \max\{0, 1.0942 + 1.517\})\,\theta_5)^2 .
\end{aligned}$$

According to the values obtained by computing the maximum functions, the RSS term has the following form:

$$
\begin{aligned}
\sum_{i=1}^{N}\left(y_i - \boldsymbol{\theta}^T \boldsymbol{\psi}(\bar{\boldsymbol{d}}_i)\right)^2 &= \left(-1.1224 - \theta_0 - 0.4519\theta_2\right)^2 + \\
\left(-0.8703 - \theta_0 - 0.4295\theta_2 - 0.1669\theta_3 - 0.4169\theta_4 - 0.3973\theta_5\right)^2 &+ \\
&\vdots \\
\left(3.5314 - \theta_0 - 4.5474\theta_1 - 2.6112\theta_3 - 13.0448\theta_4\right)^2
\end{aligned}
$$

$$
\begin{aligned}
&= \left(-1.1242 - \theta_0 - 0.4519\theta_2\right)^T \left(-1.1242 - \theta_0 - 0.4519\theta_2\right) + \\
&\left(-0.8703 - \theta_0 - 0.4295\theta_2 - 0.1669\theta_3 - 0.4169\theta_4 - 0.3973\theta_5\right) + \\
&\left(-0.8703 - \theta_0 - 0.4295\theta_2 - 0.1669\theta_3 - 0.4169\theta_4 - 0.3973\theta_5\right) + \\
&\vdots \\
&\left(3.5314 - \theta_0 - 4.5474\theta_1 - 2.6112\theta_3 - 13.0448\theta_4\right) + \\
&\left(3.5314 - \theta_0 - 4.5474\theta_1 - 2.6112\theta_3 - 13.0448\theta_4\right).
\end{aligned}
$$

If we turn the above summation into vector notation, we get the subsequent representation. By this, we have found the value of the first part of PRSS, which is RSS:

$$
\begin{aligned}
\sum_{i=1}^{N}\left(y_i - \boldsymbol{\theta}^T \boldsymbol{\psi}(\bar{\boldsymbol{d}}_i)\right)^2 &= \left(\boldsymbol{y} - \boldsymbol{\psi}(\bar{\boldsymbol{d}})\boldsymbol{\theta}\right)^T \left(\boldsymbol{y} - \boldsymbol{\psi}(\bar{\boldsymbol{d}})\boldsymbol{\theta}\right) \\
&= \left\|\boldsymbol{y} - \psi(\bar{\boldsymbol{d}})\boldsymbol{\theta}\right\|_2^2. \qquad (3.3.22)
\end{aligned}
$$

On II: The multi-dimensional integral in the second part of the equation in (3.2.9) takes the form in (3.2.12) by discretizing. The discretized form is denoted by $\boldsymbol{L}$ and finally we obtain the formulation from (3.2.15). In order to apply this discretization, we sort the data set used in the numerical example. We slightly decrease the input data value of each first predictor variable and slightly increase the input data value of each last predictor variable. That means by adding two new observations, we get a new data set. In this case,

x_1: the first discretization value of x_1 is $\bar{x}_{0,1} = -3.0$, the last discretization value is $\bar{x}_{33,1} = 3.0$.
x_2: the first discretization value of x_2 is $\bar{x}_{0,2} = -0.7$, the last discretization value is $\bar{x}_{33,2} = 5.0$.
x_3: the first discretization value of x_3 is $\bar{x}_{0,3} = -3.5$, the last discretization value is $\bar{x}_{33,3} = 1.5$.
x_4: the first discretization value of x_4 is $\bar{x}_{0,4} = -2.0$, the last discretization value is $\bar{x}_{33,4} = 2.0$.
x_5: the first discretization value of x_5 is $\bar{x}_{0,5} = -2.0$, the last discretization value is $\bar{x}_{33,5} = 2.5$.

The numbers L_{im} applied to our numeric example, corresponding to each BF, are as follows:

$$\sum_{i=1}^{(33)^{K_1}} \underbrace{\left[\left(\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_1}} \left[D^{\boldsymbol{\alpha}}_{r,s} \left(\max\left\{0, x_2 + 0.159\right\} \right) \right]^2 \right) \left(\bar{x}_{l^{\kappa_1^1}_{\sigma^{\kappa_1}+1},\kappa_1^1} - \bar{x}_{l^{\kappa_1^1}_{\sigma^{\kappa_1}},\kappa_1^1} \right) \right]}_{=L_{i1}}.$$

The value of L_1 is 1.9545.

$$\sum_{i=1}^{(33)^{K_2}} \underbrace{\left[\left(\sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s \\ r,s\in V_2}} \left[D^{\boldsymbol{\alpha}}_{r,s} \left(\max\left\{0, -0.159 - x_2\right\} \right) \right]^2 \right) \left(\bar{x}_{l^{\kappa_1^2}_{\sigma^{\kappa_1}+1},\kappa_1^2} - \bar{x}_{l^{\kappa_1^2}_{\sigma^{\kappa_1}},\kappa_1^2} \right) \right]}_{=L_{i2}}.$$

The value of L_2 is 0.5999.

$$\sum_{i=1}^{(33)^{K_3}} \underbrace{\left[\left(\sum_{\substack{|\boldsymbol{\alpha}|=1\\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s\\ r,s\in V_3}} \left[D_{r,s}^{\boldsymbol{\alpha}}\left(\max\left\{0, x_4+1.517\right\}\right)\right]^2\right)\left(\bar{x}_{l^{\kappa_1^3}_{\sigma^{\kappa_1}+1,\kappa_1^3}} - \bar{x}_{l^{\kappa_1^3}_{\sigma^{\kappa_1},\kappa_1^3}}\right)\right]}_{=L_{i3}}.$$

The value of L_3 is 2.0622.

$$\sum_{i=1}^{(33)^{K_4}} \underbrace{\left[\left(\sum_{\substack{|\boldsymbol{\alpha}|=1\\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s\\ r,s\in V_4}} \left[D_{r,s}^{\boldsymbol{\alpha}}\boldsymbol{\psi}_4(\boldsymbol{t}^4)\right]^2\right)\left(\bar{x}_{l^{\kappa_1^4}_{\sigma^{\kappa_1}+1,\kappa_1^4}} - \bar{x}_{l^{\kappa_1^4}_{\sigma^{\kappa_1},\kappa_1^4}}\right)\cdot\left(\bar{x}_{l^{\kappa_2^4}_{\sigma^{\kappa_2}+1,\kappa_2^4}} - \bar{x}_{l^{\kappa_2^4}_{\sigma^{\kappa_1},\kappa_2^4}}\right)\right]}_{=L_{i4}}.$$

Here, $\boldsymbol{\psi}_4(\boldsymbol{t}^4) = (\max\{0, x_1+2.576\} * \max\{0, x_4+1.517\})$. The value of L_4 is 1.6002.

$$\sum_{i=1}^{(33)^{K_5}} \underbrace{\left[\left(\sum_{\substack{|\boldsymbol{\alpha}|=1\\ \boldsymbol{\alpha}=(\alpha_1,\alpha_2)^T}}^{2} \sum_{\substack{r<s\\ r,s\in V_5}} \left[D_{r,s}^{\boldsymbol{\alpha}}\boldsymbol{\psi}_5(\boldsymbol{t}^5)\right]^2\right)\left(\bar{x}_{l^{\kappa_1^5}_{\sigma^{\kappa_1}+1,\kappa_1^5}} - \bar{x}_{l^{\kappa_1^5}_{\sigma^{\kappa_1},\kappa_1^5}}\right)\cdot\left(\bar{x}_{l^{\kappa_2^5}_{\sigma^{\kappa_5}+1,\kappa_2^5}} - \bar{x}_{l^{\kappa_2^5}_{\sigma^{\kappa_1},\kappa_2^5}}\right)\right]}_{L_{i5}}.$$

Here, $\boldsymbol{\psi}_5(\boldsymbol{t}^5) = (\max\{0, x_5+1.562\} * \max\{0, x_4+1.517\})$. The value of L_5 is 13.1962.

The matrix $\boldsymbol{L}$ is a (6×6)-diagonal matrix. Its first column values are zero and the diagonal values of this matrix are $\boldsymbol{L}_m$ $(m = 1, 2, ..., 5)$ which are introduced above. The matrix $\boldsymbol{L}$ of our numerical example is presented below:

$$\boldsymbol{L} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.9545 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5999 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.0622 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.6002 & 0 \\ 0 & 0 & 0 & 0 & 0 & 13.1962 \end{bmatrix}.$$

In the equation (3.15), $\|\boldsymbol{L\theta}\|_2^2$ is the squared norm of

$$\boldsymbol{L\theta} := \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.9545 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5999 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.0622 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.6002 & 0 \\ 0 & 0 & 0 & 0 & 0 & 13.1962 \end{bmatrix} \cdot \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \end{bmatrix} = \begin{bmatrix} 0 \\ \theta_1 \cdot (1.9545) \\ \theta_2 \cdot (0.5999) \\ \theta_3 \cdot (2.0622) \\ \theta_4 \cdot (1.6002) \\ \theta_5 \cdot (13.1962) \end{bmatrix}.$$

$$\begin{aligned} \|\mathbf{L}\boldsymbol{\theta}\|_2^2 &= (\theta_1 \cdot (1.9545))^2 + (\theta_2 \cdot (0.5999))^2 + (\theta_3 \cdot (2.0622))^2 + (\theta_4 \cdot (1.6002))^2 \\ &+ (\theta_5 \cdot (13.1962))^2 . \end{aligned} \tag{3.3.23}$$

From the equations (3.3.13) and (3.3.14), we obtain the objective function PRSS for the numerical example. In the previous section, we mention that PRSS is the Tikhonov regularization problem. In order to solve this problem, we can easily formulate PRSS as a CQP problem as follows:

$$\begin{aligned} &\min_{t,\boldsymbol{\theta}} \quad t, \\ &\text{subject to} \quad \left\| \boldsymbol{\psi}(\bar{\boldsymbol{d}})\boldsymbol{\theta} - \boldsymbol{y} \right\|_2 \leq t, \\ &\qquad\qquad \|\boldsymbol{L\theta}\|_2 \leq \sqrt{\bar{M}}. \end{aligned} \tag{3.3.24}$$

Although PRSS and CQP problem have different notations, they have the same solution for appropriate choice of the values λ and $\sqrt{M}$. If we decrease the values of λ and $\sqrt{M}$ a bit, then the minimum value of $\left\|\boldsymbol{\psi}(\bar{\boldsymbol{d}})\boldsymbol{\theta}-\boldsymbol{y}\right\|_2$ increases for both minimization problem (PRSS and CQP). While for CQP an interior point method is used, for PRSS *generalized singular value decomposition (GSVD)* is employed for solving problem [2, 47].

In our numerical example, this CQP problem can be written as follows:

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

subject to

$$\begin{aligned}
-1.1224-\theta_0-0.4519\theta_2 &= \theta_6,\\
-0.8703-\theta_0-0.42295\theta_2-0.1669\theta_3-0.4169\theta_4-0.3973\theta_5 &= \theta_7,\\
-0.9549-\theta_0-0.42295\theta_2-0.1669\theta_3-0.4169\theta_4-0.3973\theta_5 &= \theta_8,\\
-0.8703-\theta_0-0.42295\theta_2-0.1669\theta_3-0.4169\theta_4-0.4966\theta_5 &= \theta_9,\\
-0.9549-\theta_0-0.42295\theta_2-0.1669\theta_3-0.5960\theta_5 &= \theta_{10},\\
-0.8703-\theta_0-0.42295\theta_2-0.1669\theta_3-0.5960\theta_5 &= \theta_{11},\\
-1.0396-\theta_0-0.3347\theta_2-0.5221\theta_3-1.3042\theta_4-0.6213\theta_5 &= \theta_{12},\\
-0.447-\theta_0-0.2873\theta_2-0.6892\theta_3-1.7216\theta_4-0.8201\theta_5 &= \theta_{13},\\
-0.701-\theta_0-0.2873\theta_2-0.6892\theta_3-1.7216\theta_4-0.8201\theta_5 &= \theta_{14},\\
-0.6163-\theta_0-0.2873\theta_2-0.6892\theta_3-1.7216\theta_4-0.3973\theta_5 &= \theta_{15},\\
-0.447-\theta_0-0.2873\theta_2-0.6892\theta_3-1.7216\theta_4-1.6406\theta_5 &= \theta_{16},\\
-0.6163-\theta_0-0.2873\theta_2-0.6892\theta_3-1.7216\theta_4-1.6406\theta_5 &= \theta_{17},\\
-0.447-\theta_0-0.2873\theta_2-0.6892\theta_3-1.7216\theta_4-1.6406\theta_5 &= \theta_{18},\\
-0.1085-\theta_0-0.2789\theta_2-1.7755\theta_3-4.4350\theta_4-2.1128\theta_5 &= \theta_{19},\\
-0.1085-\theta_0-0.0557\theta_2-1.5666\theta_3-3.9132\theta_4-1.8643\theta_5 &= \theta_{20},\\
-0.0238-\theta_0-0.0557\theta_2-1.7755\theta_3-4.4350\theta_4-2.1128\theta_5 &= \theta_{21},\\
-0.1931-\theta_0-0.0001\theta_1-1.7755\theta_3-4.4350\theta_4-2.1128\theta_5 &= \theta_{22},
\end{aligned}$$

$$
\begin{aligned}
-0.1085 - \theta_0 - 0.0001\theta_1 - 1.7755\theta_3 - 4.4350\theta_4 - 2.1128\theta_5 &= \theta_{23},\\
-0.1931 - \theta_0 - 0.0001\theta_1 - 1.7755\theta_3 - 4.4350\theta_4 - 2.1128\theta_5 &= \theta_{24},\\
-0.1085 - \theta_0 - 0.0001\theta_1 - 1.7755\theta_3 - 4.4350\theta_4 - 4.2264\theta_5 &= \theta_{25},\\
0.0680 - \theta_0 - 0.0001\theta_1 - 1.7755\theta_3 - 4.4350\theta_4 - 4.2264\theta_5 &= \theta_{26},\\
-0.0238 - \theta_0 - 0.0001\theta_1 - 2.6112\theta_3 - 6.5225\theta_4 - 6.2157\theta_5 &= \theta_{27},\\
0.2301 - \theta_0 - 0.2233\theta_1 - 2.6112\theta_3 - 6.5225\theta_4 - 3.1073\theta_5 &= \theta_{28},\\
0.3148 - \theta_0 - 0.2233\theta_1 - 2.6112\theta_3 - 6.5225\theta_4 &= \theta_{29},\\
0.1455 - \theta_0 - 0.2233\theta_1 - 2.6112\theta_3 - 6.5225\theta_4 - 1.5531\theta_5 &= \theta_{30},\\
0.4841 - \theta_0 - 0.2233\theta_1 - 2.9454\theta_3 - 7.3573\theta_4 - 5.2581\theta_5 &= \theta_{31},\\
0.5687 - \theta_0 - 0.3153\theta_1 - 2.6112\theta_3 - 6.5225\theta_4 - 3.1073\theta_5 &= \theta_{32},\\
1.0766 - \theta_0 - 0.5022\theta_1 - 2.6112\theta_3 - 13.0448\theta_4 &= \theta_{33},\\
1.1613 - \theta_0 - 0.5022\theta_1 - 2.6112\theta_3 - 6.5225\theta_4 &= \theta_{34},\\
0.738 - \theta_0 - 0.5022\theta_1 - 2.6112\theta_3 - 6.5225\theta_4 - 4.6615\theta_5 &= \theta_{35},\\
2.5156 - \theta_0 - 2.8735\theta_1 - 2.6112\theta_3 - 13.0448\theta_4 &= \theta_{36},\\
3.5314 - \theta_0 - 4.5474\theta_1 - 2.6112\theta_3 - 13.0448\theta_4 &= \theta_{37},
\end{aligned}
$$

$$
\begin{aligned}
(\theta_6^2 + \theta_7^2 + \theta_8^2 + \theta_9^2 + \theta_{10}^2 + \theta_{11}^2 + \theta_{12}^2 + \theta_{13}^2 + \theta_{14}^2 + \theta_{15}^2 + \theta_{16}^2 + \theta_{17}^2 + \theta_{18}^2 +&\\
\theta_{19}^2 + \theta_{20}^2 + \theta_{21}^2 + \theta_{22}^2 + \theta_{23}^2 + \theta_{24}^2 + \theta_{25}^2 + \theta_{26}^2 + \theta_{27}^2 + \theta_{28}^2 + \theta_{29}^2 + \theta_{30}^2 +&\\
\theta_{31}^2 + \theta_{32}^2 + \theta_{33}^2 + \theta_{34}^2 + \theta_{35}^2 + \theta_{36}^2 + \theta_{37}^2)^{1/2} \leq t,&
\end{aligned}
$$

$$
\big(\theta_{38}^2 + \theta_{39}^2 + \theta_{40}^2 + \theta_{41}^2 + \theta_{42}^2 + \theta_{43}^2\big)^{1/2} \leq (\bar{M})^{1/2}.
$$

This problem involves 32 linear constraints and two quadratic cones. In equation (3.3.15), our numerical problem has only two quadratic cones. For solving our numerical problem, we transform it into the MOSEK format. For this transformation, we attribute new unknown variables to the linear notations in these two quadratic cones. By this way, we simplify the notations in the cones and write them as constraints. MOSEK uses an interior-point optimizer as a default for the CQP problem. The interior-point optimizer is an implementation of the homogeneous and self-dual algorithm and it computes the interior point solution which is an arbitrary optimal solution.

The values $\sqrt{\bar{M}}$ in our optimization problem are determined by a *model-free* (train and error) method. When we access the $\sqrt{\bar{M}}$ values in our C-MARS code, C-MARS provides us several solutions, each of them based on the five BFs.

In the next section, we apply C-MARS to different sizes and types of data sets. The results obtained from the algorithms C-MARS and MARS are also compared according to many different general *performance comparison* criteria.

CHAPTER 4

APPLICATIONS

In the previous section, MARS and C-MARS have been presented and investigated in detail. In this section for comparing these methods, different data sets are used in the applications. While Salford Systems is used for MARS application [39], for C-MARS a code is written by using MATLAB and in order to solve the CQP problem in C-MARS, MOSEK software is preferred.

4.1 Description of Data Sets Used in Applications

Three data sets are used in the applications.

Data Set 1: The first data set, *Latin Hypercube Sampling (LHS)*, is obtained by means of design of experiments performed on solid rocket motors. It contains 389 observations and ten predictor variables which are design variables for performance of solid rocket motors such as radius of grain, burn rate constant and density of propellant. The response variable is a total impulse. In this data set, the type of input variables is quantitative. The data are preprocessed for all missing values, inconsistency and outliers. The 335^{th} sample of this data set is an outlier. The matrix plot of the response variable versus predictor variables of LHS data can be seen in Appendix B. According to this matrix plot, although we can see a weak relation between predictor variables x_4, x_8 and response variable, we can not find a distinctive relation between response variable and other predictor variables. However, we can say that it is reasonable to look for such a relation between variables for this data set. For detailed information about LHS data see Kartal, E., 2007 [30].

Data Set 2: Our second data set is *Uniform Sampling (US)* which is also obtained by means of design of experiments performed on solid rocket motors. It has seven predictor variables and its sample size is 100. The input variables of this data set are design variables for performance of solid rocket motors. The outcome variable is a total impulse for this data set. The type of both input variables and response variable is quantitative. The same data preprocessing is used for the US data, too. The 78^{th} sample of this data set is an outlier. The matrix plot of the response variable versus predictor variables of US data can be seen in Appendix B. As in the case of the first data set, there is no distinctive relation between response variable and input variables according to the matrix plot, except the relation between x_4 and response variable. We try to find out a reasonable relation between variables. For detailed information about US data you can refer to Kartal, E., 2007 [30].

Data Set 3: The last data set consists of real-world data provided by a manifacturing company from the metal casting industry. It includes 34 predictor variables and 92 observations. The input variables are process and product parameters. The response variable of the real-world data is a percent defective of production. All variables are quantitative. This data set is handled according to all missing values, inconsistancy and outliers. There is no outlier for this data set. The matrix plot of response variable versus predictor variables of the third data can be found in Appendix B. For this data set, variables start from x_2. Although there is no remarkable relation between response variable and input variables, in this data set it can be searched for a reasonable relation between variables. More detailed information about this data set can be found in this study Bakır, B., 2006 [4].

4.2 Validation Approach and Comparison Measures

In our applicatios, to compare the methods we prefer to use a *3-times replicated 3-fold cross validation (CV)* approach. In 3-fold CV, the original data are randomly divided into three sub-samples (folds). While a single sub-sample is retained as the data for testing the model, the remaining two sub-samples are used as training data. This process is then repeated three times; thus, each of the three sub-samples is used exactly once as the test data. To produce a single estimate for each measure, the three results from the folds can be averaged. Since the proportion of labels in the response variable is not equal, there is a possibility that a given fold may not contain one of the labels. To guarantee that this does not happen, a *stratified 3-fold CV* is used where each fold includes roughly the same proportion of class labels as in the original set of data. Moreover, to increase the reliability of the model, the CV process is replicated three times, each time with a new partitioning.

To evaluate the performance of MARS and C-MARS methods, several measures can be used. The performance measures that we used in our applications and their general notation 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 fitted 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 $\boldsymbol{X}$ is $(N \times p)$ design matrix and $\text{rank}(\boldsymbol{X}) = p$ $(p \leq N)$.

Adjusted $\boldsymbol{R}^2$

Accounts for the number of predictors in your model and is useful for comparing models with different numbers of predictors. The higher the *Adjusted* $\boldsymbol{R}^2$ (Adj-$\boldsymbol{R}^2$), the better the model fits your data. The formula is:

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

where $(N - p - 1) \neq 0$.

$\boldsymbol{R}^2$

This value is a coefficient of determination; it indicates how much variation in response is explained by the model. The higher the R^2, the better the model fits your data. The formula is:

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

Mean Absolute Error (MAE)

MAE measures the average magnitude of error. The smaller MAE, the better it is. The formula is:

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

Mean Absolute Percentage Error (MAPE)

MAPE represents the scale independent (relative) error. The smaller MAPE, the better it is. The formula is:

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

Mean Square Error (MSE)

MSE emphasizes the grossly inaccurate estimates. The smaller MSE, the better it is. The formula is:

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

Root Mean Square Error (RMSE)

RMSE measures the magnitude with more weight on grossly inaccurate estimates. The smaller RMSE, the better it is. A model independent formula is:

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

Correlation Coefficient

A *correlation coefficient* is a measure of linear association between actual and predicted response values. The formula is:

$$r := \frac{\sum_{i=1}^{n}(y-\bar{y})(\hat{y}-\bar{\hat{y}})/(n-1)}{\sqrt{s(y)^2 s(\hat{y})^2}}.$$

Prediction error sum of squares (PRESS)

PRESS is an assessment of your model's predictive ability. PRESS, similar to the residual sum of squares, is the sum of squares of the prediction error. In general, the smaller the PRESS value, the better the model's predictive ability. In least squares regression, PRESS is calculated with the following formula:

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

Predicted $\boldsymbol{R}^2$

The *predicted* R^2 indicates how well the model predicts responses for new observations. Larger values of predicted R2 suggest models of greater predictive ability.

The higher predicted R^2, the better it is. The formula is:

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

Mallows' Cp

Mallows' Cp is a measure of the goodness-of-prediction. The formula is:

$$Cp := (RSS_p/MSE_m) - (N - 2p).$$

Here, SSE_p is SSE for the model under consideration; MSE_m is the mean square error for the model with all predictors included. In general, we look for models where Mallows' Cp is less than or equal to p. A small Cp value indicates that the model is relatively precise (has small variance) in estimating the true regression coefficients and predicting future responses. Models with poor predictive ability and bias have values of Cp larger than p.

Many of these measures can be found in any statistic text book such as Mendenhall and Sincich (2003) [38].

Proportion of Responses Within Some User-specified Range (PWI)

PWI is the proportion of responses within some user-specified range is the sum of indicator variables over all observations. The indicator variables take the value of one if the absolute value of the difference between actual and predicted response is within some user-specified thresholds [43].

Stability

The prediction model obtained from the methods is stable when it performs just as well on both seen (training) and unseen (test) data sets. The stability can be measured as a positive or negative number between 0 and 1 (or -1), where 0 means completely stable and -1 or 1 means completely unstable. This value is calculated for all measures. Stability can be calculated as the arithmetic difference divided by the arithmetic sum of training and test of performance criterion [26, 27]:

$$(CR_{TR} - CR_{TE}) - (CR_{TR} + CR_{TE}).$$

4.3 Construction of Models

As we mentioned before, MARS algorithm creates the best model by using two step-wise stages: forward and backward. The obtained models having different numbers of BFs and interaction terms are trained in CV analysis. The best model is selected among the models with minimum GCV and the highest Adjusted R^2. By using these two criteria, nine best models are generated for MARS.

In order to construct C-MARS models, we use the BFs of the large model of MARS produced by the forward step-wise algorithm.

We access the BFs of the large model and choose $\sqrt{M}$ to our C-MARS code. As you remember, $\sqrt{M}$ is a boundary value for CQP and this value is determined by training and error. C-MARS algorithm provides us many different models without identifying the best one.

After developing both MARS and C-MARS models for all training data sets, their performances are compared with respect to the following criteria:

- $\|\boldsymbol{L\theta}\|_2$ versus $SQRT(RSS)$,
- GCV, and
- $\|\boldsymbol{L\theta}\|_2$.

To compare MARS and C-MARS, we choose three representative solutions, S1, S2 and S3, provided by the developed program. Here, S1 is the best solution among the ones that respect to goodness of fit and S3 is the best solution for $SQRT(RSS)$. In order to determine S2, we plotted a log-log scale curve of $\|\boldsymbol{L\theta}\|_2$ and $SQRT(RSS)$ of values of the solutions obtained from CQP problem (3.2.18). It has a characteristic L shape. The sharpness of the corner value is the S2 solution [2].

We applied this procedure to all of our data sets. In order to see the results, we present one of the applications on data set 1. For the first replication and first fold

(CV) of this data set, Table 4.1 and Table 4.2 show the results of Salford MARS and C-MARS according to the $SQRT(RSS)$, $\|\boldsymbol{L\theta}\|_2$ and GCV are obtained from MARS and C-MARS.

Table 4.1: The results of Salford MARS.

No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
1	8.3751	2.2017	0.2750	0.9698
2	5.5860	1.8111	0.1252	0.9475
3	4.9931	2.1467	0.1049	0.9036
4	4.4397	2.1621	0.0857	0.8749
5	3.7117	2.2089	0.0619	0.8467
6	3.2703	2.2486	0.0497	0.8189
7	3.0401	2.2508	0.0444	0.7916
8	2.6179	2.1476	0.0341	0.7648
9	2.1788	2.1373	0.0244	0.7384
10	1.7619	2.1328	0.0166	0.7125
11	1.5087	2.2359	0.0126	0.6871
12	1.2909	2.1778	0.0096	0.6621
13	1.1557	2.1504	0.0080	0.6376
14	1.0271	2.1220	0.0065	0.6135
15	0.9639	2.0372	0.0060	0.5899
16	0.9096	2.0411	0.0055	0.5668
17	0.8691	2.0333	0.0053	0.5441
18	0.8584	2.0407	0.0054	0.5219
19	0.8528	2.0476	0.0055	0.5002
20	0.8501	2.0511	0.0057	0.4789
21	0.8480	2.0515	0.0060	0.4581
22	0.8480	2.0515	0.0062	0.4377

No. BF: number of basis function, Denominator: denominator of GCV.

As it is seen from Table 4.1, MARS generates 22 solutions and the 17th solution is the best one. Its GCV value is 0.0053 and Adjusted R^2 is 0.997.

In the following Table 4.2, C-MARS results are presented according to the same measures and values of $\sqrt{M}$.

Table 4.2: The results of C-MARS.

$\sqrt{M}$	No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
0.265	22	6.5015	0.265	0.3672	0.4377
0.3	22	6.2446	0.3	0.3387	0.4377
0.35	22	5.9403	0.35	0.3065	0.4377
0.4	22	5.6755	0.4	0.2798	0.4377
0.45	22	5.434	0.45	0.2565	0.4377
0.5	22	5.2083	0.5	0.2356	0.4377
0.55	22	4.9941	0.55	0.2167	0.4377
0.6	22	4.7889	0.6	0.1992	0.4377
0.7	22	4.3994	0.7	0.1681	0.4377
0.8	22	4.0313	0.8	0.1412	0.4377
0.9	22	3.6799	0.9	0.1176	0.4377
1	22	3.3424	1	0.097	0.4377
1.1	22	3.0171	1.1	0.0791	0.4377
1.2	22	2.7031	1.2	0.0635	0.4377
1.25	22	2.5501	1.25	0.0565	0.4377
1.3	22	2.3999	1.3	0.05	0.4377
1.4	22	2.1081	1.4	0.0386	0.4377
1.5	22	1.8292	1.5	0.0291	0.4377
1.6	22	1.5661	1.6	0.0213	0.4377
1.7	22	1.3243	1.7	0.0152	0.4377
1.8	22	1.1138	1.8	0.0108	0.4377
1.9	22	0.9512	1.9	0.0079	0.4377
2	22	0.86	2	0.0064	0.4377
2.1	22	0.8478	2.0509	0.0062	0.4377
2.2	22	0.8478	2.0509	0.0062	0.4377
2.3	22	0.8478	2.0509	0.0062	0.4377

No. BF: number of basis function, Denominator: denominator of GCV.

C-MARS provides many solutions, each one having 22 basis functions.

Let us consider only three solutions of C-MARS. Here, S1 solution is the best for $\|\boldsymbol{L\theta}\|_2$ and worst for $SQRT(RSS)$; S3 solution is the best for $SQRT(RSS)$ and worst for $\|\boldsymbol{L\theta}\|_2$. S2 solution is the minimizing solution for PRSS. It is obtained by plotting a log-log scale curve of values of $\|\boldsymbol{L\theta}\|_2$ versus $SQRT(RSS)$.

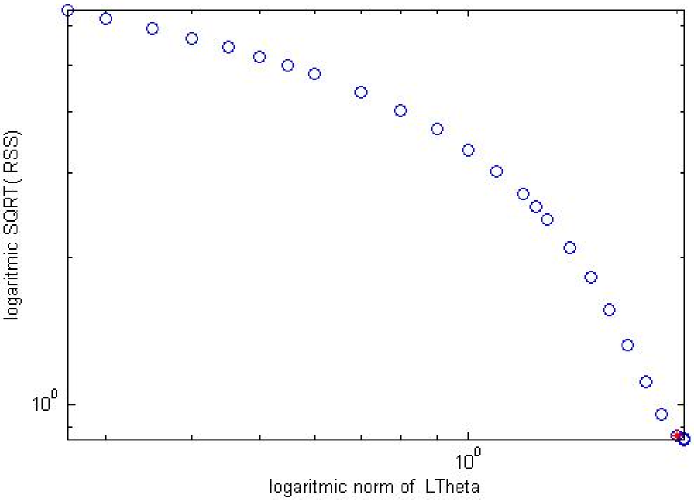

Figure 4.1: A log-log scale, the curve of norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS).

The corner of the L curve in Figure 4.1, demonstrated with red point, represents S2 solution of C-MARS.

C-MARS solutions obtained by using CQP change according to the changes in the values of $\sqrt{M}$. The soltions S1, S2 and S3 are extreme solutions and the values of $\sqrt{M}$ for S1, S2 and S3 are 0. 265, 2, 2.3 respectively. For appropriate choices of λ and $\sqrt{M}$, PRSS and CQP are equivalent.

Figure 4.2 shows the $\|\boldsymbol{L\theta}\|_2$ versus $SQRT(RSS)$ for MARS and C-MARS solutions. These two objectives are taken into account with respect to provide the minimization of the mentioned objectives. As it is expected, from Figure 4.2 we see that when the value of $\|\boldsymbol{L\theta}\|_2$ gets better (decreases), the vaue of $SQRT(RSS)$ gets worst (increases). C-MARS solutions dominate MARS solutions according to $\|\boldsymbol{L\theta}\|_2$ and $\|RSS\|_2$.

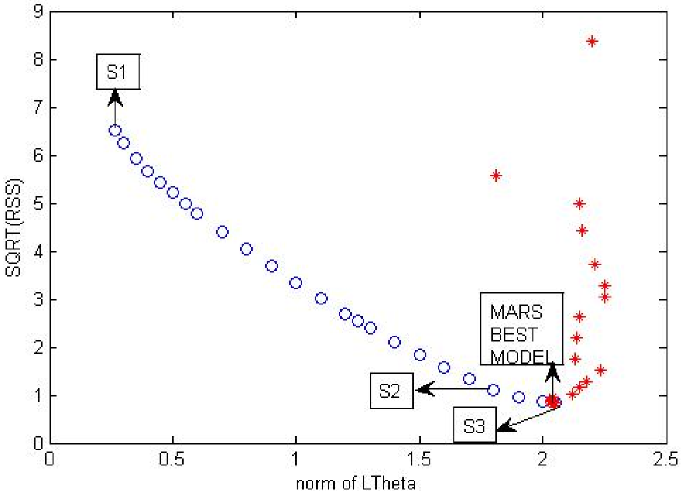

Figure 4.2: Norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for the solutions of methos (*: MARS solutions; o: C-MARS solutions).

Figure 4.3 shows the GCV values of the C-MARS solutions (S1, S2 and S3) and MARS solutions. As it is mentioned in Section 3, the best model is one that has the minimum GCV value. According to the GCV, all MARS models dominate the solutions of our problem (3.2.18).

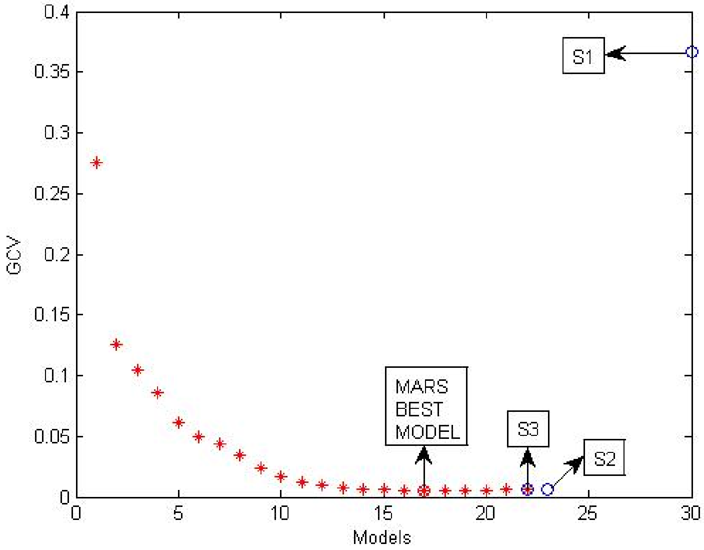

Figure 4.3: GCV values for the solutions of methos (*: MARS solutions; o: C-MARS solutions).

Figure 4.4 indicates the $\|\boldsymbol{L\theta}\|_2$ values of the methods solutions. The model having minimum value of $\|\boldsymbol{L\theta}\|_2$ is considered as the best solution. With regard to $\|\boldsymbol{L\theta}\|_2$, MARS solutions are dominated by C-MARS solutions. This means that C-MARS solutions have lower $\|\boldsymbol{L\theta}\|_2$ values.

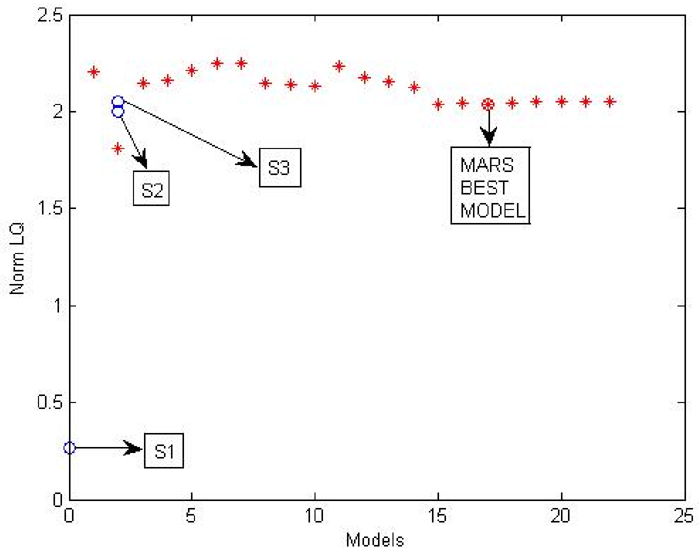

Figure 4.4: Norm of $\boldsymbol{L\theta}$ for the solutions of methos (*: MARS solutions; o: C-MARS solutions).

According to the employed measures, the remained replications and folds CVs for the first data set indicate the same results. The related tables and figures can be found in Appendix C. We notice that according to the above comparisons (GCV and $\|\boldsymbol{L\theta}\|_2$), we can not see any significant differences between the performances of these two methods. As it is seen from the above Figure 4.2, Figure 4.3 and Figure 4.4, each method has a better performances with respect to their own criteria. In other words, while according to $\|\boldsymbol{L\theta}\|_2$, C-MARS has a better performance, according to GCV, MARS has a better performance. Because of this, the models for all test data sets are also compared according to the method-free measures such as MSE, Adjusted-R^2, Mallow's Cp, Correlation Coefficient (Cor. Coeff.), etc., as presented in Section 4.2. The comparison measures are based on the average of nine values (one for each fold and each replication) and the stability of measures obtained from the training and test results are in Table 4.3.

Table 4.3: Averages of performance measure values for the models and stability of measures for LHS data.

Measures	MARS	S1	S2	S3
MAE	0.057467	0.33404	0.082378	0.057456
MSE	0.006133333	0.19782	0.019978	0.006378
RMSE	0.078022222	0.42921	0.111133	0.079511
MAPE	20.49863333	81.9345	26.61973	20.89033
Cor. Coeff.	0.997022222	0.89832	0.9912	0.996933
R^2	0.994	0.81077	0.982756	0.993878
Adj-R^2	0.992822222	0.75593	0.975567	0.992189
PWI-1	0.944611111	0.95477	0.942033	0.936911
PWI-2	0.986377778	0.99486	0.987244	0.983811
Press	2.339188889	20.8966	3.178644	2.706878
R^2-Pred	1.017355556	1.16516	1.023656	1.020044
Mallows Cp	35.5427	49.5556	49.55556	49.55556
Stability MSE	-0.291855556	-0.05214	-0.27894	-0.31692
Stability Cor. Coeff.	0.000666667	0.003211111	0.000777778	0.000744444
Stability R^2	0.001377778	0.0064	0.001511	0.001433
Stability Adj-R^2	0.001822222	0.025411111	0.004166667	0.002133333
Stability PWI-1	0.003211111	-0.00172	0.001367	0.004556
Stability PWI-2	0.003455556	-0.090955556	0.002144444	0.004988889
Stability Press	-0.999866667	-0.78309	-0.83231	-0.94402
Stability R^2-Pred	-0.008366667	-0.05284	-0.01132	0.085233
Stability Mallows Cp	0.1717	0	0	0.090911

When we consider the results in Table 4.3 with respect to the fit measures such as MSE, MAPE, R^2, etc., and to the complexity measure Adjusted R^2, the best solution of the MARS and S3 solution of C-MARS have a better performance. According to the PWI-1 and PWI-2, S1 solution of C-MARS has a better performance. When we consider the stability of the measures, generally C-MARS solutions have a better performance. From these results, again we can not found a meaningful difference between the methods and not decide which method has a better performance. Therefore, we handle all performance measures for considering the relationship between the measures and their efficiency. Tukey multiple comparison tests (=0.50) are used to decide whether the differences among the averages of different measures are statistically significant or not [38]. An ordinal semantic scale of “very poor”, “poor”, “good” and “very good” are used in order to reevaluate and express the performances of the models. If the Tukey test do not indicate a statistically significant difference between these two methods, then the same semantic evaluation is used for both methods based on the measure under consideration. The results are presented in Table 4.4.

Table 4.4: Evaluation of the models of LHS data based on Tukey test and on the ordinal semantic scale.

Measures	MARS	S1	S2	S3
MSE	good	poor	good	good
Cor. Coeff.	very good	good	very good	very good
R^2	very good	good	very good	very good
Adj-R^2	very good	good	very good	very good
PWI-1	very good	very good	very good	very good
PWI-2	very good	very good	very good	very good
Press	good	poor	good	good
R^2-Pred	very poor	poor	very poor	very poor
Mallows Cp	poor	very poor	very poor	very poor
Stability MSE	poor	good	poor	poor
Stability Cor. Coeff.	very good	very good	very good	very good
Stability R^2	very good	very good	very good	very good
Stability Adj-R^2	very good	good	very good	very good
Stability PWI-1	very good	very good	very good	very good
Stability PWI-2	very good	good	very good	very good
Stability Press	very poor	very poor	very poor	very poor
Stability R^2-Pred	good	good	good	good
Stability Mallows Cp	good	good	good	good

In the first data set, the C-MARS models S2 and S3, have the same performance. Except for Mallows’ Cp, there is no significant difference between the MARS, S2 and S3 solutions. Mallows’ Cp criterion focuses on minimizing total mean square error and the regression bias. We may prefer a model that yields a Cp value slightly larger than the minimum but which has slight (or no) bias. With respect to Mallows’ Cp, MARS has a better performance than the other three solutions of C-MARS. The solution S1 shows a lower performance than the other solutions with regard to fit, complexity and stability measures.

As in our first data set, we again determine three representative solutions for C-MARS: S1, showing minimum $\|\boldsymbol{L\theta}\|_2$, S2, minimizing solution for PRSS, S3, having minimum $SQRT(RSS)$. In the second case, with respect to $\|\boldsymbol{L\theta}\|_2$ and $SQRT(RSS)$, C-MARS solutions dominate MARS solutions. As for GCV, C-MARS solutions are dominated by MARS solutions. According to $\|\boldsymbol{L\theta}\|_2$, C-MARS solutions dominate MARS solutions. The related tables and figures of the US data, for the results obtained from all replications and CVs are represented in Appendix D. For the second data set, the results indicate that MARS and C-MARS solutions have advantages according to their own criteria. Therefore, the models are compared according to the method-free measures. These measures are represented in Table 4.5. This table also contains the stability of the measures that based on the average of nine replications for US data set.

When we compare the preformance of solutions, we see that MARS and solution S3 of C-MARS have a better performance with respect to fit and complexity measures. For stability of measures, all of the C-MARS solutions have better performance.

For the second data set, we can not find a remarkable difference between the methods. In order to define a significant difference between them, we apply the Tukey test on the ordinal semantic scale to this data set. The results are given in Table 4.6.

Table 4.5: Averages of performance measure values for the models and stability of measures for US data.

Measures	MARS	S1	S2	S3
MAE	0.060356	0.253289	0.093656	0.067233
MSE	0.007678	0.112378	0.024211	0.009311
RMSE	0.084367	0.3291	0.131044	0.092156
MAPE	19.38231	60.35537	25.95136	19.9871
Cor. Coeff.	0.996611	0.9593	0.990756	0.995822
R^2	0.993256	0.921489	0.981711	0.991689
Adj-R^2	0.947022	0.943133	0.940256	0.946611
PWI-1	0.983111	0.996411	0.986789	0.980256
PWI-2	0.986377778	0.99486	0.987244	0.983811
Press	1.018322	1.1079	1.021311	1.013489
R^2-Pred	25.97532	32.44444	32.44444	32.44444
Mallows Cp	35.5427	49.5556	49.55556	49.55556
Stability MSE	-0.51944	-0.07933	-0.41617	-0.48554
Stability Cor. Coeff.	0.001122222	0.002733333	0.000744444	0.000944444
Stability R^2	0.002222	0.005467	0.0015	0.001822
Stability Adj-R^2	0.004055556	0.058067	0.016711	0.005867
Stability PWI-1	0.007711	0.006133	0.003511	0.003567
Stability PWI-2	0.004366667	-0.00327	0.004211	0.007522
Stability Press	-0.99791	-0.39339	-0.9899	-0.85568
Stability R^2-Pred	-0.00882	-0.02006	-0.01029	-0.00633
Stability Mallows Cp	0.131356	0	0	0

The S2 and S3 models have completely the same performance for the second data set. For the stability of Mallows' Cp, all of the C-MARS solution have a better performance than the MARS solution. Except this measure, there is not a remarkable difference between MARS and S2 and S3 solutions. We may prefer S2 and S3 solutions instead of MARS solution, because of their better stability performances for Mallows' Cp. On the other hand, S1 solution has lower performance compared with the other solutions.

The same comparisons are applied to the last data set which is a real-world data obtained from metal casting industry. When we evaluate the solutions according to the method based performance measures $SQRT(RSS)$, $\|\boldsymbol{L\theta}\|_2$ and GCV, for all

Table 4.6: Evaluation of the models of US data based on Tukey test and on the ordinal semantic scale.

Measures	MARS	S1	S2	S3
MSE	very good	good	very good	very good
Cor. Coeff.	very good	good	very good	very good
R^2	very good	good	very good	very good
Adj-R^2	very good	good	very good	very good
PWI-1	very good	very good	very good	very good
PWI-2	very good	very good	very good	very good
Press	very good	very good	very good	very good
R^2-Pred	very good	very good	very good	very good
Mallows Cp	very poor	very poor	very poor	very poor
Stability MSE	very poor	good	very poor	very poor
Stability Cor. Coeff.	good	good	good	good
Stability R^2	very good	very good	very good	very good
Stability Adj-R^2	very good	good	very good	very good
Stability PWI-1	very good	very good	very good	very good
Stability PWI-2	very good	very good	very good	very good
Stability Press	very poor	poor	very poor	very poor
Stability R^2-Pred	very good	very good	very good	very good
Stability Mallows Cp	good	very good	very good	very good

replications and all CVs, MARS and C-MARS solutions have better performance with respect to their own criteria. In other words, while C-MARS solutions dominate MARS solutions according to $\|\boldsymbol{L\theta}\|_2$ and $SQRT(RSS)$, MARS solutions dominate C-MARS solutions with respect to GCV. The tables and figures containing the replication and CV results can be found in Appendix E.

For a general evaluation, as we apply to the first two data sets, with respect to the performance measures, Table 4.7 includes averages of performance measure values and their stability based on the average of nine replications for the real-world data set.

As it is seen from this table, MARS and C-MARS solutions have poor performances with respect to fit and complexity measures. Because in this data set, there are not any relations between predictors and response variable. This is an expected

Table 4.7: Averages of performance measure values for the models and stability of measures for metal casting data.

Measures	MARS	S1	S2	S3
MAE	0.951778	0.761089	0.982511	1.129667
MSE	1.897422	0.962722	1.810789	2.316311
RMSE	1.273433	0.980578	1.284978	1.4784
MAPE	353.4088	168.5833	316.4185	412.5994
Cor. Coeff.	0.053844	0.154356	0.088789	-0.08333
R^2	0.007311	0.043311	0.044211	0.038467
Adj-R^2	-1.59646	-2.39168	-4.90102	-7.69542
PWI-1	0.841911	0.942056	0.953056	0.9528
PWI-2	0.8853	1 0.992711	0.992711	
Press	86.66932	15.10758	126.565	161.991
R^2-Pred	4.063689	1.526	5.371867	6.616667
Mallows Cp	7.390556	33.33333	33.33333	33.33333
Stability MSE	-0,521944444	-0,12768	-0,48579	-0,73672
Stability Cor. Coeff.	0.671511111	0.621055556	0.870422222	-1.287366667
Stability R^2	0.8662	0.803667	0.845433	0.904378
Stability Adj-R^2	-0.261511111	-1.098022222	-2.092011111	-1.964122222
Stability PWI-1	0.002077778	0.003	0.002844444	0.003977778
Stability PWI-2	7,77778E-05	-0,00451	-0,00082	0,0019
Stability Press	0,936288889	-0,52274	-0,9842	-0,86552
Stability R^2-Pred	-0,415277778	-0,14463	-0,29448	-0,40852
Stability Mallows Cp	0,695133333	0,0825	0,0825	0,0825

situation for this data set. When we look at the stability of measures we notice that MARS and the solution S1 of C-MARS have a better performance. As in the first two cases, there is no meaningful difference between MARS and C-MARS in order to define the best model. Therefore, we apply a Tukey test to the last data set.

When we look at Table 4.8, these two methods can not provide a best model for this real-world data set. As it is stated before, because of the structure of this data set, MARS and C-MARS have the same performance both for measures and their stabilities.

Table 4.8: Evaluation of the models of metal casting data based on Tukey test and on the ordinal semantic scale.

Measures	MARS	S1	S2	S3
MSE	very poor	very poor	very poor	very poor
Cor. Coeff.	very poor	very poor	very poor	very poor
R^2	poor	poor	poor	poor
Adj-R^2	poor	poor	poor	very poor
PWI-1	very good	very good	very good	very good
PWI-2	very good	very good	very good	very good
Press	very poor	very poor	very poor	very poor
R^2-Pred	very poor	very poor	very poor	very poor
Mallows Cp	good	poor	poor	poor
Stability MSE	very poor	poor	very poor	very poor
Stability Cor. Coeff.	very poor	very poor	very poor	very poor
Stability R^2	poor	poor	poor	poor
Stability Adj-R^2	very poor	very poor	very poor	very poor
Stability PWI-1	very good	very good	very good	very good
Stability PWI-2	very good	very good	very good	very good
Stability Press	very poor	poor	very poor	very poor
Stability R^2-Pred	poor	poor	poor	poor
Stability Mallows Cp	very poor	poor	poor	poor

CHAPTER 5

CONCLUSION AND FURTHER STUDIES

This study on regression and classification provides a new contribution to the MARS method which is applied in many areas during the last decades. The MARS algorithm is modified by constructing a penalized residual sum of squares (PRSS) as a Tikhonov regularization problem. This problem is solved by using continuous optimization, especially, conic quadratic programming (CQP). This provides us an alternative modeling technique for MARS. We named our method as C-MARS.

For examining the efficiency of C-MARS, it is compared with MARS method by using three different data sets. This comparison is applied first of all according to these measures: Norm of RSS, norm of $\boldsymbol{L\theta}$ and GCV. The results of these applications show that C-MARS has a better performance with respect to the norm of $\boldsymbol{L\theta}$. On the other hand, according to the GCV, MARS has a better performance.

According to the method-free performance measures, the application results indicate that there is not a significant difference between C-MARS and MARS solutions. However, performance measures of C-MARS show higher stability. Besides these comparisons, by using the Tukey test, it is aimed at to determine whether there are statistically significant differences between the averaged values of employed measures. The results obtained from the Tukey test do not indicate statistically significant differences and, then, according to an ordinal semantic scale ("very poor", "poor", "good", "very good") the results have been re-evaluated.

When we consider the data sets according to the data structure, the first two data sets, LHS and US, are containing outliers and do not promise a defining relationship between predictors and response variable. In these two data sets, the solutions S2 and S3 of C-MARS and MARS solutions have similar performances. However, the solution S1 of C-MARS has a worse performance compared with the other solutions. For the real-world data set which does not include an outlier, MARS and C-MARS generate models having not a remarkable difference between each other. In general, for all data sets, there is not a remarkable difference between MARS and C-MARS solutions according to the methpd-free measures. C-MARS generates more complex models than MARS with respect to the number of BFs. Because C-MARS employs all the BFs obtained from the forward stepwise algorithm of MARS. Even it does not remove the BFs having coefficients close to zero. Moreover, $\sqrt{M}$ values determined by CQP are choosen as model free. C-MARS provides at least one solution very similar to MARS solution. Moreover, this solution is sometimes better than MARS solution. Because of this the solution S2, which is minimizing solution for PRSS, is more preferable in our cases.

C-MARS provides its solutions by using CQP. In this respect, it has the advantage of speed and complexity as defined by Arkadi Nemirovski [47].

For all three data sets used in the applications, C-MARS generates better models according to norm of $\boldsymbol{L\theta}$. Hence, the minimization of norm of $\boldsymbol{L\theta}$ is itself maximizing the stability as explained in (3.2.2). After discretizing the integrals which are measures of energy, we try to keep that energy under control by bounding it (in CQP) or minimizing it in the framework of PRSS [22].

As a future work, MARS and C-MARS can be compared with other modeling techniques such as artificial neural networks and robust regression in the case of continuous data sets and different distributions of variables. Morever, under the normality assumption, MARS and C-MARS can be compared with linear regression.

In our application on C-MARS, data sets and BFs accessions were made manually. Beside of this, C-MARS does not represent the results as a model form. This takes time when compared with MARS software. In this respect, C-MARS can be improved and made a user friendly DM tool. In addition, when we compare MARS and C-MARS with regard to their computational time it is obviously seen that MARS has a very high speed.

C-MARS generates models with a maximum number of basis functions. However, some coefficients can be "very near" to zero (which can and will be statistically well defined). By removing these coefficients, the model size will be decreased and C-

MARS generated models will have a different, actually smaller number of basis functions.

As we observed in our applications, the models generated from C-MARS tend to a better fitting than MARS models. On the other hand, C-MARS constructs more complex models than MARS. In order to provide less complex models having better fitting we can apply continuous multiobjective approaches such as *Epsilon constraint, goal programming.*

Unlike MARS, C-MARS does not select a better model. For overcoming this difficulty, multiobjective approaches can be used.

The importance and benefit of CQP in manufacturing have already been demonstrated in this study. For further study, the CQP problem in the way of *robust optimization* will be generalized. This kind of optimization is introduced by Aharon Ben-Tal, and used by Laurent El Ghaoui in the area of DM. This robustification of CQP with robust optimization can be compared with our previous contributions via CQP which are based on Tikhonov regularization, and with the traditional MARS method.

As Prof. Dr. Gerhard Wilhelm Weber also Efsun Kurum and I understood, piecewise linear functions over a compact interval can be represented by a linear combination of one-dimentional MARS basis functions. For example, for a stochastic process these basis functions can be used for a approximative representation of the trajectories.

In general, for solving Tikhonov regularization problem, SVD is used where $\boldsymbol{L} = \boldsymbol{I}$. However, in *our* case because of the form of $\boldsymbol{L}$, GSVD is employed, so that GSVD allows the solution to this problem to be expressed by a sum of filter factors times generalized singular vectors. When the generalized singular values γ_j tend to zero the corresponding contributious of independent variables vanish. This causes a feature selection. In this frame, as a further study, γ_j and penalty parameter λ for Tikhonov regularization problem can be compared and truncation conducted.

C-MARS includes an improvement on the second part of MARS algorithm. A similar improvement is also possible for the first part of the algorithm by using clustering techniques. The first part of MARS algorithm, e.g., forward stepwise algorithm, determines knot points among the data points for obtaining basis functions. Increasing in the number of data points results in a one-to-one manner in an increase in the number of knot points. Therefore it gives rise to complexity. For this reason we decide to determine suitable knot points for the data set by using *clustering theory*. There are two ways for doing this: *we can first cluster, then project*, or, *we can first project, then cluster*. This approach is the challenging core idea of a new research project. By this thesis we intented to give a contribution to the theory, methods and applications of mathematical data mining, for displaying its beauty and inviting to future research challenges.

REFERENCES

[1] Abraham, A., Steinberg, D., and Philip, N.S., *Rainfall forecasting using soft computing models and multivariate adaptive regression splines*, IEEE, 2001.

[2] Aster, A., Borchers, B., and Thurber, C., *Parameter Estimation and Inverse Problems,* Academic Press, 2004.

[3] Austin, P.C., *A comparison of regression trees, logistic regression, generalized additive models and multivariate adaptive regression splines for predicting, AMI mortality*, Statistics in Medicine, 26, 15 (2007) 2937-2957.

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

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

[6] Bates, D.M., Watts, D.G., *Nonlinear Regression Analysis and Its Applications,* New York: Wiley, 1988.

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

[8] Chou, S.M., Lee, T.S., Shao, Y.E., and Chen, I.F., *Mining the breat cancer pattern using artificial neural networks and multivariate adaptive regression splines*, 2003.

[9] Copyright StatSoft, Inc., *Multivariate Adaptive Regression Splines*, http://www.statsoft.com/textbook/stmars.html (accessed 05 Sep. 2008).

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

[11] Crino, S., and Brown, D.E., *Global optimization with multivariate adaptive regression splines*, IEEE Transactions on Systems Man and Cybernetics Part b - cybernetics, 37, 2 (2007) 333-340.

[12] De Veaux, R.D., Psichogios, D.C., and Ungar, L.H., *A Comparison of Two Nonparametric Schemes: MARS and Neural Networks*, Computers in Chemical Engineering, 17, (1993) 819-837.

[13] Deconinck, E., Coomons, D., and Heyden, Y.V., *Explorations of linear modeling techniques and their combinations with multivariate adaptive regression splines to predict gastro-intestinal absorption of drugs*, Journal of Pharmaceutical and Biomedical Analysis, 43, 1 (2007) 119-130.

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

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

[16] El Ghaoui, L., *Robust Optimization and Applications*, IMA Tutorial, 2003.

[17] Elith, J., and Leathwick, J., *Predicting species distribution from museum and herborium records using multiresponse models fitted with multivariate adaptive regression splines*, Diversity and Distributions, 13, 3 (2007) 265-275.

[18] Fox, J., *Nonparametric Regression*, in: B. Everitt and D. Howell, eds. Encyclopedia of Statistics in the Behavioral Sciences. London: Wiley, 2005.

[19] Fox, J., *Nonparametric Regression*, An R and S-PLUS Companion to Applied Regression, Sage, 2002.

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

[21] Friedman, J.H., *Discussion*, Technometrics, 33 (1991) 145-148.

[22] Friedman, J.H., personel communication with Yerlikaya, F., and Weber, G.W., 2007.

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

[24] Hansen, P.C., *Rank-Deficient and Discrete III-Posed Problems: Numerical Aspects of Linear Inversion*, SIAM, Philadelphia, 1998.

[25] Hastie, T., Tibshirani, R., and Friedman, J., *The Elements of Statistical Learning, Data Mining, Inference and Prediction,* Springer, 2001.

[26] Hildeman, R.J., and Hamilton, H.J., *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.

[27] Hildeman, R.J., and Hamilton, H.J., *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.

[28] Isaacson E., and Keller, HB., *Analysis of Numerical Methods*, John Willy and Sons, New York, 1966.

[29] Karmakar, N., *A new polynomial-time algorithm for linear programming*, Combinatorica, 4 (1984) 373-395.

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

[31] Ko, Myung, and Bryson, K.M.O., *Reexamining the impact of information technology investment an productivity using regression tree and multivariate*

adaptive regression splines (MARS), Information Technology and Management, Springer Netherlands, 2008.

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

[33] Kubin, G., *Nonlinear prediction of mobile radio channels: measurments and mars model designs*, IEEE Proc. International Conference on Acoustics, Speech, and Signal Processing, 5, 15-19 (1999) 2667-2670.

[34] Larose, D.T., *Data Mining Methods and Models*, Hoboken, NJ: Wiley-Interscience, 2006.

[35] LeBlanc, M., Crowley, J., *Adaptive regression splines in the Cox model*, Biometrics, 55, 1 (2004) 204-213.

[36] Lee, T.S., Chiu, C.C., Chou, Y.C., and Lu, C.J., *Mining the customer credit using classification and regression tree and multivariate adaptive regression splines*, Computational Statistics and Data Analysis 50, 4 (2006) 1113-1130.

[37] Liang, K.Y., *Generalized Linear Models, Estimating Functions and Multivariate Extentions,* Lecture Series in Statistics, Institute of Statistical Science, Academia Sinica, Taiwan, R.O.C., 2000.

[38] Mendenhall, W., and Sincich, T., *Regression Analysis: A Second Course in Statistics*, Pearson Prentice Hall: Upper Saddle River, NJ., 2003.

[39] MARS from Salford Systems, http://www.salfordsystems.com/mars/phb (accessed 05 Sep. 2008).

[40] *MARS User Guide*, San Diego, CA: Salford Systems, 2001.

[41] MOSEK, *A very powerful commercial software for CQP*, http://www.mosek.com (accessed 05 Sep. 2008).

[42] Mukkamala, S., and Sung, A.H., *A comparative study of techniques for intrusion detection*, in IEEE Proc. 15th International Conference on Tools with Artificial Intelligence, (2003) 570-577.

[43] Moisen, G. G., and Frescino, T. S., *Comparing five modeling techniques for predicting forest characteristics*, Ecological Modelling, 2002.

[44] Myers, R.H., and Montgomery, D.C., *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, New York: Wiley, 2002.

[45] Nash, G., Sofer, A., *Linear and Nonlinear Programming*, McGraw-Hill, New York, 1996.

[46] Nelder, J.A., and Wedderburn, R.W.M., *Generalized linear models*, Journal of the Royal Statistical Society A, 145 (1972) 470-484.

[47] Nemirovski, A., *A lectures on modern convex optimization*, Israel Institute of Technology (2002), http://iew3.technion.ac.il/Labs/Opt/opt/LN/Final.pdf (accessed 05 Sep. 2008).

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

[49] Sheid, F., *Numerical Analysis*, McGraw-Hill Book Company, New-York, 1968.

[50] Steuer, R.E., *Multiple Criteria Optimization: Theory, Computation and Application*, John Wiley and Sons, New York, NY, 1985.

[51] Taylan, P., and Weber, G.W., *New approaches to regression in financial mathematics by additive models*, Journal of Computational Technologies, 12, 2 (2007) 3-22.

[52] Taylan, P., Weber, G.W., and Beck, A., *New approaches to regression by generalized additive, models and continues optimization for modern applications in finance, science and technology*, Journal Optimization, 56, 5-6 (2007) 675-698.

[53] Taylan, P., Weber, G.W., and Yerlikaya, F., *Continuous optimization applied in MARS for modern applications in finance, science and technology*, in the ISI Proceedings of 20th Mini-EURO Conference Continuous Optimization and Knowledge-Based Technologies, Neringa, Lithuania (2008) 317-322.

[54] Tsai, J.C.C., and Chen, V.C.P., *Flexible and robust implamentations of multivariate adaptive regression splines within a wastewater treatment stochastic dynamic program*, Quality and Reliability Engineering International, 21, 7 (2005) 689-699.

[55] Weber, G.W., Taylan, P., Özögür, S., Öztürk and Akteke B., *Statistical Learning and Optimization Methods in Data Mining,* in: Recent Advances in Statistics, eds.: H.O. Ayhan and I. Batmaz, Turkish Statistical Institute Press, Ankara, at the occasion of "Graduate Summer School on New Advances in Statistics" (August 2007) 181-195.

[56] Weber, G.W., Taylan, P., Sezer, D., Köksal, G., Batmaz, I., Yerlikaya, F., Özöğür, S., Shawe-Taylor, J., Özbudak, F., and Akyıldız, E., *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.

[57] Wood, S.N., *Generalized Additive Models, An Introduction with R*, Chapman and Hall, New-York, 2006.

[58] Xiong, R., and Meullenet, J.F., *Application of multivariate Adaptive Regression Splines (MARS) to the preference mapping of cheese sticks*, Journal of Food Science, 69 (2004) 131-139.

[59] Xu, Q.S., Daeyaert, F., Lewi, P.J., and Massart, D.L., *Studies of relationship between biological activities and HIV reverse transcriptase inhibitors by multivariate adaptive regression splines with Curds and Whey*, 2005.

[60] Xu, Q.S., Massart, D.L., Liang, Y.Z., and Fang, K.T., *Two-step multivariate adaptive regression splines for modeling a quantitative relationship between gas*

chromatograph retention indices and molecular descriptors, Journal Of Chromatograph 998, 1-2 (2003) 155-167.

[61] Yang, C.C., Prasher, S.O., Lacroix, R., and Kum, S.H., *Application of multivariate adaptive regression splines (MARS) to simulate soil temparature*, Transactions of the ASAE, 47, 3 (2004) 881-887.

[62] York, T.P., Eaves, L.J., and Van den Oord, E.J.C.G., *Multivariate adaptive regression splines: a powerful method for detecting disease-risk relationship differences among subgroubs*, Statistics in Medicine, 25, 8 (2006) 1355-1367.

[63] York, T.P., Eaves, and Lindan J., *A multivariate adaptive regression splines model for simulations of pesticide transport in soils*, 2003.

[64] Zareipour, H., Bhattacharya, K., and Canizares, C.A., *Forecasting the hourly Ontario energy price by multivariate adaptive regression splines*, IEEE, Power Engineering Society General Meeting, 2006.

[65] Zhou, Y., and Leung, H., *Predicting object-oriented software maintainability using multivariate adaptive regression splines*, Journal of Systems and Software, 80, 8 (2007) 1349-1361.

APPENDIX A

RSS in a Numerical Example

The following function RSS became addressed in Section 3.3. On $I\ (RSS)$:

$$\begin{aligned}
\sum_{i=1}^{N}\big(y_i - \boldsymbol{\theta}^T \boldsymbol{\psi}(\bar{\boldsymbol{d}}_i)\big)^2 = (-1.1242 - \theta_0 - (\max\{0, -0.6109 + 0.159\})\,\theta_1 - \\
(\max\{0, -0.159 + 0.6109\})\,\theta_2 - \\
(\max\{0, -0.5172 + 1.517\})\,\theta_3 - \\
(\max\{0, -0.0781 + 2.576\} * \max\{0, -1.5172 + 1.517\})\,\theta_4 - \\
(\max\{0, -0.8184 + 1.562\} * \max\{0, -1.5172 + 1.517\})\,\theta_5)^2 + \\
(-0.8703 - \theta_0 - (\max\{0, -0.5885 + 0.159\})\,\theta_1 - \\
(\max\{0, -0.159 + 0.5885\})\,\theta_2 - \\
(\max\{0, -1.3501 + 1.517\})\,\theta_3 - \\
(\max\{0, -0.0781 + 2.576\} * \max\{0, -1.3501 + 1.517\})\,\theta_4 - \\
(\max\{0, -0.8184 + 1.562\} * \max\{0, -1.3501 + 1.517\})\,\theta_5)^2 + \\
(-0.9549 - \theta_0 - (\max\{0, -0.5885 + 0.159\})\,\theta_1 - \\
(\max\{0, -0.159 + 0.5885\})\,\theta_2 - \\
(\max\{0, -1.3501 + 1.517\})\,\theta_3 - \\
(\max\{0, -0.0781 + 2.576\} * \max\{0, -1.3501 + 1.517\})\,\theta_4 - \\
(\max\{0, 0.8184 + 1.562\} * \max\{0, -1.3501 + 1.517\})\,\theta_5)^2 + \\
(-0.8703 - \theta_0 - (\max\{0, -0.5885 + 0.159\})\,\theta_1 - \\
(\max\{0, -0.159 + 0.5885\})\,\theta_2 - \\
(\max\{0, -1.3501 + 1.517\})\,\theta_3 - \\
(\max\{0, -0.0781 + 2.576\} * \max\{0, -1.3501 + 1.517\})\,\theta_4 -
\end{aligned}$$

$$\left(\max\left\{0, 1.4136 + 1.562\right\} * \max\left\{0, -1.3501 + 1.517\right\}\right)\theta_5)^2 +$$

$$(-0.9549 - \theta_0 - \left(\max\left\{0, -0.5885 + 0.159\right\}\right)\theta_1 -$$

$$\left(\max\left\{0, -0.159 + 0.5885\right\}\right)\theta_2 -$$

$$\left(\max\left\{0, -1.3501 + 1.517\right\}\right)\theta_3 -$$

$$\left(\max\left\{0, -2.5759 + 2.576\right\} * \max\left\{0, -1.3501 + 1.517\right\}\right)\theta_4 -$$

$$\left(\max\left\{0, 2.0089 + 1.562\right\} * \max\left\{0, -1.3501 + 1.517\right\}\right)\theta_5)^2 +$$

$$(-0.8703 - \theta_0 - \left(\max\left\{0, -0.5885 + 0.159\right\}\right)\theta_1 -$$

$$\left(\max\left\{0, -0.159 + 0.5885\right\}\right)\theta_2 -$$

$$\left(\max\left\{0, -1.3501 + 1.517\right\}\right)\theta_3 -$$

$$\left(\max\left\{0, -2.5759 + 2.576\right\} * \max\left\{0, -1.3501 + 1.517\right\}\right)\theta_4 -$$

$$\left(\max\left\{0, 2.0089 + 1.562\right\} * \max\left\{0, -1.3501 + 1.517\right\}\right)\theta_5)^2 +$$

$$(-1.0396 - \theta_0 - \left(max\left\{0, -0.4937 + 0.159\right\}\right)\theta_1 -$$

$$\left(\max\left\{0, -0.159 + 0.4937\right\}\right)\theta_2 -$$

$$\left(\max\left\{0, -0.9949 + 1.517\right\}\right)\theta_3 -$$

$$\left(\max\left\{0, -0.0781 + 2.576\right\} * \max\left\{0, -0.9949 + 1.517\right\}\right)\theta_4 -$$

$$\left(\max\left\{0, -0.372 + 1.562\right\} * \max\left\{0, -0.9949 + 1.517\right\}\right)\theta_5)^2 +$$

$$(-0.447 - \theta_0 - \left(\max\left\{0, -0.4463 + 0.159\right\}\right)\theta_1 -$$

$$\left(\max\left\{0, -0.159 + 0.4463\right\}\right)\theta_2 -$$

$$\left(\max\left\{0, -0.8278 + 1.517\right\}\right)\theta_3 -$$

$$\left(\max\left\{0, -0.0781 + 2.576\right\} * \max\left\{0, -0.8278 + 1.517\right\}\right)\theta_4 -$$

$$\left(\max\left\{0, -0.372 + 1.562\right\} * \max\left\{0, -0.8278 + 1.517\right\}\right)\theta_5)^2 +$$

$$(-0.701 - \theta_0 - \left(\max\left\{0, -0.4463 + 0.159\right\}\right)\theta_1 -$$

$$\left(\max\left\{0, -0.159 + 0.4463\right\}\right)\theta_2 -$$

$$\left(\max\left\{0, -0.8278 + 1.517\right\}\right)\theta_3 -$$

$$\left(\max\left\{0, -0.0781 + 2.576\right\} * \max\left\{0, -0.8278 + 1.517\right\}\right)\theta_4 -$$

$$\left(\max\left\{0, -0.372 + 1.562\right\} * \max\left\{0, -0.8278 + 1.517\right\}\right)\theta_5)^2 +$$

$$(-0.6163-\theta_0-(\max\{0,-0.4463+0.159\})\,\theta_1-$$

$$(\max\{0,-0.159+0.4463\})\,\theta_2-$$

$$(\max\{0,-0.8278+1.517\})\,\theta_3-$$

$$(\max\{0,-0.0781+2.576\}*\max\{0,-0.8278+1.517\})\,\theta_4-$$

$$(\max\{0,-0.372+1.562\}*\max\{0,-0.8278+1.517\})\,\theta_5)^2+$$

$$(-0.447-\theta_0-(\max\{0,-0.4463+0.159\})\,\theta_1-$$

$$(\max\{0,-0.159+0.4463\})\,\theta_2-$$

$$(\max\{0,-0.8278+1.517\})\,\theta_3-$$

$$(\max\{0,-0.0781+2.576\}*\max\{0,-0.8278+1.517\})\,\theta_4-$$

$$(\max\{0,0.8184+1.562\}*\max\{0,-0.8278+1.517\})\,\theta_5)^2+$$

$$(-0.6163-\theta_0-(\max\{0,-0.4463+0.159\})\,\theta_1-$$

$$(\max\{0,-0.159+0.4463\})\,\theta_2-$$

$$(\max\{0,-0.8278+1.517\})\,\theta_3-$$

$$(\max\{0,-0.0781+2.576\}*\max\{0,-0.8278+1.517\})\,\theta_4-$$

$$(\max\{0,0.8184+1.562\}*\max\{0,-0.8278+1.517\})\,\theta_5)^2+$$

$$(-0.447-\theta_0-(\max\{0,-0.4463+0.159\})\,\theta_1-$$

$$(\max\{0,-0.159+0.4463\})\,\theta_2-$$

$$(\max\{0,-0.8278+1.517\})\,\theta_3-$$

$$(\max\{0,-0.0781+2.576\}*\max\{0,-0.8278+1.517\})\,\theta_4-$$

$$(\max\{0,0.8184+1.562\}*\max\{0,-0.8278+1.517\})\,\theta_5)^2+$$

$$(-0.1085-\theta_0-(\max\{0,-0.4379+0.159\})\,\theta_1-$$

$$(\max\{0,-0.159+0.4379\})\,\theta_2-$$

$$(\max\{0,0.2585+1.517\})\,\theta_3-$$

$$(\max\{0,-0.0781+2.576\}*\max\{0,0.2585+1.517\})\,\theta_4-$$

$$(\max\{0,-0.372+1.562\}*\max\{0,0.2585+1.517\})\,\theta_5)^2+$$

$$(-0.1085-\theta_0-(\max\{0,-0.2147+0.159\})\,\theta_1-$$

$$
\begin{aligned}
(\max\{0,-0.159+0.2147\})\,\theta_2 - \\
(\max\{0,0.0496+1.517\})\,\theta_3 - \\
(\max\{0,-0.0781+2.576\}*\max\{0,0.0496+1.517\})\,\theta_4 - \\
(\max\{0,-0.372+1.562\}*\max\{0,0.0496+1.517\})\,\theta_5)^2 + \\
(-0.0238-\theta_0-(\max\{0,-0.2147+0.159\})\,\theta_1 - \\
(\max\{0,-0.159+0.2147\})\,\theta_2 - \\
(\max\{0,0.2585+1.517\})\,\theta_3 - \\
(\max\{0,-0.0781+2.576\}*\max\{0,0.2585+1.517\})\,\theta_4 - \\
(\max\{0,-0.372+1.562\}*\max\{0,0.2585+1.517\})\,\theta_5)^2 + \\
(-0.1931-\theta_0-(\max\{0,-0.1589+0.159\})\,\theta_1 - \\
(\max\{0,-0.159+0.1589\})\,\theta_2 - \\
(\max\{0,0.2585+1.517\})\,\theta_3 - \\
(\max\{0,-0.0781+2.576\}*\max\{0,0.2585+1.517\})\,\theta_4 - \\
(\max\{0,-0.372+1.562\}*\max\{0,0.2585+1.517\})\,\theta_5)^2 + \\
(-0.1085-\theta_0-(\max\{0,-0.1589+0.159\})\,\theta_1 - \\
(\max\{0,-0.159+-0.1589\})\,\theta_2 - \\
(\max\{0,0.2585+1.517\})\,\theta_3 - \\
(\max\{0,-0.0781+2.576\}*\max\{0,0.2585+1.517\})\,\theta_4 - \\
(\max\{0,-0.372+1.562\}*\max\{0,0.2585+1.517\})\,\theta_5)^2 + \\
(-0.1931-\theta_0-(\max\{0,-0.1589+0.159\})\,\theta_1 - \\
(\max\{0,-0.159+0.1589\})\,\theta_2 - \\
(\max\{0,0.2585+1.517\})\,\theta_3 - \\
(\max\{0,-0.0781+2.576\}*\max\{0,0.2585+1.517\})\,\theta_4 - \\
(\max\{0,-0.372+1.562\}*\max\{0,0.2585+1.517\})\,\theta_5)^2 + \\
(-0.1085-\theta_0-(\max\{0,-0.1589+0.159\})\,\theta_1 - \\
(\max\{0,-0.159+0.1589\})\,\theta_2 -
\end{aligned}
$$

$$
\begin{aligned}
\left(\max\left\{0, 0.2585 + 1.517\right\}\right)\theta_3 - &\\
\left(\max\left\{0, -0.0781 + 2.576\right\} * \max\left\{0, 0.2585 + 1.517\right\}\right)\theta_4 - &\\
\left(\max\left\{0, 0.8184 + 1.562\right\} * \max\left\{0, 0.2585 + 1.517\right\}\right)\theta_5)^2 + &\\
(0.0608 - \theta_0 - \left(\max\left\{0, -0.1589 + 0.159\right\}\right)\theta_1 - &\\
\left(\max\left\{0, -0.159 + 0.1589\right\}\right)\theta_2 - &\\
\left(\max\left\{0, 0.2585 + 1.517\right\}\right)\theta_3 - &\\
\left(\max\left\{0, -0.0781 + 2.576\right\} * \max\left\{0, 0.2585 + 1.517\right\}\right)\theta_4 - &\\
\left(\max\left\{0, 0.8184 + 1.562\right\} * \max\left\{0, 0.2585 + 1.517\right\}\right)\theta_5)^2 + &\\
(-0.0238 - \theta_0 - \left(\max\left\{0, -0.1589 + 0.159\right\}\right)\theta_1 - &\\
\left(\max\left\{0, -0.159 + 0.1589\right\}\right)\theta_2 - &\\
\left(\max\left\{0, 1.0942 + 1.517\right\}\right)\theta_3 - &\\
\left(\max\left\{0, -0.0781 + 2.576\right\} * \max\left\{0, 1.0942 + 1.517\right\}\right)\theta_4 - &\\
\left(\max\left\{0, 0.8184 + 1.562\right\} * \max\left\{0, 1.0942 + 1.517\right\}\right)\theta_5)^2 + &\\
(0.2301 - \theta_0 - \left(\max\left\{0, 0.0643 + 0.159\right\}\right)\theta_1 - &\\
\left(\max\left\{0, -0.159 - 0.0643\right\}\right)\theta_2 - &\\
\left(\max\left\{0, 1.0942 + 1.517\right\}\right)\theta_3 - &\\
\left(\max\left\{0, -0.0781 + 2.576\right\} * \max\left\{0, 1.0942 + 1.517\right\}\right)\theta_4 - &\\
\left(\max\left\{0, -0.372 + 1.562\right\} * \max\left\{0, 1.0942 + 1.517\right\}\right)\theta_5)^2 + &\\
(0.3148 - \theta_0 - \left(\max\left\{0, 0.0643 + 0.159\right\}\right)\theta_1 - &\\
\left(\max\left\{0, -0.159 - 0.0643\right\}\right)\theta_2 - &\\
\left(\max\left\{0, 1.0942 + 1.517\right\}\right)\theta_3 - &\\
\left(\max\left\{0, -0.0781 + 2.576\right\} * \max\left\{0, 1.0942 + 1.517\right\}\right)\theta_4 - &\\
\left(\max\left\{0, -1.5624 + 1.562\right\} * \max\left\{0, 1.0942 + 1.517\right\}\right)\theta_5)^2 + &\\
(0.1455 - \theta_0 - \left(\max\left\{0, 0.0643 + 0.159\right\}\right)\theta_1 - &\\
\left(\max\left\{0, -0.159 - 0.0643\right\}\right)\theta_2 - &\\
\left(\max\left\{0, 1.0942 + 1.517\right\}\right)\theta_3 - &
\end{aligned}
$$

$$(\max\{0, -0.0781 + 2.576\} * \max\{0, 1.0942 + 1.517\})\,\theta_4 -$$

$$(\max\{0, -0.9672 + 1.562\} * \max\{0, 1.0942 + 1.517\})\,\theta_5)^2 +$$

$$(0.4841 - \theta_0 - (\max\{0, 0.0643 + 0.159\})\,\theta_1 -$$

$$(\max\{0, -0.159 - 0.0643\})\,\theta_2 -$$

$$(\max\{0, 1.4284 + 1.517\})\,\theta_3 -$$

$$(\max\{0, -0.0781 + 2.576\} * \max\{0, 1.4284 + 1.517\})\,\theta_4 -$$

$$(\max\{0, 0.2232 + 1.562\} * \max\{0, 1.4284 + 1.517\})\,\theta_5)^2 +$$

$$(0.5687 - \theta_0 - (\max\{0, 0.1563 + 0.159\})\,\theta_1 -$$

$$(\max\{0, -0.159 - 0.1563\})\,\theta_2 -$$

$$(\max\{0, 1.0942 + 1.517\})\,\theta_3 -$$

$$(\max\{0, -0.0781 + 2.576\} * \max\{0, 1.0942 + 1.517\})\,\theta_4 -$$

$$(\max\{0, -0.372 + 1.562\} * \max\{0, 1.0942 + 1.517\})\,\theta_5)^2 +$$

$$(1.0766 - \theta_0 - (\max\{0, 0.3432 + 0.159\})\,\theta_1 -$$

$$(\max\{0, -0.159 - 0.3432\})\,\theta_2 -$$

$$(\max\{0, 1.0942 + 1.517\})\,\theta_3 -$$

$$(\max\{0, 2.4197 + 2.576\} * \max\{0, 1.0942 + 1.517\})\,\theta_4 -$$

$$(\max\{0, -1.5624 + 1.562\} * \max\{0, 1.0942 + 1.517\})\,\theta_5)^2 +$$

$$(1.1613 - \theta_0 - (\max\{0, 0.3432 + 0.159\})\,\theta_1 -$$

$$(\max\{0, -0.159 - 0.3432\})\,\theta_2 -$$

$$(\max\{0, 1.0942 + 1.517\})\,\theta_3 -$$

$$(\max\{0, -0.0781 + 2.576\} * \max\{0, 1.0942 + 1.517\})\,\theta_4 -$$

$$(\max\{0, -1.5624 + 1.562\} * \max\{0, 1.0942 + 1.517\})\,\theta_5)^2 +$$

$$(0.738 - \theta_0 - (\max\{0, 0.3432 + 0.159\})\,\theta_1 -$$

$$(\max\{0, -0.159 - 0.3432\})\,\theta_2 -$$

$$(\max\{0, 1.0942 + 1.517\})\,\theta_3 -$$

$$(\max\{0, -0.0781 + 2.576\} * \max\{0, 1.0942 + 1.517\})\,\theta_4 -$$

$$
\begin{aligned}
(\max\{0, 0.2232 + 1.562\} * \max\{0, 1.0942 + 1.517\})\,\theta_5)^2 +\\
(2.5156 - \theta_0 - (\max\{0, 2.7145 + 0.159\})\,\theta_1 -\\
(\max\{0, -0.159 - 2.7145\})\,\theta_2 -\\
(\max\{0, 1.0942 + 1.517\})\,\theta_3 -\\
(\max\{0, 2.4197 + 2.576\} * \max\{0, 1.0942 + 1.517\})\,\theta_4 -\\
(\max\{0, -1.5624 + 1.562\} * \max\{0, 1.0942 + 1.517\})\,\theta_5)^2 +\\
(3.5314 - \theta_0 - (\max\{0, 4.3884 + 0.159\})\,\theta_1 -\\
(\max\{0, -0.159 - 4.3884\})\,\theta_2 -\\
(\max\{0, 1.0942 + 1.517\})\,\theta_3 -\\
(\max\{0, 2.4197 + 2.576\} * \max\{0, 1.0942 + 1.517\})\,\theta_4 -\\
(\max\{0, -1.5624 + 1.562\} * \max\{0, 1.0942 + 1.517\})\,\theta_5)^2.
\end{aligned}
$$

After computing the maximum functions, the RSS term has the following form:

$$
\begin{aligned}
\sum_{i=1}^{N} \left(y_i - \boldsymbol{\theta}^T \boldsymbol{\psi}(\bar{\boldsymbol{d}}_i)\right)^2 = (-1.1224 - \theta_0 - 0.4519\theta_2)^2 +\\
(-0.8703 - \theta_0 - 0.4295\theta_2 - 0.1669\theta_3 - 0.4169\theta_4 - 0.3973\theta_5)^2 +\\
(-0.9549 - \theta_0 - 0.4295\theta_2 - 0.1669\theta_3 - 0.4169\theta_4 - 0.3973\theta_5)^2 +\\
(-0.8703 - \theta_0 - 0.4295\theta_2 - 0.1669\theta_3 - 0.4169\theta_4 - 0.4966\theta_5)^2 +\\
(-0.9549 - \theta_0 - 0.4295\theta_2 - 0.1669\theta_3 - 0.5960\theta_5)^2 +\\
(-0.8703 - \theta_0 - 0.4295\theta_2 - 0.1669\theta_3 - 0.5960\theta_5)^2 +\\
(-1.0396 - \theta_0 - 0.3347\theta_2 - 0.5221\theta_3 - 1.3042\theta_4 - 0.6213\theta_5)^2 +\\
(-0.447 - \theta_0 - 0.2873\theta_2 - 0.6892\theta_3 - 1.7216\theta_4 - 0.8201\theta_5)^2 +\\
(-0.701 - \theta_0 - 0.2873\theta_2 - 0.6892\theta_3 - 1.7216\theta_4 - 0.8201\theta_5)^2 +\\
(-0.6163 - \theta_0 - 0.2873\theta_2 - 0.6892\theta_3 - 1.7216\theta_4 - 0.3973\theta_5)^2 +\\
(-0.447 - \theta_0 - 0.2873\theta_2 - 0.6892\theta_3 - 1.7216\theta_4 - 1.6406\theta_5)^2 +
\end{aligned}
$$

$$
\begin{aligned}
(-0.6163-\theta_0-0.2873\theta_2-0.6892\theta_3-1.7216\theta_4-1.6406\theta_5)^2 &+\\
(-0.447-\theta_0-0.2873\theta_2-0.6892\theta_3-1.7216\theta_4-1.6406\theta_5)^2 &+\\
(-0.1085-\theta_0-0.2789\theta_2-1.7755\theta_3-4.4350\theta_4-2.1128\theta_5)^2 &+\\
(-0.1085-\theta_0-0.0557\theta_2-1.5666\theta_3-3.9132\theta_4-1.8643\theta_5)^2 &+\\
(-0.0238-\theta_0-0.0557\theta_2-1.7755\theta_3-4.4350\theta_4-2.1128\theta_5)^2 &+\\
(-0.1931-\theta_0-0.0001\theta_1-1.7755\theta_3-4.4350\theta_4-2.1128\theta_5)^2 &+\\
(-0.1085-\theta_0-0.0001\theta_1-1.7755\theta_3-4.4350\theta_4-2.1128\theta_5)^2 &+\\
(-0.1931-\theta_0-0.0001\theta_1-1.7755\theta_3-4.4350\theta_4-2.1128\theta_5)^2 &+\\
(-0.1085-\theta_0-0.0001\theta_1-1.7755\theta_3-4.4350\theta_4-4.2264\theta_5)^2 &+\\
(0.0680-\theta_0-0.0001\theta_1-1.7755\theta_3-4.4350\theta_4-4.2264\theta_5)^2 &+\\
(-0.0238-\theta_0-0.0001\theta_1-2.6112\theta_3-6.5225\theta_4-6.2157\theta_5)^2 &+\\
(0.2301-\theta_0-0.2233\theta_1-2.6112\theta_3-6.5225\theta_4-3.1073\theta_5)^2 &+\\
(0.3148-\theta_0-0.2233\theta_1-2.6112\theta_3-6.5225\theta_4)^2 &+\\
(0.1455-\theta_0-0.2233\theta_1-2.6112\theta_3-6.5225\theta_4-1.5531\theta_5)^2 &+\\
(0.4841-\theta_0-0.2233\theta_1-2.9454\theta_3-7.3573\theta_4-5.2581\theta_5)^2 &+\\
(0.5687-\theta_0-0.3153\theta_1-2.6112\theta_3-6.5225\theta_4-3.1073\theta_5)^2 &+\\
(1.0766-\theta_0-0.5022\theta_1-2.6112\theta_3-13.0448\theta_4)^2 &+\\
(1.1613-\theta_0-0.5022\theta_1-2.6112\theta_3-6.5225\theta_4)^2 &+\\
(0.738-\theta_0-0.5022\theta_1-2.6112\theta_3-6.5225\theta_4-4.6615\theta_5)^2 &+\\
(2.5156-\theta_0-2.8735\theta_1-2.6112\theta_3-13.0448\theta_4)^2 &+\\
(3.5314-\theta_0-4.5474\theta_1-2.6112\theta_3-13.0448\theta_4)^2 &.
\end{aligned}
$$

APPENDIX B

Matrix Plot of Data Sets

For LHS data

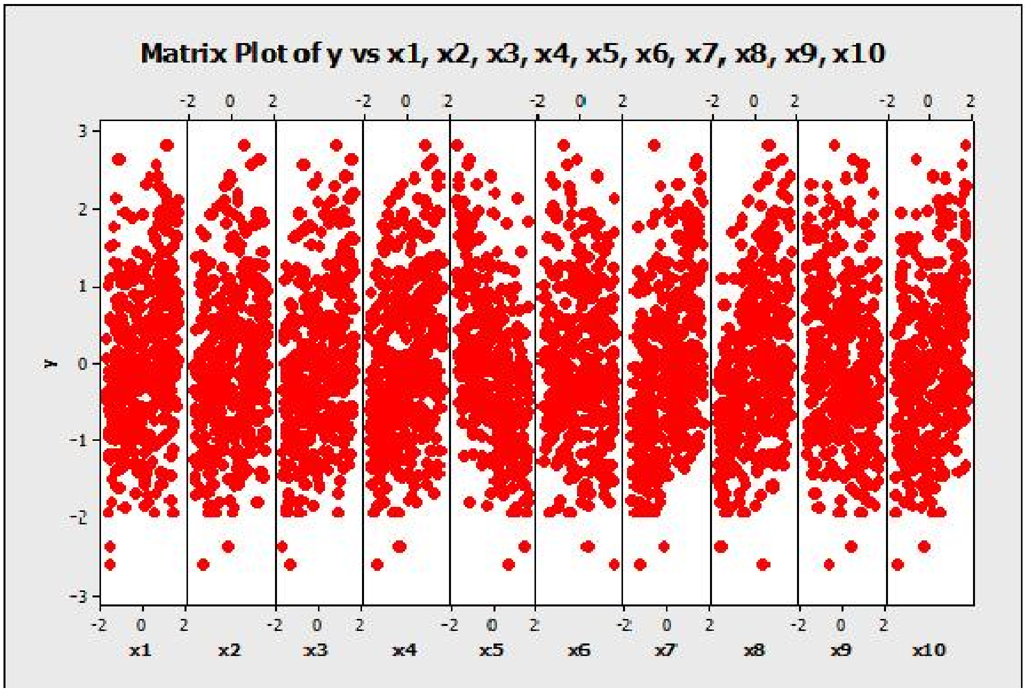

Figure 5.1: Matrix plot of response variable vs. predictor variables for LHS data.

For US data

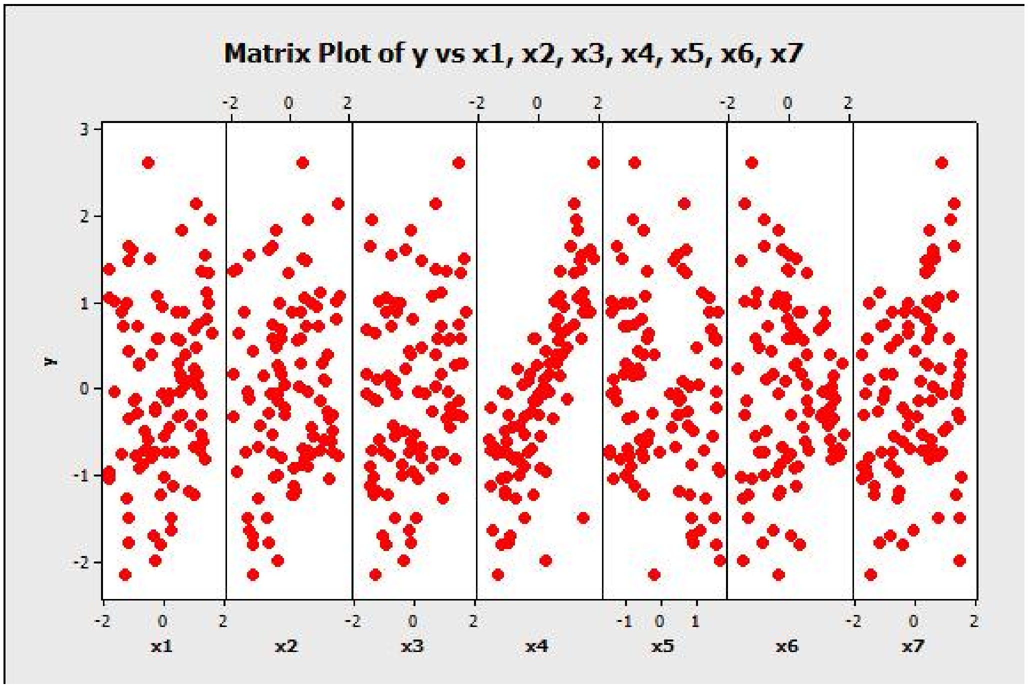

Figure 5.2: Matrix plot of response variable vs. predictor variables for US data.

For Metal Casting data

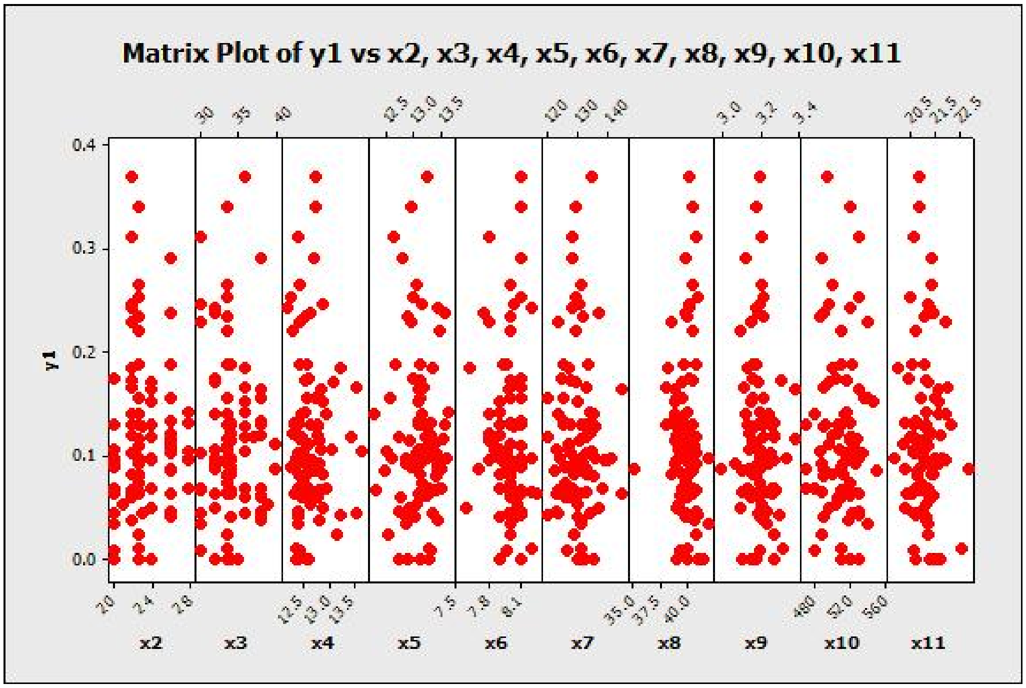

Figure 5.3: Matrix plot of response variable vs. predictor variables ($x_2 - x_{11}$) for metal casting data.

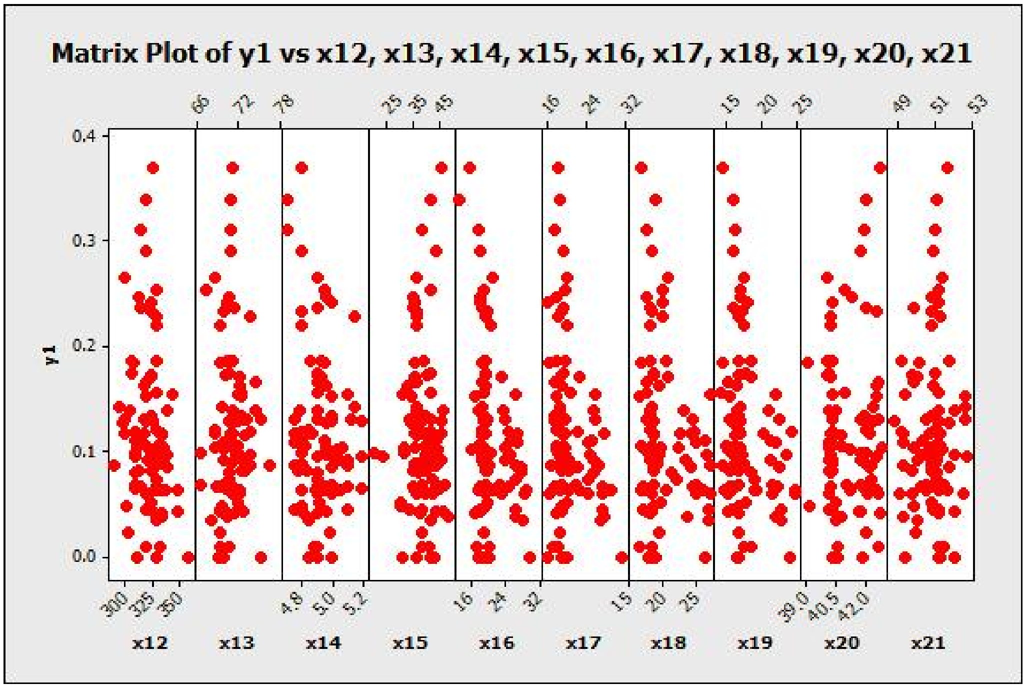

Figure 5.4: Matrix plot of response variable vs. predictor variables ($x_{12} - x_{21}$) for casting data.

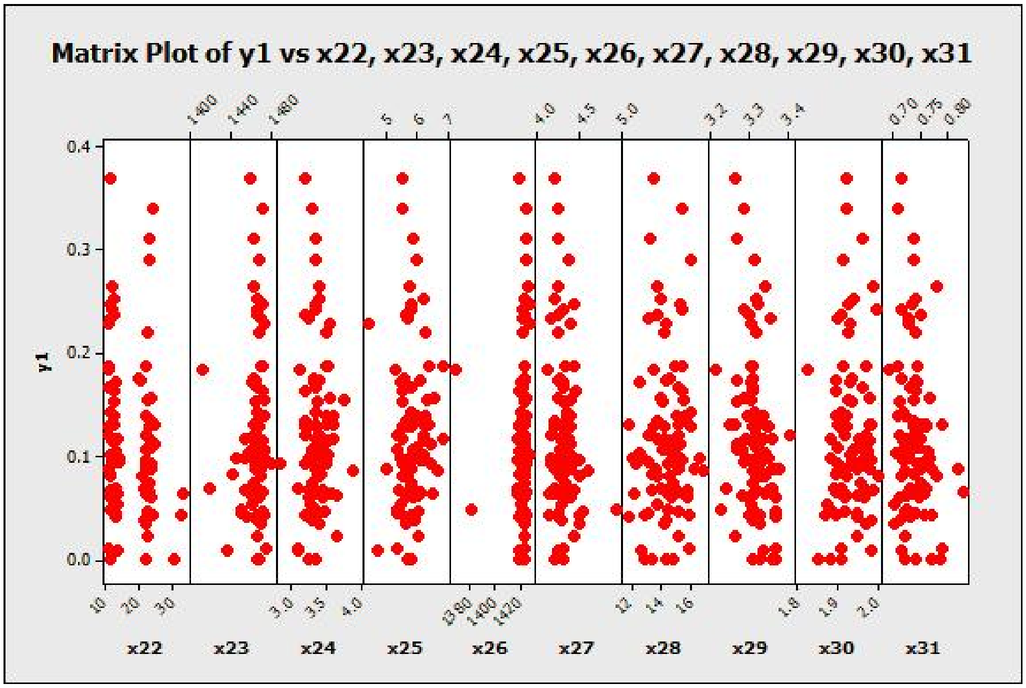

Figure 5.5: Matrix plot of response variable vs. predictor variables ($x_{22} - x_{31}$) for casting data.

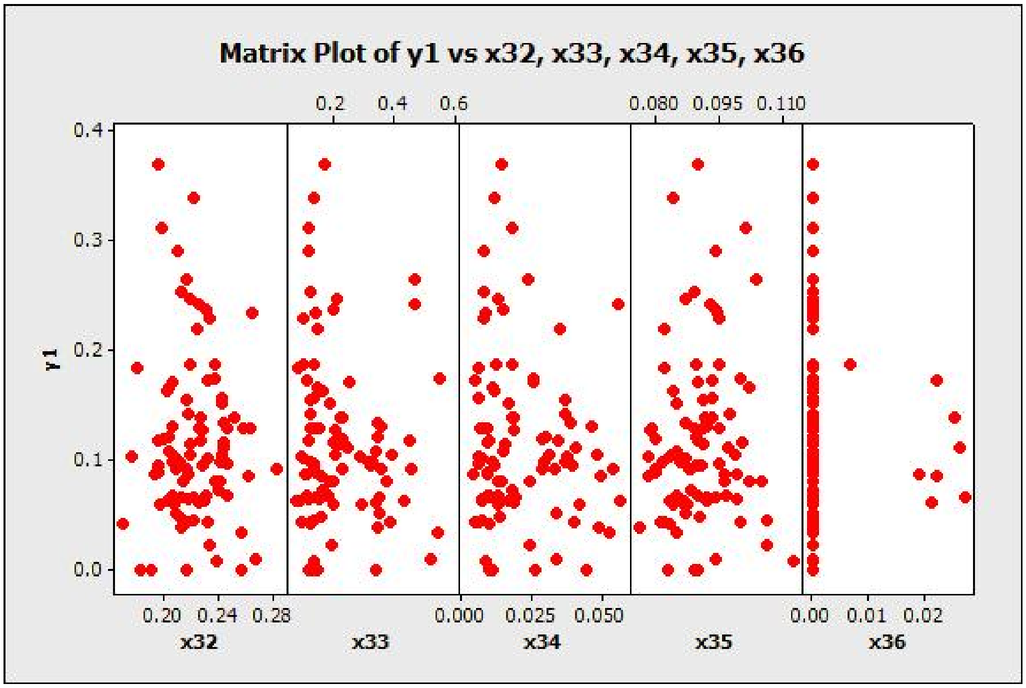

Figure 5.6: Matrix plot of response variable vs. predictor variables ($x_{32} - x_{36}$) for casting data.

APPENDIX C

Figures and Tables of LHS Data

REPLICATION 1 CV 1

Table 5.1: The results of Salford MARS for LHS data (Rep1-CV1).

No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
1	8.3751	2.2017	0.2750	0.9698
2	5.5860	1.8111	0.1252	0.9475
3	4.9931	2.1467	0.1049	0.9036
4	4.4397	2.1621	0.0857	0.8749
5	3.7117	2.2089	0.0619	0.8467
6	3.2703	2.2486	0.0497	0.8189
7	3.0401	2.2508	0.0444	0.7916
8	2.6179	2.1476	0.0341	0.7648
9	2.1788	2.1373	0.0244	0.7384
10	1.7619	2.1328	0.0166	0.7125
11	1.5087	2.2359	0.0126	0.6871
12	1.2909	2.1778	0.0096	0.6621
13	1.1557	2.1504	0.0080	0.6376
14	1.0271	2.1220	0.0065	0.6135
15	0.9639	2.0372	0.0060	0.5899
16	0.9096	2.0411	0.0055	0.5668
17	0.8691	2.0333	0.0053	0.5441
18	0.8584	2.0407	0.0054	0.5219
19	0.8528	2.0476	0.0055	0.5002
20	0.8501	2.0511	0.0057	0.4789
21	0.8480	2.0515	0.0060	0.4581
22	0.8480	2.0515	0.0062	0.4377

No. BF: Number of basis function, Denominator: Denominator of GCV.

Table 5.2: The results of C-MARS for LHS data (Rep1-CV1).

$\sqrt{\bar{M}}$	No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
0.265	22	6.5015	0.265	0.3672	0.4377
0.3	22	6.2446	0.3	0.3387	0.4377
0.35	22	5.9403	0.35	0.3065	0.4377
0.4	22	5.6755	0.4	0.2798	0.4377
0.45	22	5.434	0.45	0.2565	0.4377
0.5	22	5.2083	0.5	0.2356	0.4377
0.55	22	4.9941	0.55	0.2167	0.4377
0.6	22	4.7889	0.6	0.1992	0.4377
0.7	22	4.3994	0.7	0.1681	0.4377
0.8	22	4.0313	0.8	0.1412	0.4377
0.9	22	3.6799	0.9	0.1176	0.4377
1	22	3.3424	1	0.097	0.4377
1.1	22	3.0171	1.1	0.0791	0.4377
1.2	22	2.7031	1.2	0.0635	0.4377
1.25	22	2.5501	1.25	0.0565	0.4377
1.3	22	2.3999	1.3	0.05	0.4377
1.4	22	2.1081	1.4	0.0386	0.4377
1.5	22	1.8292	1.5	0.0291	0.4377
1.6	22	1.5661	1.6	0.0213	0.4377
1.7	22	1.3243	1.7	0.0152	0.4377
1.8	22	1.1138	1.8	0.0108	0.4377
1.9	22	0.9512	1.9	0.0079	0.4377
2	22	0.86	2	0.0064	0.4377
2.1	22	0.8478	2.0509	0.0062	0.4377
2.2	22	0.8478	2.0509	0.0062	0.4377
2.3	22	0.8478	2.0509	0.0062	0.4377

No. BF: Number of basis function, Denominator: Denominator of GCV.

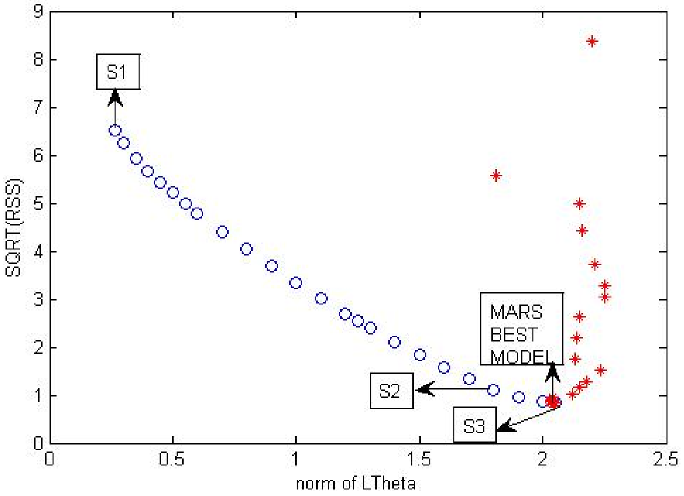

Figure 5.7: Norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for LHS data (Rep1-CV1).

(*: MARS solutions; o: C-MARS solutions)

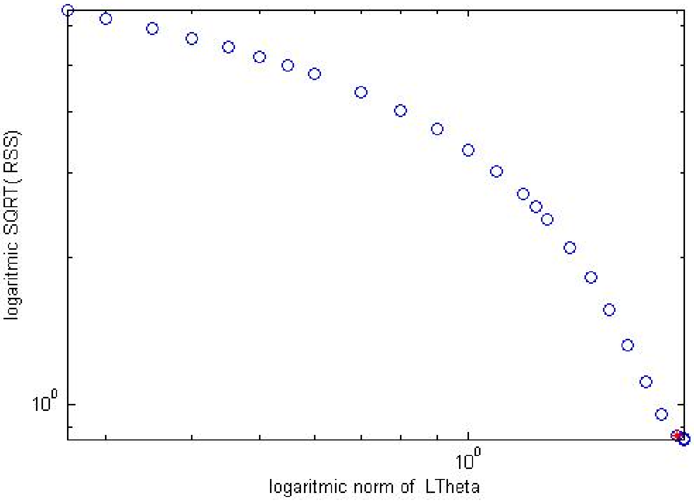

Figure 5.8: A log-log scale, the curve of norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for LHS data (Rep1-CV1).

REPLICATION 1 CV 2

Table 5.3: The results of Salford MARS for LHS data (Rep1-CV2).

No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
1	6.3910	1.8203	0.1614	0.9696
2	5.4280	1.9018	0.1192	0.9471
3	4.9394	1.8291	0.1011	0.9248
4	4.3671	1.8363	0.0809	0.9029
5	3.6609	1.8228	0.0583	0.8812
6	2.9169	1.8562	0.0379	0.8597
7	2.6310	1.8507	0.0316	0.8385
8	2.3304	1.8446	0.0254	0.8176
9	2.0608	1.8336	0.0204	0.7969
10	1.6890	2.0256	0.0154	0.7105
11	1.3677	2.0702	0.0105	0.6849
12	1.1639	2.0815	0.0079	0.6598
13	1.5194	2.0920	0.0139	0.6351
14	1.0563	2.0529	0.0070	0.6109
15	1.0277	2.0286	0.0069	0.5872
16	0.9903	2.0175	0.0067	0.5639
17	0.9705	2.0057	0.0067	0.5412
18	0.9571	2.0377	0.0068	0.5188
19	0.9487	2.0298	0.0069	0.4970
20	0.9471	2.0348	0.0072	0.4756
21	0.9459	2.0350	0.0075	0.4547
22	0.9458	2.0342	0.0079	0.4343
23	0.9458	2.0342	0.0083	0.4143

No. BF: Number of basis function, Denominator: Denominator of GCV.

Table 5.4: The results of C-MARS for LHS data (Rep1-CV2).

$\sqrt{\bar{M}}$	No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
0.265	24	9.0583	0.265	0.7963	0.3948
0.3	24	8.7009	0.3	0.7346	0.3948
0.35	24	8.2551	0.35	0.6613	0.3948
0.4	24	7.8617	0.4	0.5998	0.3948
0.45	24	7.5035	0.45	0.5464	0.3948
0.5	24	7.1702	0.5	0.4989	0.3948
0.55	24	6.8554	0.55	0.4561	0.3948
0.7	24	5.9855	0.7	0.3477	0.3948
0.9	24	4.9336	0.9	0.2362	0.3948
1	24	4.4384	1	0.1912	0.3948
1.1	24	3.9596	1.1	0.1521	0.3948
1.15	24	3.7258	1.15	0.1347	0.3948
1.2	24	3.4958	1.2	0.1186	0.3948
1.25	24	3.2696	1.25	0.1037	0.3948
1.3	24	3.0472	1.3	0.0901	0.3948
1.4	24	2.6152	1.4	0.0664	0.3948
1.5	24	2.2034	1.5	0.0471	0.3948
1.7	24	1.4727	1.7	0.021	0.3948
1.8	24	1.1891	1.8	0.0137	0.3948
1.9	24	1.0039	1.9	0.0098	0.3948
2	24	0.9457	2	0.0087	0.3948
2.1	24	0.9454	2.0119	0.0087	0.3948
2.2	24	0.9454	2.0119	0.0087	0.3948
2.3	24	0.9454	2.0119	0.0087	0.3948

No. BF: Number of basis function, Denominator: Denominator of GCV.

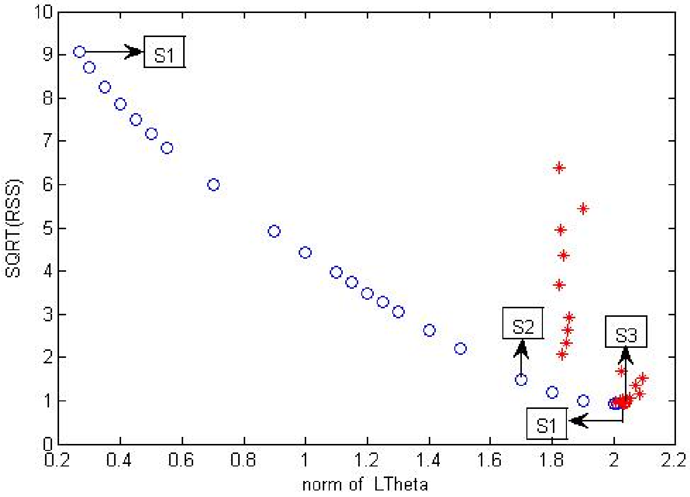

Figure 5.9: Norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for LHS data (Rep1-CV2).

(*: MARS solutions; o: C-MARS solutions)

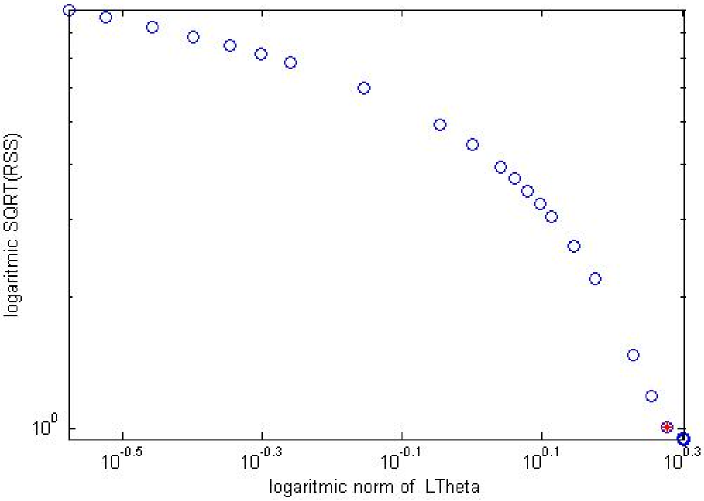

Figure 5.10: A log-log scale, the curve of norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for LHS data (Rep1-CV2).

REPLICATION 1 CV 3

Table 5.5: The results of Salford MARS for LHS data (Rep1-CV3).

No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
1	6.5470	1.9356	0.1714	0.9692
2	5.7802	1.9005	0.1390	0.9314
3	4.6694	1.7505	0.0937	0.9018
4	4.0854	1.7694	0.0741	0.8726
5	3.3285	1.7599	0.0509	0.8438
6	2.8992	1.7801	0.0399	0.8156
7	2.4624	1.8152	0.0298	0.7878
8	1.9886	1.8235	0.0202	0.7605
9	1.6056	1.8081	0.0136	0.7337
10	1.3763	1.9469	0.0104	0.7074
11	1.1813	2.0038	0.0079	0.6816
12	1.0777	2.0178	0.0069	0.6562
13	1.0120	1.9790	0.0063	0.6313
14	0.9602	1.9697	0.0059	0.6069
15	0.9153	1.9773	0.0056	0.5830
16	0.8840	1.9281	0.0054	0.5596
17	0.8577	1.8846	0.0053	0.5366
18	0.8321	1.9061	0.0052	0.5142
19	0.8198	1.9130	0.0053	0.4922
20	0.8134	1.9188	0.0054	0.4707
21	0.8035	1.9045	0.0056	0.4496
22	0.8002	1.9014	0.0058	0.4291
23	0.7993	1.9054	0.0061	0.4090
24	0.7968	1.9044	0.0063	0.3894
25	0.7966	1.9021	0.0066	0.3703

No. BF: Number of basis function, Denominator: Denominator of GCV.

Table 5.6: The results of C-MARS for LHS data (Rep1-CV3).

$\sqrt{\bar{M}}$	No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
0.265	25	6.0754	0.265	0.3863	0.3703
0.3	25	5.747	0.3	0.3457	0.3703
0.4	25	5.0452	0.4	0.2664	0.3703
0.45	25	4.7686	0.45	0.238	0.3703
0.55	25	4.2936	0.55	0.193	0.3703
0.7	25	3.6952	0.7	0.1429	0.3703
0.8	25	3.3426	0.8	0.1169	0.3703
0.9	25	3.0156	0.9	0.0952	0.3703
1	25	2.7087	1	0.0768	0.3703
1.1	25	2.4189	1.1	0.0612	0.3703
1.15	25	2.2797	1.15	0.0544	0.3703
1.2	25	2.1442	1.2	0.0481	0.3703
1.25	25	2.0123	1.25	0.0424	0.3703
1.3	25	1.8841	1.3	0.0372	0.3703
1.4	25	1.6393	1.4	0.0281	0.3703
1.5	25	1.4123	1.5	0.0209	0.3703
1.6	25	1.208	1.6	0.0153	0.3703
1.8	25	0.9063	1.8	0.0086	0.3703
1.9	25	0.8381	1.9	0.0074	0.3703
2	25	0.8297	1.9499	0.0072	0.3703
2.1	25	0.8297	1.95	0.0072	0.3703
2.2	25	0.8297	1.9499	0.0072	0.3703
2.3	25	0.8297	1.9499	0.0072	0.3703

No. BF: Number of basis function, Denominator: Denominator of GCV.

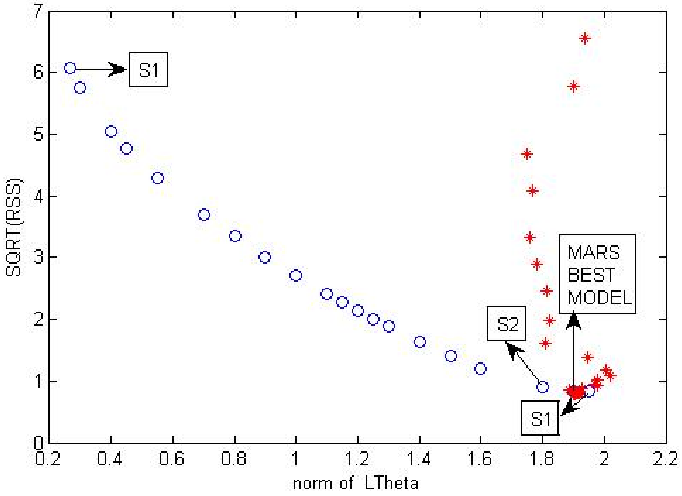

Figure 5.11: Norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for LHS data (Rep1-CV3).

(*: MARS solutions; o: C-MARS solutions)

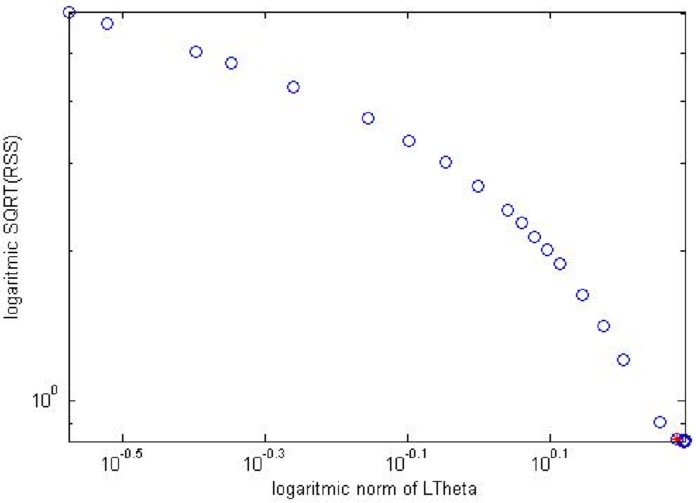

Figure 5.12: A log-log scale, the curve of norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for LHS data (Rep1-CV3).

REPLICATION 2 CV 1

Table 5.7: The results of Salford MARS for LHS data (Rep2-CV1).

No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
1	7.2598	1.9277	0.2074	0.9697
2	6.1564	1.7808	0.1527	0.9473
3	5.8271	1.7848	0.1435	0.9032
4	4.7911	1.8344	0.1002	0.8744
5	4.1722	1.8471	0.0785	0.8461
6	3.3968	1.8524	0.0538	0.8183
7	2.7688	1.8773	0.0370	0.7909
8	2.3230	1.8756	0.0270	0.7640
9	2.0295	1.8620	0.0213	0.7375
10	1.6256	1.7203	0.0142	0.7115
11	1.4658	1.6859	0.0120	0.6860
12	1.2874	1.6843	0.0096	0.6609
13	1.1592	1.6796	0.0081	0.6363
14	1.0771	1.7084	0.0072	0.6122
15	1.0113	1.7374	0.0066	0.5886
16	0.9810	1.6976	0.0065	0.5654
17	0.9622	1.6889	0.0065	0.5426
18	0.9571	1.7050	0.0067	0.5204
19	0.9534	1.7029	0.0070	0.4986
20	0.9534	1.7049	0.0073	0.4773

No. BF: Number of basis function, Denominator: Denominator of GCV.

Table 5.8: The results of C-MARS for LHS data (Rep2-CV1).

$\sqrt{\bar{M}}$	No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
1	20	3.4776	1	0.0967	0.4773
1.1	20	2.9197	1.1	0.0682	0.4773
1.15	20	2.6579	1.15	0.0565	0.4773
1.2	20	2.4077	1.2	0.0464	0.4773
1.25	20	2.1696	1.25	0.0376	0.4773
1.3	20	1.9447	1.3	0.0302	0.4773
1.4	20	1.541	1.4	0.019	0.4773
1.5	20	1.2187	1.5	0.0119	0.4773
1.6	20	1.0138	1.6	0.0082	0.4773
1.7	20	0.9524	1.7	0.0073	0.4773
1.8	20	0.9523	1.7055	0.0073	0.4773
1.9	20	0.9523	1.7055	0.0073	0.4773
2	20	0.9523	1.7055	0.0073	0.4773
2.1	20	0.9523	1.7055	0.0073	0.4773

No. BF: Number of basis function, Denominator: Denominator of GCV.

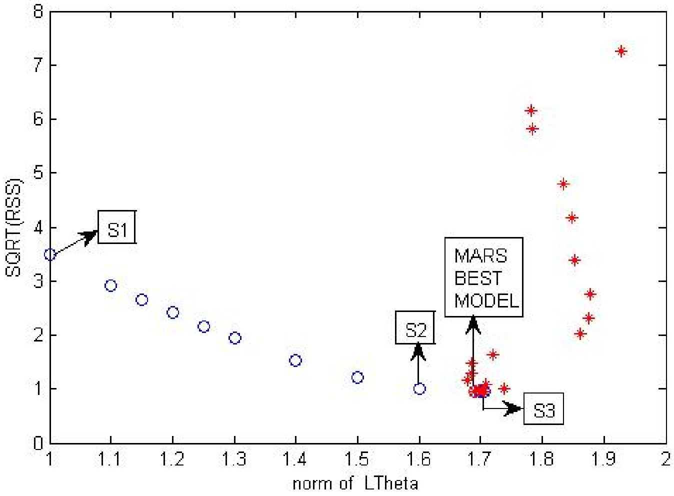

Figure 5.13: Norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for LHS data (Rep2-CV1).

(*: MARS solutions; o: C-MARS solutions)

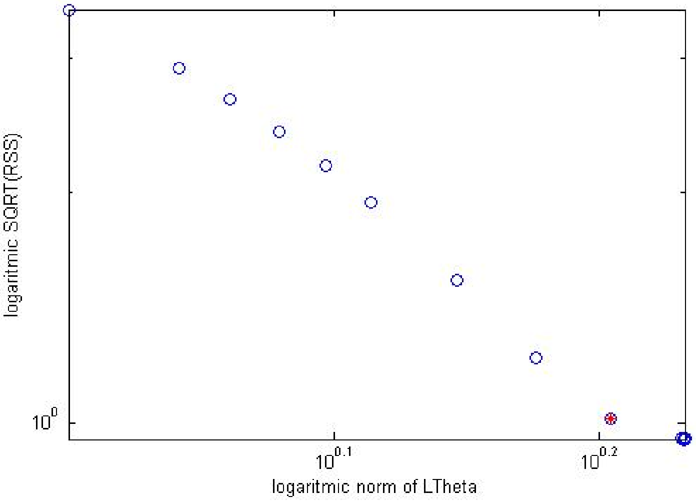

Figure 5.14: A log-log scale, the curve of norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for LHS data (Rep2-CV1).

REPLICATION 2 CV 2

Table 5.9: The results of Salford MARS for LHS data (Rep2-CV2).

No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
1	5.7892	1.8252	0.1324	0.9696
2	4.9299	1.8209	0.0999	0.9322
3	4.5277	1.7501	0.0870	0.9029
4	4.1625	1.7523	0.0760	0.8740
5	3.5392	1.7360	0.0568	0.8456
6	2.9421	1.7629	0.0406	0.8176
7	2.4838	1.8126	0.0299	0.7901
8	2.0207	1.8071	0.0205	0.7631
9	1.7641	1.8726	0.0162	0.7366
10	1.5679	1.9353	0.0133	0.7105
11	1.4538	1.8815	0.0118	0.6849
12	1.3078	1.8945	0.0099	0.6598
13	1.2343	1.9077	0.0092	0.6351
14	1.1811	1.8819	0.0087	0.6109
15	1.1523	1.9212	0.0087	0.5872
16	1.1308	1.9238	0.0087	0.5639
17	1.1079	1.9395	0.0087	0.5412
18	1.0983	1.9233	0.0089	0.5188
19	1.0892	1.9043	0.0091	0.4970
20	1.0816	1.8937	0.0094	0.4756
21	1.0782	1.8946	0.0098	0.4547
22	1.0747	1.8933	0.0102	0.4343
23	1.0741	1.8943	0.0107	0.4143
24	1.0739	1.8944	0.0112	0.3948

No. BF: Number of basis function, Denominator: Denominator of GCV.

Table 5.10: The results of C-MARS for LHS data (Rep2-CV2).

$\sqrt{\bar{M}}$	No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
0.5	24	6.0425	0.5	0.3543	0.3948
0.55	24	5.753	0.55	0.3212	0.3948
0.6	24	5.473	0.6	0.2907	0.3948
0.8	24	4.4315	0.8	0.1906	.3948
0.9	24	3.9531	0.9	0.1516	0.3948
1	24	3.5016	1	0.119	0.3948
1.1	24	3.0772	1.1	0.0919	0.3948
1.15	24	2.8752	1.15	0.0802	0.3948
1.2	24	2.6804	1.2	0.0697	0.3948
1.25	24	2.4928	1.25	0.0603	0.3948
1.3	24	2.3128	1.3	0.0519	0.3948
1.4	24	1.9773	1.4	0.0379	0.3948
1.7	24	1.2313	1.7	0.0147	0.3948
1.8	24	1.1103	1.8	0.012	0.3948
1.9	24	1.0732	1.8938	0.0112	0.3948
2	24	1.0732	1.8938	0.0112	0.3948
2.1	24	1.0732	1.8939	0.0112	0.3948

No. BF: Number of basis function, Denominator: Denominator of GCV.

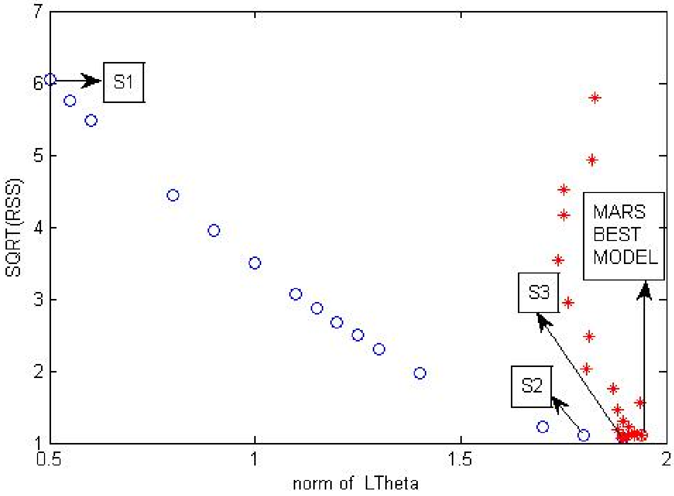

Figure 5.15: Norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for LHS data (Rep2-CV2).

(*: MARS solutions; o: C-MARS solutions)

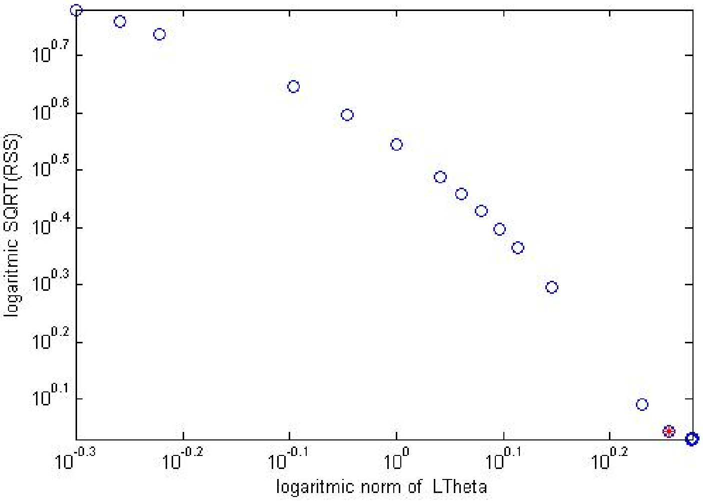

Figure 5.16: A log-log scale, the curve of norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for LHS data (Rep2-CV2).

REPLICATION 2 CV 3

Table 5.11: The results of Salford MARS for LHS data (Rep2-CV3).

No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
1	8.7096	1.7923	0.3021	0.9694
2	5.5985	1.8149	0.1278	0.9467
3	4.8013	1.8793	0.0963	0.9243
4	4.1940	1.8779	0.0753	0.9021
5	3.4572	1.8698	0.0524	0.8803
6	2.9288	1.8996	0.0386	0.8587
7	2.6148	1.9073	0.0315	0.8373
8	2.2879	1.9080	0.0248	0.8163
9	1.9853	1.9018	0.0191	0.7955
10	1.7059	2.0441	0.0145	0.7749
11	1.4949	2.0603	0.0114	0.7547
12	1.4464	2.0651	0.0110	0.7347
13	1.2617	2.0674	0.0086	0.7150
14	1.1078	2.0658	0.0068	0.6955
15	1.0388	1.9720	0.0062	0.6763
16	0.9950	1.9686	0.0058	0.6574
17	0.9685	1.9577	0.0057	0.6388
18	0.9575	1.9514	0.0057	0.6204
19	0.9517	1.9590	0.0058	0.6023
20	0.9482	1.9581	0.0059	0.5844
21	0.9438	1.9606	0.0061	0.5669
22	0.9410	1.9653	0.0062	0.5495
23	0.9384	1.9688	0.0064	0.5325

No. BF: Number of basis function, Denominator: Denominator of GCV.

Table 5.12: The results of C-MARS for LHS data (Rep2-CV3).

$\sqrt{M}$	No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
0.3	23	6.2435	0.3	0.3664	0.4108
0.35	23	5.9408	0.35	0.3317	0.4108
0.4	23	5.6721	0.4	0.3024	0.4108
0.45	23	5.4248	0.45	0.2766	0.4108
0.5	23	5.1927	0.5	0.2534	0.4108
0.55	23	4.9721	0.55	0.2324	0.4108
0.6	23	4.7606	0.6	0.213	0.4108
0.7	23	4.359	0.7	0.1786	0.4108
0.8	23	3.9797	0.8	0.1489	0.4108
0.9	23	3.6176	0.9	0.123	0.4108
1	23	3.27	1	0.1005	0.4108
1.1	23	2.9351	1.1	0.081	0.4108
1.15	23	2.7722	1.15	0.0722	0.4108
1.2	23	2.6122	1.2	0.0641	0.4108
1.25	23	2.4553	1.25	0.0567	0.4108
1.3	23	2.3017	1.3	0.0498	0.4108
1.4	23	2.0047	1.4	0.0378	0.4108
1.5	23	1.7245	1.5	0.028	0.4108
1.6	23	1.4666	1.6	0.0202	0.4108
1.7	23	1.2411	1.7	0.0145	0.4108
1.8	23	1.0644	1.8	0.0106	0.4108
1.9	23	0.9592	1.9	0.0086	0.4108
2	23	0.9382	1.9683	0.0083	0.4108
2.1	23	0.9382	1.9683	0.0083	0.4108

No. BF: Number of basis function, Denominator: Denominator of GCV.

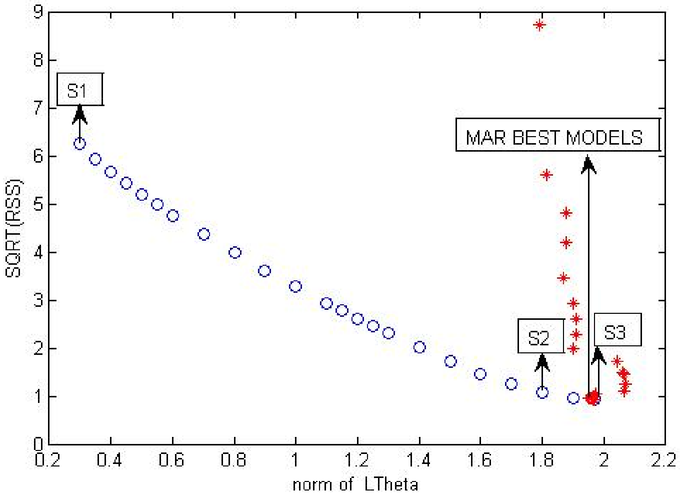

Figure 5.17: Norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for LHS data (Rep2-CV3).

(*: MARS solutions; o: C-MARS solutions)

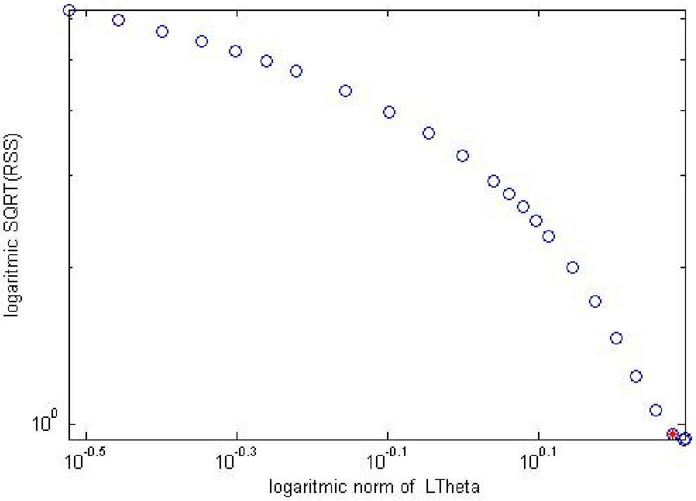

Figure 5.18: A log-log scale, the curve of norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for LHS data (Rep2-CV3).

REPLICATION 3 CV 1

Table 5.13: The results of Salford MARS for LHS data (Rep3-CV1).

No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
1	6.6977	1.9253	0.1766	0.9697
2	5.6394	1.7648	0.1281	0.9473
3	5.0141	1.9076	0.1062	0.9032
4	4.4867	1.8879	0.0879	0.8744
5	3.8604	1.9053	0.0672	0.8461
6	3.4779	1.9443	0.0564	0.8183
7	2.9540	1.9421	0.0421	0.7909
8	2.5216	1.9437	0.0318	0.7640
9	2.1377	1.9531	0.0236	0.7375
10	1.8324	1.9394	0.0180	0.7115
11	1.5303	1.9975	0.0130	0.6860
12	1.3816	1.9691	0.0110	0.6609
13	1.1732	1.9879	0.0083	0.6363
14	1.0785	1.9895	0.0073	0.6122
15	1.0245	1.9703	0.0068	0.5886
16	0.9850	1.9317	0.0066	0.5654
17	0.9592	1.9053	0.0065	0.5426
18	0.9421	1.9126	0.0065	0.5204
19	0.9325	1.9190	0.0067	0.4986
20	0.9274	1.9207	0.0069	0.4773
21	0.9262	1.9189	0.0072	0.4564
22	0.9248	1.9150	0.0075	0.4360
23	0.9241	1.9184	0.0078	0.4161
24	0.9237	1.9169	0.0082	0.3966
25	0.9262	1.9175	0.0087	0.3776
26	0.9258	1.9173	0.0091	0.3591

No. BF: Number of basis function, Denominator: Denominator of GCV.

Table 5.14: The results of C-MARS for LHS data (Rep3-CV1).

$\sqrt{\bar{M}}$	No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
0.5	26	6.4138	0.5	0.4373	0.3591
0.55	26	6.1261	0.55	0.3989	0.3591
0.7	26	5.3219	0.7	0.301	0.3591
0.8	26	4.8226	0.8	0.2472	0.3591
0.85	26	4.5817	0.85	0.2231	0.3591
0.9	26	4.3459	0.9	0.2007	0.3591
0.95	26	4.1147	0.95	0.18	0.3591
1	26	3.888	1	0.1607	0.3591
1.05	26	3.6654	1.05	0.1428	0.3591
1.1	26	3.4469	1.1	0.1263	0.3591
1.15	26	3.2323	1.15	0.1111	0.3591
1.2	26	3.0217	1.2	0.0971	0.3591
1.25	26	2.8151	1.25	0.0842	0.3591
1.3	26	2.6128	1.3	0.0726	0.3591
1.35	26	2.415	1.35	0.062	0.3591
1.4	26	2.2221	1.4	0.0525	0.3591
1.7	26	1.2297	1.7	0.0161	0.3591
1.8	26	1.0191	1.8	0.011	0.3591

No. BF: Number of basis function, Denominator: Denominator of GCV.

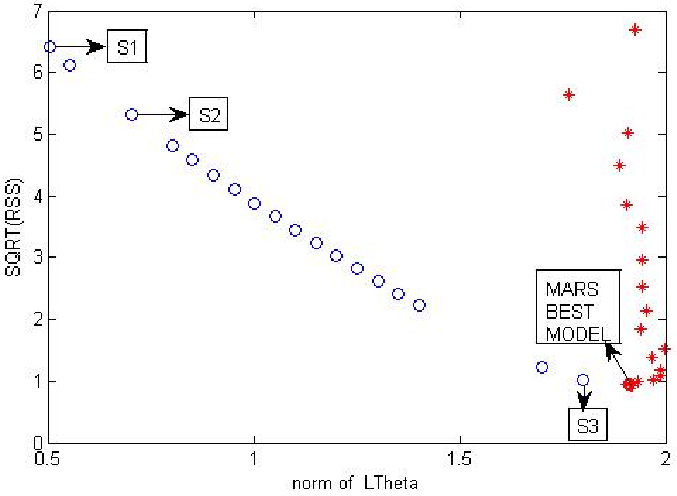

Figure 5.19: Norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for LHS data (Rep3-CV1).

(*: MARS solutions; o: C-MARS solutions)

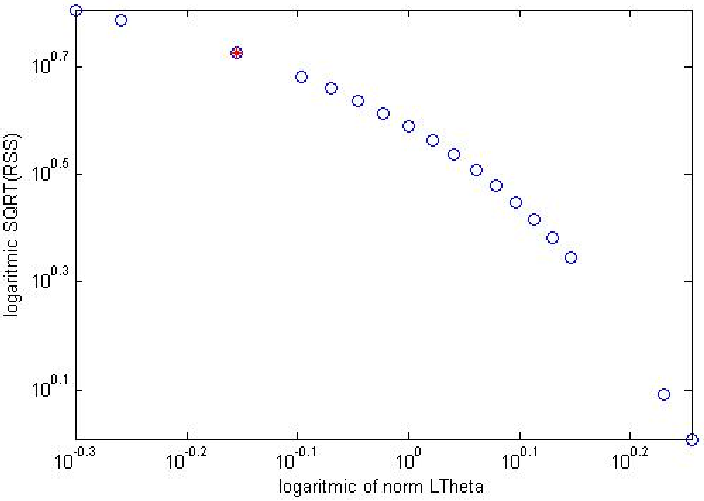

Figure 5.20: A log-log scale, the curve of norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for LHS data (Rep3-CV1).

REPLICATION 3 CV 2

Table 5.15: The results of Salford MARS for LHS data (Rep3-CV2).

No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
1	6.4559	1.8651	0.1647	0.9696
2	5.6460	1.9202	0.1290	0.9471
3	4.9750	1.8244	0.1025	0.9248
4	4.4526	1.8159	0.0841	0.9029
5	3.7742	1.8400	0.0619	0.8812
6	3.0704	1.8783	0.0420	0.8597
7	2.7224	1.8665	0.0339	0.8385
8	2.4020	1.8724	0.0270	0.8176
9	2.0620	1.8631	0.0204	0.7969
10	1.8451	1.9803	0.0184	0.7105
11	1.7879	1.9755	0.0179	0.6849
12	1.6036	1.9857	0.0149	0.6598
13	1.3977	2.0466	0.0118	0.6351
14	1.1581	2.0417	0.0084	0.6109
15	1.0869	2.0540	0.0077	0.5872
16	1.0513	2.0565	0.0075	0.5639
17	1.0140	2.0378	0.0073	0.5412
18	0.9967	2.0433	0.0073	0.5188
19	0.9685	2.0375	0.0072	0.4970
20	0.9510	2.0333	0.0073	0.4756
21	0.9335	2.0317	0.0073	0.4547
22	0.9170	2.0554	0.0074	0.4343
23	0.9110	2.0596	0.0077	0.4143
24	0.9100	2.0598	0.0080	0.3948
25	0.9081	2.0648	0.0084	0.3758
26	0.9078	2.0621	0.0088	0.3572

No. BF: Number of basis function, Denominator: Denominator of GCV.

Table 5.16: The results of C-MARS for LHS data (Rep3-CV2).

$\sqrt{\bar{M}}$	No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
0.265	26	11.3581	0.265	1.3836	0.3572
0.3	26	11.0462	0.3	1.3086	0.3572
0.35	26	10.6111	0.35	1.2076	0.3572
0.4	26	10.1864	0.4	1.1128	0.3572
0.45	26	9.7707	0.45	1.0239	0.3572
0.5	26	9.3631	0.5	0.9402	0.3572
0.55	26	8.9631	0.55	0.8616	0.3572
0.6	26	8.57	0.6	0.7877	0.3572
0.8	26	7.062	0.8	0.5349	0.3572
0.9	26	6.3437	0.9	0.4316	0.3572
1	26	5.6481	1	0.3421	0.3572
1.1	26	4.9752	1.1	0.2655	0.3572
1.15	26	4.6476	1.15	0.2317	0.3572
1.2	26	4.3262	1.2	0.2007	0.3572
1.25	26	4.0114	1.25	0.1726	0.3572
1.3	26	3.7037	1.3	0.1471	0.3572
1.4	26	3.1125	1.4	0.1039	0.3572
1.5	26	2.5608	1.5	0.0703	0.3572
1.6	26	2.0614	1.6	0.0456	0.3572
1.7	26	1.6316	1.7	0.0286	0.3572
1.8	26	1.29	1.8	0.0178	0.3572
1.9	26	1.0521	1.9	0.0119	0.3572
2	26	0.9278	2	0.0092	0.3572
2.1	26	0.9075	2.0622	0.0088	0.3572
2.2	26	0.9075	2.0622	0.0088	0.3572
2.3	26	0.9075	2.0622	0.0088	0.3572

No. BF: Number of basis function, Denominator: Denominator of GCV.

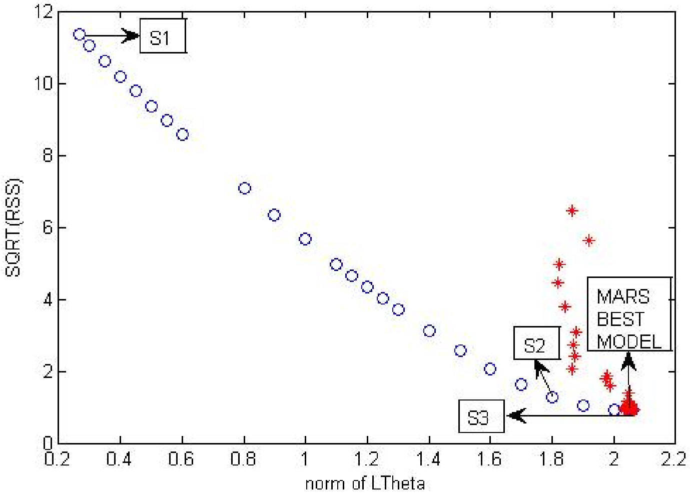

Figure 5.21: Norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for LHS data (Rep3-CV2).

(*: MARS solutions; o: C-MARS solutions)

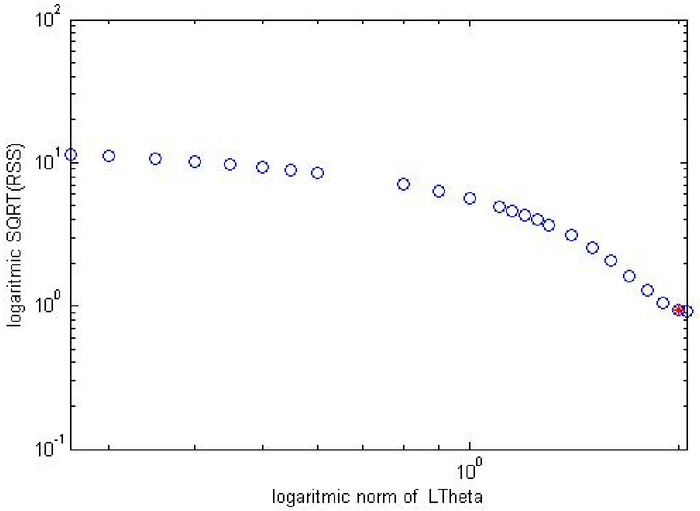

Figure 5.22: A log-log scale, the curve of norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for LHS data (Rep3-CV2).

REPLICATION 3 CV 3

Table 5.17: The results of Salford MARS for LHS data (Rep3-CV3).

No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
1	8.6712	2.2557	0.2995	0.9694
2	5.8419	1.8342	0.1392	0.9467
3	4.7287	1.7911	0.0957	0.9021
4	4.1811	1.7863	0.0773	0.8730
5	3.4723	1.7864	0.0551	0.8444
6	2.8727	1.8050	0.0390	0.8163
7	2.4819	1.8022	0.0302	0.7886
8	2.0395	2.0360	0.0211	0.7614
9	1.6186	2.0367	0.0138	0.7347
10	1.3604	2.1131	0.0101	0.7085
11	1.2272	2.0595	0.0085	0.6827
12	1.1086	2.0591	0.0072	0.6574
13	1.0414	2.1357	0.0066	0.6326
14	0.9855	2.0903	0.0062	0.6083
15	0.9324	2.1021	0.0057	0.5844
16	0.9080	2.1345	0.0057	0.5611
17	0.8895	2.0940	0.0057	0.5382
18	0.8726	2.0848	0.0057	0.5157
19	0.8576	2.0743	0.0058	0.4938
20	0.8492	2.0677	0.0059	0.4723
21	0.8348	2.0923	0.0060	0.4513
22	0.8228	2.1109	0.0061	0.4308
23	0.8155	2.1079	0.0063	0.4108
24	0.8096	2.1018	0.0065	0.3912

No. BF: Number of basis function, Denominator: Denominator of GCV.

Table 5.18: The results of C-MARS for LHS data (Rep3-CV3).

$\sqrt{\bar{M}}$	No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
0.3	24	4.6825	0.3	0.2164	0.3912
0.35	24	4.4841	0.35	0.1984	0.3912
0.4	24	4.3116	0.4	0.1835	0.3912
0.45	24	4.1534	0.45	0.1702	0.3912
0.5	24	4.0043	0.5	0.1582	0.3912
0.55	24	3.8615	0.55	0.1472	0.3912
0.6	24	3.7232	0.6	0.1368	0.3912
0.7	24	3.4564	0.7	0.1179	0.3912
0.8	24	3.1989	0.8	0.101	0.3912
1.15	24	2.3451	1.15	0.0543	0.3912
1.2	24	2.2283	1.2	0.049	0.3912
1.25	24	2.113	1.25	0.0441	0.3912
1.3	24	1.9992	1.3	0.0394	0.3912
1.5	24	1.5638	1.5	0.0241	0.3912
1.6	24	1.3627	1.6	0.0183	0.3912
1.7	24	1.179	1.7	0.0137	0.3912
1.9	24	0.901	1.9	0.008	0.3912
2	24	0.83	2	0.0068	0.3912
2.1	24	0.8091	2.1	0.0065	0.3912
2.2	24	0.8091	2.1016	0.0065	0.3912
2.3	24	0.8091	2.1016	0.0065	0.3912
2.4	24	0.8091	2.1016	0.0065	0.3912

No. BF: Number of basis function, Denominator: Denominator of GCV.

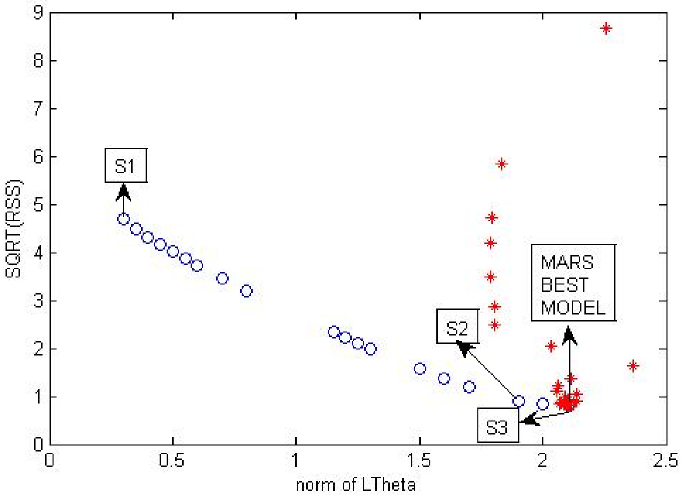

Figure 5.23: Norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for LHS data (Rep3-CV3).

(*: MARS solutions; o: C-MARS solutions)

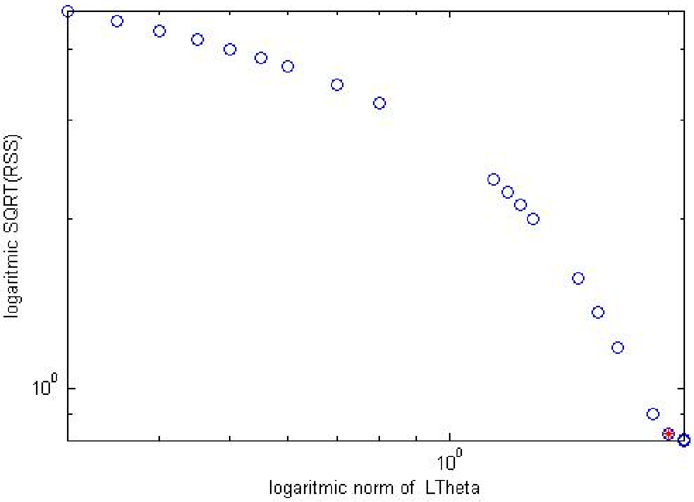

Figure 5.24: A log-log scale, the curve of norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for LHS data (Rep3-CV3).

APPENDIX D

Figures and Tables of US Data

REPLICATION 1 TRAIN 1

Table 5.19: The results of Salford MARS for US data (Rep1-CV1).

No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
1	2.7475	1.8294	0.1233	0.8874
2	2.1375	1.9349	0.0820	0.8074
3	1.7808	1.9344	0.0629	0.7311
4	1.4815	1.9095	0.0483	0.6587
5	1.2799	1.8719	0.0402	0.5900
6	1.0364	1.9140	0.0296	0.5251
7	0.7525	2.0373	0.0244	0.3361
8	0.4125	2.0417	0.0091	0.2722
9	0.3377	1.9627	0.0077	0.2151
10	0.3026	1.9363	0.0081	0.1647
11	0.2895	1.9141	0.0100	0.1210
12	0.2840	1.9418	0.0139	0.0840
13	0.2825	1.9411	0.0215	0.0538
14	0.2804	1.9411	0.0377	0.0302
15	0.2804	1.9426	0.0847	0.0134

No. BF: Number of basis function, Denominator: Denominator of GCV.

Table 5.20: The results of C-MARS for US data (Rep1-CV1).

$\sqrt{\bar{M}}$	No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
0.4	15	2.8138	0.4	8.5357	0.0134
0.45	15	2.6224	0.45	7.4141	0.0134
0.5	15	2.4585	0.5	6.5165	0.0134
0.6	15	2.1823	0.6	5.1347	0.0134
0.7	15	1.9495	0.7	4.0974	0.0134
0.8	15	1.7442	0.8	3.28	0.0134
0.9	15	1.5581	0.9	2.6173	0.0134
1	15	1.3859	1	2.0707	0.0134
1.2	15	1.0713	1.2	1.2374	0.0134
1.3	15	0.9257	1.3	0.9238	0.0134
1.4	15	0.7869	1.4	0.6676	0.0134
1.5	15	0.6555	1.5	0.4632	0.0134
1.6	15	0.5329	1.6	0.3062	0.0134
1.7	15	0.4233	1.7	0.1932	0.0134
1.8	15	0.3354	1.8	0.1213	0.0134
2	15	0.28	1.9431	0.0846	0.0134
2.1	15	0.28	1.9431	0.0846	0.0134

No. BF: Number of basis function, Denominator: Denominator of GCV.

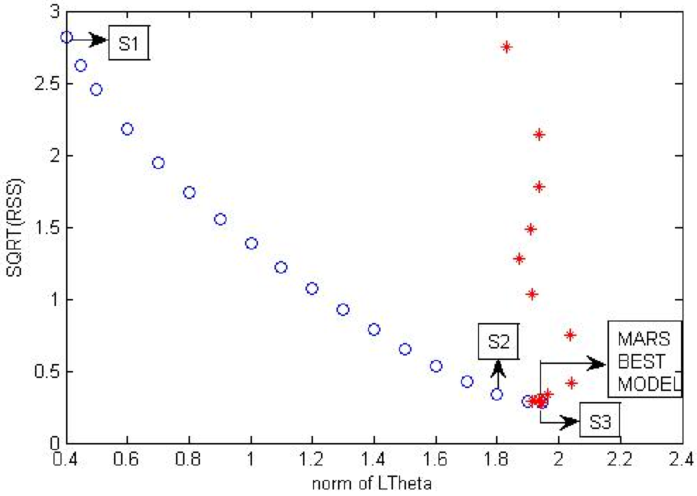

Figure 5.25: Norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for US data (Rep1-CV1).

(*: MARS solutions; o: C-MARS solutions)

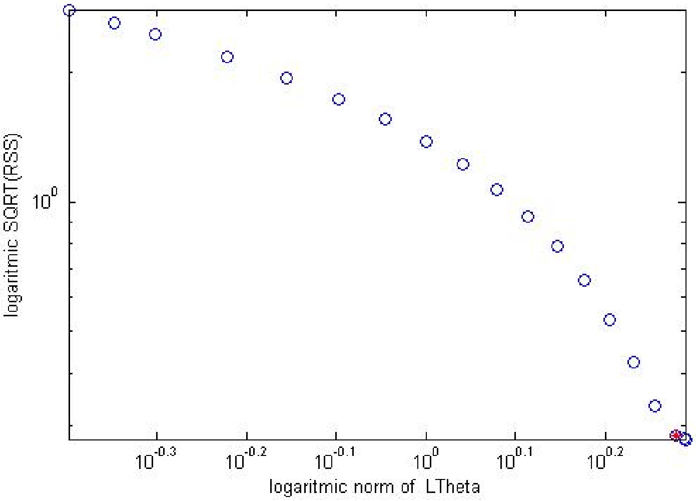

Figure 5.26: A log-log scale, the curve of norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for US data (Rep1-CV1).

REPLICATION 1 TRAIN 2

Table 5.21: The results of Salford MARS for US data (Rep1-CV2).

No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
1	2.0412	1.7765	0.0715	0.8825
2	1.7020	1.8096	0.0549	0.7991
3	1.3903	1.7945	0.0407	0.7199
4	1.2117	1.7850	0.0345	0.6449
5	0.9176	1.8212	0.0222	0.5739
6	0.6608	1.8156	0.0130	0.5071
7	0.6147	1.4453	0.0129	0.4444
8	0.5699	1.3977	0.0197	0.2500
9	0.4807	1.4738	0.0181	0.1931
10	0.4366	1.3875	0.0201	0.1435
11	0.4047	1.3683	0.0245	0.1012
12	0.3265	1.3714	0.0244	0.0663
13	0.3069	1.3659	0.0368	0.0388
14	0.2912	1.4099	0.0691	0.0186
15	0.2830	1.4144	0.2114	0.0057
16	0.2708	1.3682	4.8413	0.0002
17	0.2602	1.2977	0.4966	0.0021
18	0.2560	1.2984	0.0883	0.0112
19	0.2560	1.3018	0.0358	0.0278

No. BF: Number of basis function, Denominator: Denominator of GCV.

Table 5.22: The results of C-MARS for US data (Rep1-CV2).

$\sqrt{\bar{M}}$	No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
0.265	22	2.2584	0.265	0.6363	0.1214
0.3	22	1.9923	0.3	0.4952	0.1214
0.35	22	1.7158	0.35	0.3673	0.1214
0.4	22	1.4993	0.4	0.2804	0.1214
0.45	22	1.3174	0.45	0.2165	0.1214
0.5	22	1.1602	0.5	0.1679	0.1214
0.55	22	1.0229	0.55	0.1305	0.1214
0.6	22	0.9025	0.6	0.1016	0.1214
1.1	22	0.283	1.1	0.01	0.1214
1.15	22	0.2663	1.15	0.0089	0.1214
1.2	22	0.2568	1.2	0.0082	0.1214
1.25	22	0.2535	1.25	0.008	0.1214
1.3	22	0.2535	1.3	0.008	0.1214

No. BF: Number of basis function, Denominator: Denominator of GCV.

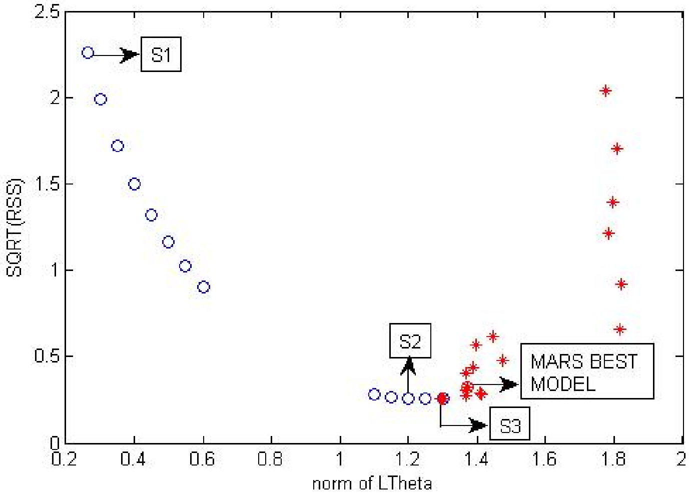

Figure 5.27: Norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for US data (Rep1-CV2).

(*: MARS solutions; o: C-MARS solutions)

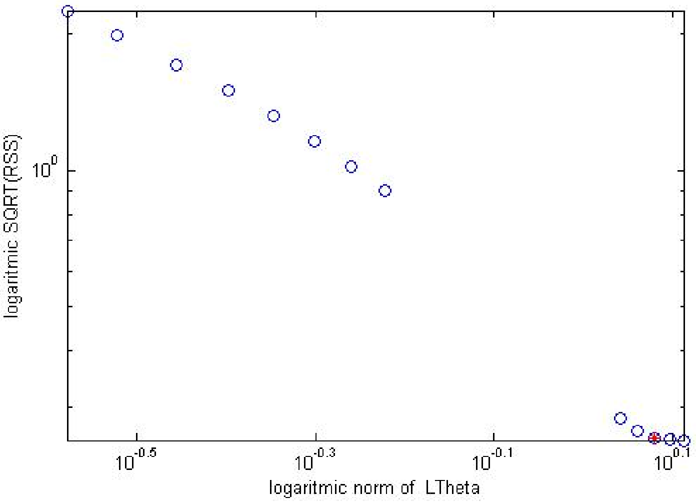

Figure 5.28: A log-log scale, the curve of norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for US data (Rep1-CV2).

REPLICATION 1 TRAIN 3

Table 5.23: The results of Salford MARS for US data (Rep1-CV3).

No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
1	2.2418	1.7626	0.0878	0.8807
2	1.8068	1.7677	0.0631	0.7962
3	1.5164	1.8288	0.0494	0.7160
4	1.2315	1.8551	0.0365	0.6400
5	0.9558	1.8376	0.0247	0.5683
6	0.5563	1.8482	0.0095	0.5008
7	0.5142	1.8099	0.0093	0.4376
8	0.4594	1.4558	0.0086	0.3787
9	0.4578	1.4481	0.0100	0.3240

No. BF: Number of basis function, Denominator: Denominator of GCV.

Table 5.24: The results of C-MARS for US data (Rep1-CV3).

$\sqrt{\bar{M}}$	No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
0.95	11	2.4106	0.95	0.393	0.2275
1	11	2.135	1	0.3083	0.2275
1.05	11	1.8648	1.05	0.2352	0.2275
1.1	11	1.6014	1.1	0.1735	0.2275
1.15	11	1.3474	1.15	0.1228	0.2275
1.2	11	1.1064	1.2	0.0828	0.2275
1.25	11	0.8849	1.25	0.053	0.2275
1.3	11	0.6932	1.3	0.0325	0.2275
1.35	11	0.5478	1.35	0.0203	0.2275
1.4	11	0.4688	1.4	0.0149	0.2275
1.45	11	0.4577	1.45	0.0142	0.2275
1.5	11	0.4577	1.5	0.0142	0.2275

No. BF: Number of basis function, Denominator: Denominator of GCV.

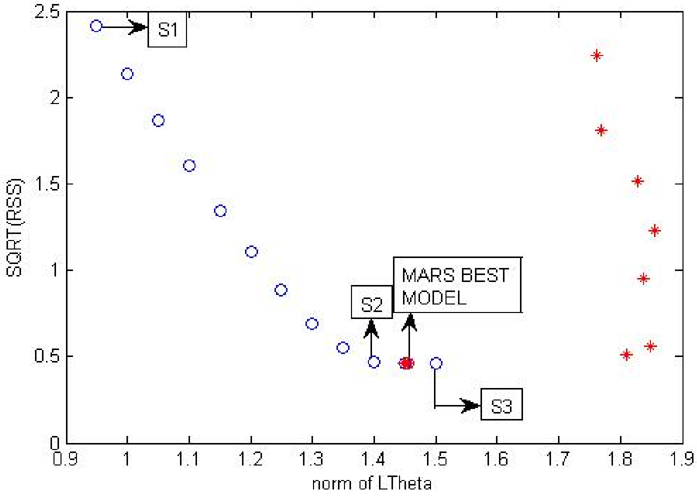

Figure 5.29: Norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for US data (Rep1-CV3).

(*: MARS solutions; o: C-MARS solutions)

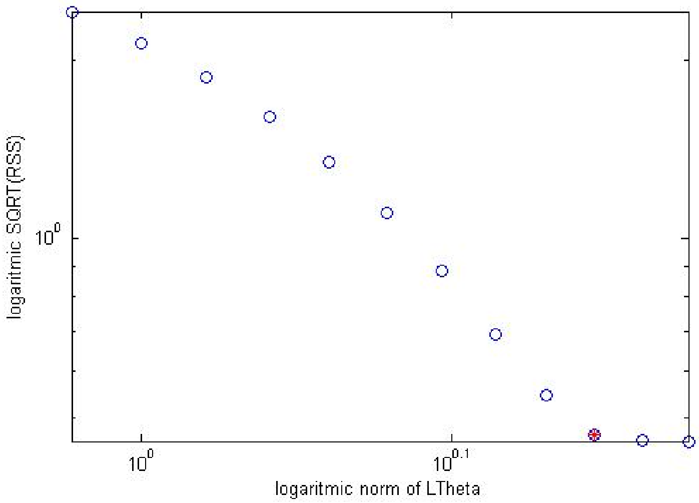

Figure 5.30: A log-log scale, the curve of norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for US data (Rep1-CV3).

REPLICATION 2 TRAIN 1

Table 5.25: The results of Salford MARS for US data (Rep2-CV1).

No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
1	1.9893	1.7628	0.0646	0.8874
2	1.7084	1.7923	0.0524	0.8074
3	1.4309	1.7840	0.0406	0.7311
4	1.1038	1.7769	0.0268	0.6587
5	0.8877	1.7840	0.0194	0.5900
6	0.6002	1.7902	0.0099	0.5251
7	0.5315	1.7049	0.0088	0.4640
8	0.5258	1.6498	0.0099	0.4066
9	0.5219	1.4559	0.0112	0.3531
10	0.5199	1.4436	0.0129	0.3033

No. BF: Number of basis function, Denominator: Denominator of GCV.

Table 5.26: The results of C-MARS for US data (Rep2-CV1).

$\sqrt{\bar{M}}$	No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
0.85	12	2.0082	0.85	0.2717	0.2151
0.87	12	1.874	0.87	0.2366	0.2151
0.9	12	1.6764	0.9	0.1894	0.2151
0.93	12	1.4842	0.93	0.1484	0.2151
0.95	12	1.3597	0.95	0.1246	0.2151
0.97	12	1.2388	0.97	0.1034	0.2151
1	12	1.0657	1	0.0765	0.2151
1.03	12	0.9059	1.03	0.0553	0.2151
1.05	12	0.8092	1.05	0.0441	0.2151
1.07	12	0.7232	1.07	0.0352	0.2151
1.1	12	0.6204	1.1	0.0259	0.2151
1.13	12	0.557	1.13	0.0209	0.2151
1.15	12	0.5358	1.15	0.0193	0.2151
1.17	12	0.5254	1.17	0.0186	0.2151
1.2	12	0.52	1.2	0.0182	0.2151
1.23	12	0.5197	1.2087	0.0182	0.2151
1.25	12	0.5197	1.2089	0.0182	0.2151

No. BF: Number of basis function, Denominator: Denominator of GCV.

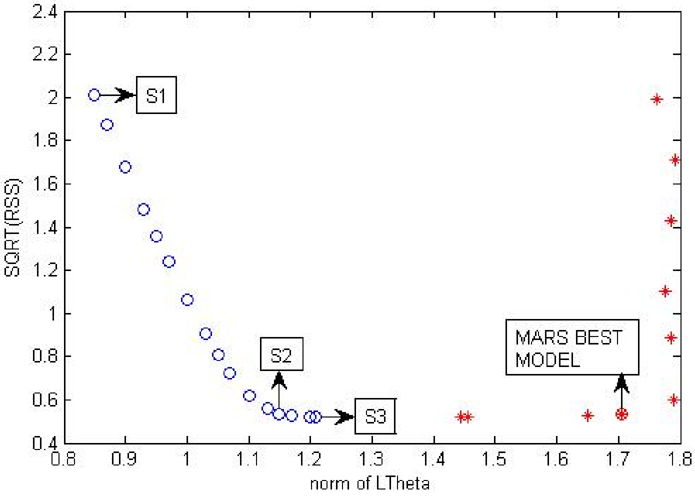

Figure 5.31: Norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for US data (Rep2-CV1).

(*: MARS solutions; o: C-MARS solutions)

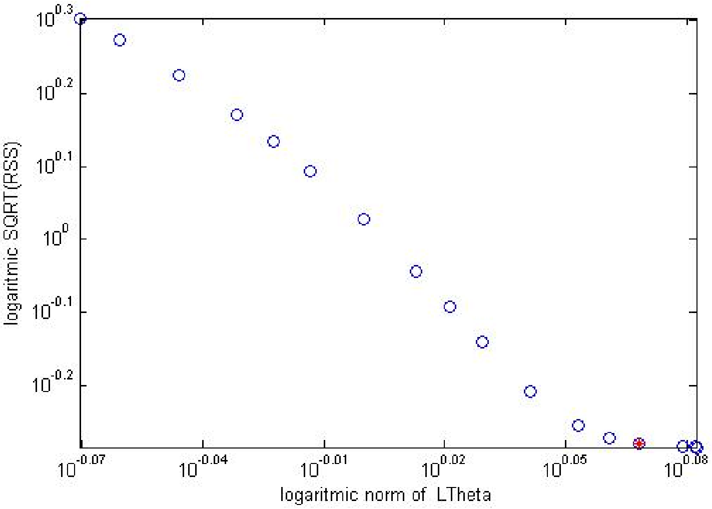

Figure 5.32: A log-log scale, the curve of norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for US data (Rep2-CV1).

REPLICATION 2 TRAIN 2

Table 5.27: The results of Salford MARS for US data (Rep2-CV2).

No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
1	2.3004	1.7700	0.0909	0.8825
2	1.8715	1.7466	0.0664	0.7991
3	1.3900	1.8073	0.0407	0.7199
4	1.1199	1.8345	0.0295	0.6449
5	0.8069	1.8255	0.0172	0.5739
6	0.6320	1.8282	0.0119	0.5071
7	0.5481	1.8093	0.0102	0.4444
8	0.4796	1.7505	0.0090	0.3859
9	0.4324	1.7536	0.0085	0.3315
10	0.3977	1.7504	0.0085	0.2812
11	0.3776	1.5446	0.0092	0.2351
12	0.3775	1.5375	0.0112	0.1931
13	0.3771	1.5390	0.0139	0.1552

No. BF: Number of basis function, Denominator: Denominator of GCV.

Table 5.28: The results of C-MARS for US data (Rep2-CV2).

$\sqrt{\bar{M}}$	No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
0.97	15	2.2504	0.97	0.8356	0.0918
1	15	2.0817	1	0.715	0.0918
1.1	15	1.5352	1.1	0.3889	0.0918
1.13	15	1.3777	1.13	0.3132	0.0918
1.15	15	1.275	1.15	0.2682	0.0918
1.17	15	1.1745	1.17	0.2276	0.0918
1.2	15	1.0284	1.2	0.1745	0.0918
1.23	15	0.8895	1.23	0.1306	0.0918
1.25	15	0.8019	1.25	0.1061	0.0918
1.27	15	0.719	1.27	0.0853	0.0918
1.3	15	0.6058	1.3	0.0606	0.0918
1.33	15	0.51	1.33	0.0429	0.0918
1.35	15	0.4585	1.35	0.0347	0.0918
1.37	15	0.4188	1.37	0.0289	0.0918
1.4	15	0.3839	1.4	0.0243	0.0918
1.5	15	0.3769	1.5	0.0234	0.0918
1.6	15	0.3769	1.6	0.0234	0.0918
1.8	15	0.3768	1.8	0.0234	0.0918
3.1	15	0.3767	3.1	0.0234	0.0918
3.2	15	0.3767	3.1999	0.0234	0.0918

No. BF: Number of basis function, Denominator: Denominator of GCV.

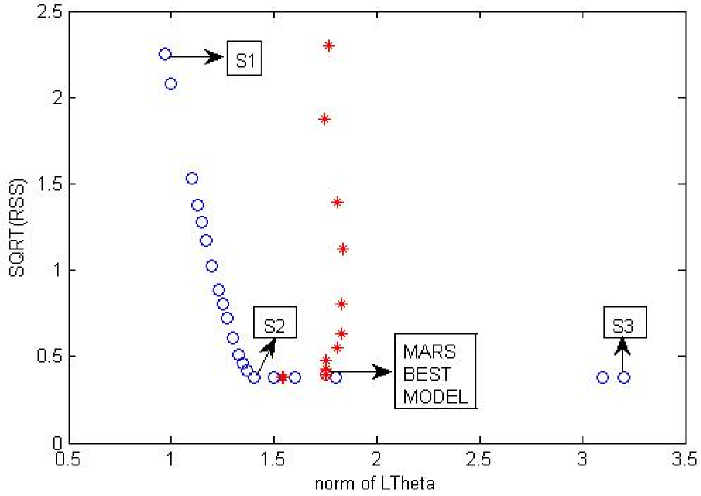

Figure 5.33: Norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for US data (Rep2-CV2).

(*: MARS solutions; o: C-MARS solutions)

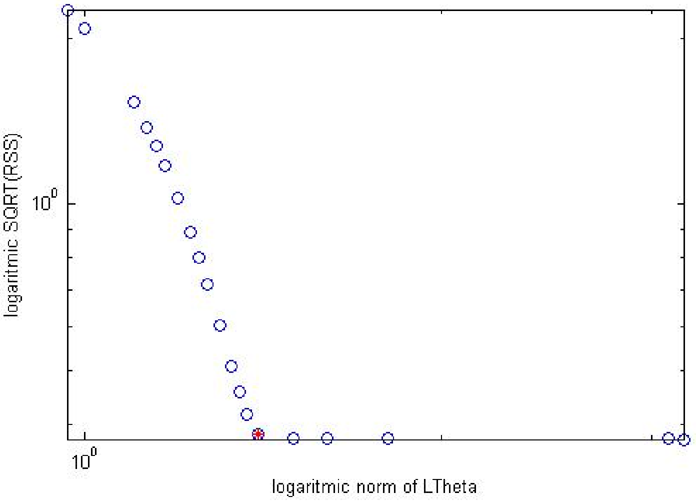

Figure 5.34: A log-log scale, the curve of norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for US data (Rep2-CV2).

REPLICATION 2 TRAIN 3

Table 5.29: The results of Salford MARS for US data (Rep2-CV3).

No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
1	2.1360	1.7741	0.0797	0.8807
2	1.7485	1.5734	0.0634	0.7422
3	1.4663	1.6116	0.0517	0.6400
4	1.1353	1.8161	0.0364	0.5453
5	0.9273	1.8415	0.0289	0.4582
6	0.7272	1.6774	0.0215	0.3787
7	0.5649	1.7038	0.0160	0.3067
8	0.4815	1.7907	0.0147	0.2424
9	0.4040	1.8164	0.0135	0.1856
10	0.3955	1.7464	0.0177	0.1363
11	0.3921	1.7327	0.0250	0.0947
12	0.3907	1.7321	0.0388	0.0606
13	0.3891	1.7205	0.0683	0.0341
14	0.3891	1.7204	0.1538	0.0151

No. BF: Number of basis function, Denominator: Denominator of GCV.

Table 5.30: The results of C-MARS for US data (Rep2-CV3).

$\sqrt{\bar{M}}$	No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
0.35	15	2.2146	0.35	19.9236	0.0038
0.4	15	2.0423	0.4	16.9454	0.0038
0.45	15	1.8931	0.45	14.5596	0.0038
0.5	15	1.7569	0.5	12.5391	0.0038
0.55	15	1.6291	0.55	10.7816	0.0038
0.6	15	1.5075	0.6	9.2323	0.0038
0.65	15	1.3908	0.65	7.8584	0.0038
0.7	15	1.2783	0.7	6.6383	0.0038
0.75	15	1.1696	0.75	5.5572	0.0038
0.8	15	1.0645	0.8	4.6037	0.0038
0.85	15	0.9632	0.85	3.7691	0.0038
0.9	15	0.8659	0.9	3.0462	0.0038
0.95	15	0.7732	0.95	2.4288	0.0038
1	15	0.6859	1	1.911	0.0038
1.05	15	0.6052	1.05	1.4878	0.0038
1.1	15	0.533	1.1	1.154	0.0038
1.15	15	0.4719	1.15	0.9047	0.0038
1.2	15	0.4253	1.2	0.7349	0.0038
1.25	15	0.3969	1.25	0.6399	0.0038

No. BF: Number of basis function, Denominator: Denominator of GCV.

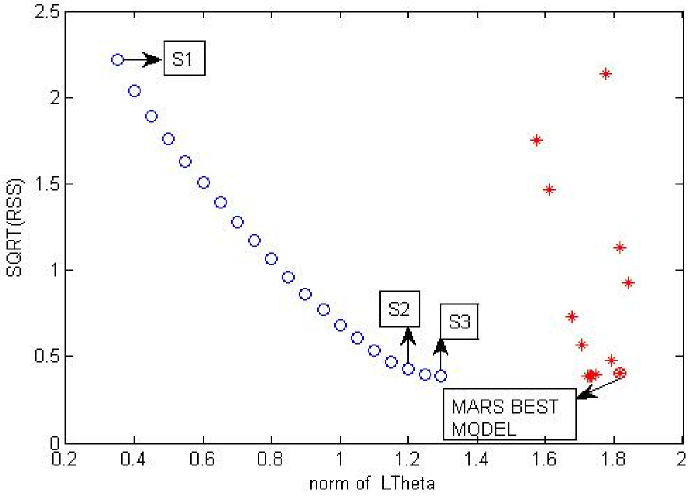

Figure 5.35: Norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for US data (Rep2-CV3).

(*: MARS solutions; o: C-MARS solutions)

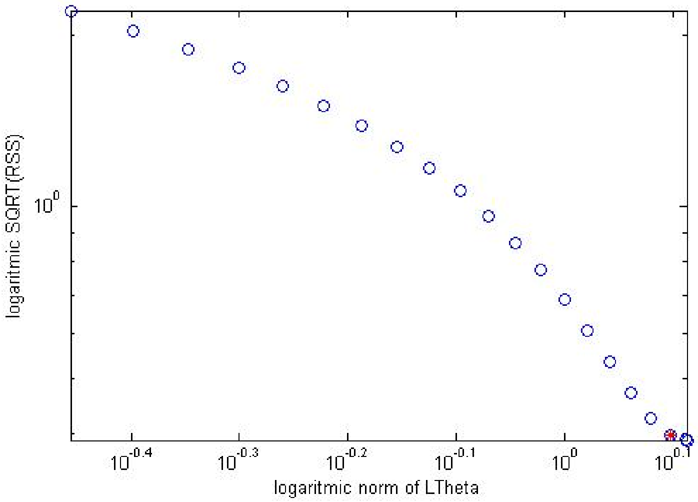

Figure 5.36: A log-log scale, the curve of norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for US data (Rep2-CV3).

REPLICATION 3 TRAIN 1

Table 5.31: The results of Salford MARS for US data (Rep3-CV1).

No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
1	2.0029	1.7748	0.0655	0.8874
2	1.7220	1.8096	0.0532	0.8074
3	1.3521	1.7760	0.0362	0.7311
4	1.0991	1.7770	0.0266	0.6587
5	0.9052	1.7977	0.0201	0.5900
6	0.5479	1.7938	0.0083	0.5251
7	0.5269	1.5252	0.0087	0.4640

No. BF: Number of basis function, Denominator: Denominator of GCV.

Table 5.32: The results of C-MARS for US data (Rep3-CV1).

$\sqrt{\bar{M}}$	No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
0.98	8	2.1256	0.98	0.161	0.4066
1	8	2.0088	1	0.1438	0.4066
1.05	8	1.7213	1.05	0.1056	0.4066
1.08	8	1.5526	1.08	0.0859	0.4066
1.1	8	1.4422	1.1	0.0741	0.4066
1.13	8	1.2806	1.13	0.0584	0.4066
1.15	8	1.1762	1.15	0.0493	0.4066
1.18	8	1.0262	1.18	0.0375	0.4066
1.2	8	0.932	1.2	0.031	0.4066
1.23	8	0.8024	1.23	0.0229	0.4066
1.25	8	0.7263	1.25	0.0188	0.4066
1.28	8	0.6324	1.28	0.0143	0.4066
1.3	8	0.5861	1.3	0.0122	0.4066
1.33	8	0.5425	1.33	0.0105	0.4066
1.35	8	0.5295	1.35	0.01	0.4066
1.38	8	0.5268	1.38	0.0099	0.4066
1.4	8	0.5268	1.4	0.0099	0.4066
1.5	8	0.5268	1.5	0.0099	0.4066
2.4	8	0.5267	2.4	0.0099	0.4066
2.5	8	0.5267	2.4999	0.0099	0.4066

No. BF: Number of basis function, Denominator: Denominator of GCV.

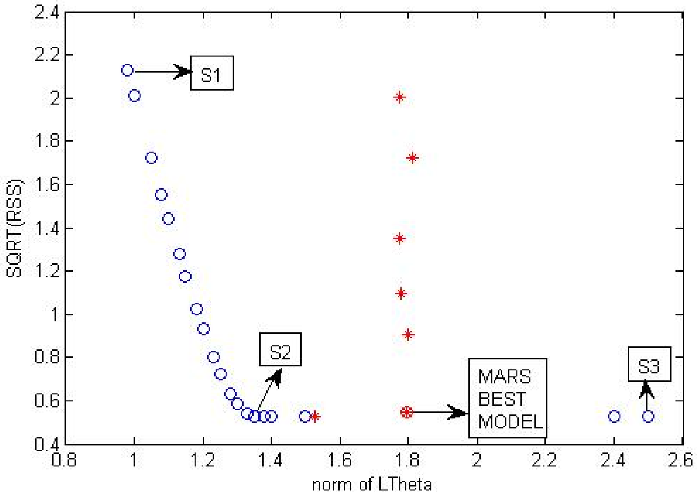

Figure 5.37: Norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for US data (Rep3-CV1).

(*: MARS solutions; o: C-MARS solutions)

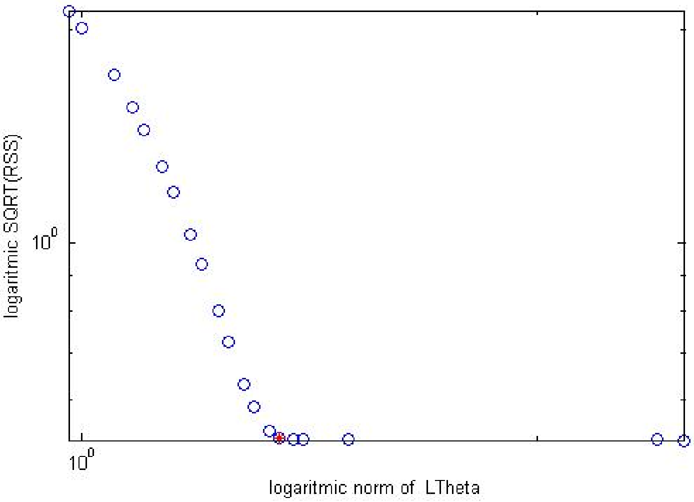

Figure 5.38: A log-log scale, the curve of norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for US data (Rep3-CV1).

REPLICATION 3 TRAIN 2

Table 5.33: The results of Salford MARS for US data (Rep3-CV2).

No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
1	2.6593	1.7597	0.1214	0.8825
2	2.1278	1.7511	0.0858	0.7991
3	1.7621	2.0812	0.0653	0.7199
4	1.3939	2.1854	0.0534	0.5512
5	1.2543	2.1740	0.0513	0.4649
6	1.0962	2.1072	0.0472	0.3859
7	0.8183	2.1532	0.0323	0.3143
8	0.6239	2.1725	0.0236	0.2500
9	0.4790	2.1391	0.0180	0.1931
10	0.4119	2.1082	0.0179	0.1435
11	0.3545	2.1079	0.0188	0.1012
12	0.2794	2.0338	0.0178	0.0663
13	0.2506	2.0345	0.0245	0.0388
14	0.2217	2.0445	0.0401	0.0186
15	0.2052	2.0678	0.1111	0.0057
16	0.1977	2.0748	2.5802	0.0002
17	0.1922	2.0493	0.2709	0.0021
18	0.1882	2.0332	0.0477	0.0112
19	0.1870	2.0333	0.0191	0.0278
20	0.1846	2.0304	0.0100	0.0517
21	0.1845	2.0276	0.0062	0.0829

No. BF: Number of basis function, Denominator: Denominator of GCV.

Table 5.34: The results of C-MARS for US data (Rep3-CV2).

$\sqrt{\bar{M}}$	No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
0.45	21	2.3151	0.45	0.9799	0.0829
0.5	21	2.1697	0.5	0.8607	0.0829
0.55	21	2.0427	0.55	0.7629	0.0829
0.6	21	1.9282	0.6	0.6797	0.0829
0.7	21	1.7239	0.7	0.5433	0.0829
0.75	21	1.6308	0.75	0.4862	0.0829
0.8	21	1.5423	0.8	0.4349	0.0829
1.1	21	1.0814	1.1	0.2138	0.0829
1.15	21	1.0135	1.15	0.1878	0.0829
1.2	21	0.9475	1.2	0.1641	0.0829
1.25	21	0.8835	1.25	0.1427	0.0829
1.3	21	0.8213	1.3	0.1233	0.0829
1.45	21	0.6447	1.45	0.076	0.0829
1.5	21	0.589	1.5	0.0634	0.0829
1.55	21	0.5349	1.55	0.0523	0.0829
1.6	21	0.4826	1.6	0.0426	0.0829
1.8	21	0.295	1.8	0.0159	0.0829
1.85	21	0.2565	1.85	0.012	0.0829
1.9	21	0.2239	1.9	0.0092	0.0829
1.95	21	0.1994	1.95	0.0073	0.0829

No. BF: Number of basis function, Denominator: Denominator of GCV.

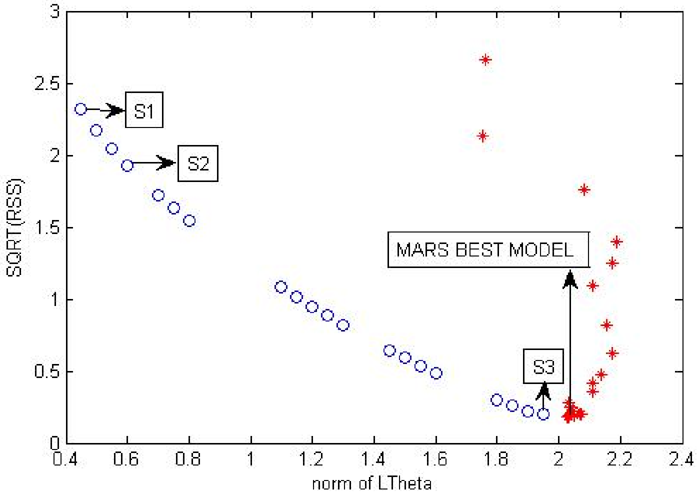

Figure 5.39: Norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for US data (Rep3-CV2).

(*: MARS solutions; o: C-MARS solutions)

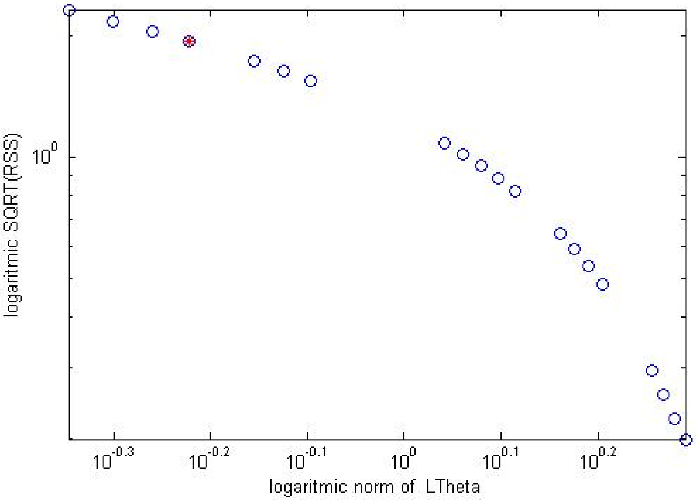

Figure 5.40: A log-log scale, the curve of norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for US data (Rep3-CV2).

REPLICATION 3 TRAIN 3

Table 5.35: The results of Salford MARS for US data (Rep3-CV3).

No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
1	2.1483	1.7543	0.0806	0.8807
2	1.8740	1.7921	0.0679	0.7962
3	1.5081	1.8337	0.0489	0.7160
4	1.1457	2.0227	0.0370	0.5453
5	0.9023	2.0271	0.0273	0.4582
6	0.6954	2.0299	0.0196	0.3787
7	0.5415	1.9643	0.0147	0.3067
8	0.4392	2.0161	0.0122	0.2424
9	0.3816	2.0062	0.0121	0.1856
10	0.3137	1.9103	0.0111	0.1363
11	0.2758	1.8780	0.0124	0.0947
12	0.2430	1.8359	0.0150	0.0606
13	0.2196	1.8484	0.0218	0.0341
14	0.1978	1.8594	0.0397	0.0151
15	0.1955	1.8602	0.1553	0.0038
16	0.1942	1.8807	Inf	0
17	0.1941	1.8808	0.1531	0.0038

No. BF: Number of basis function, Denominator: Denominator of GCV.

Table 5.36: The results of C-MARS for US data (Rep3-CV3).

$\sqrt{\bar{M}}$	No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
0.35	18	2.2074	0.35	4.9486	0.0151
0.4	18	1.9938	0.4	4.0375	0.0151
0.5	18	1.633	0.5	2.7083	0.0151
0.55	18	1.476	0.55	2.2127	0.0151
0.6	18	1.3315	0.6	1.8006	0.0151
0.65	18	1.198	0.65	1.4577	0.0151
0.7	18	1.0746	0.7	1.1727	0.0151
0.75	18	0.9603	0.75	0.9365	0.0151
0.8	18	0.8544	0.8	0.7415	0.0151
0.85	18	0.7566	0.85	0.5814	0.0151
0.9	18	0.6665	0.9	0.4512	0.0151
0.95	18	0.5843	0.95	0.3467	0.0151
1	18	0.5103	1	0.2644	0.0151
1.1	18	0.3919	1.1	0.156	0.0151
1.15	18	0.3514	1.15	0.1254	0.0151
1.2	18	0.3266	1.2	0.1083	0.0151
1.25	18	0.3191	1.25	0.1034	0.0151
1.3	18	0.319	1.3	0.1034	0.0151
1.4	18	0.319	1.4	0.1034	0.0151
2.4	18	0.3189	2.4	0.1033	0.0151
2.5	18	0.3189	2.5	0.1033	0.0151

No. BF: Number of basis function, Denominator: Denominator of GCV.

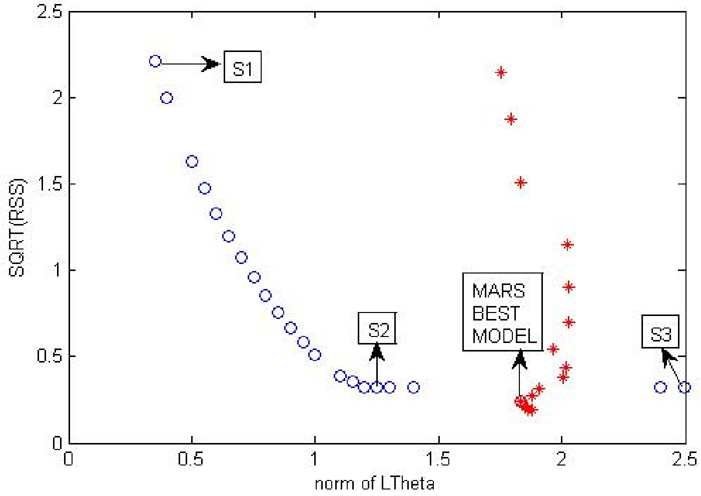

Figure 5.41: Norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for US data (Rep3-CV3).

(*: MARS solutions; o: C-MARS solutions)

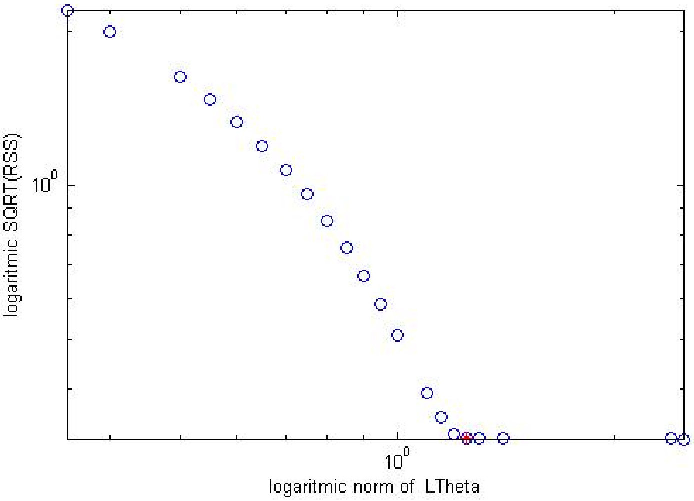

Figure 5.42: A log-log scale, the curve of norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for US data (Rep3-CV3).

APPENDIX E

Figures and Tables of Metal Casting Data

REPLICATION 1 TRAIN 1

Table 5.37: The results of Salford MARS for metal casting data (Rep1-CV1).

No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
1	7.1749	1.5306	0.9665	0.8732
2	6.6122	3.1318	0.9863	0.7267
3	6.3178	3.1152	1.0568	0.6192
4	5.8651	3.4130	1.0839	0.5203
5	5.4363	3.9396	1.1267	0.4300
6	5.3829	4.1418	1.3638	0.3483

No. BF: Number of basis function, Denominator: Denominator of GCV.

Table 5.38: The results of C-MARS for metal casting data (Rep1-CV1).

$\sqrt{\tilde{M}}$	No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
1.1	8	6.5484	1.1	3.3364	0.2107
1.2	8	6.4706	1.2	3.2577	0.2107
1.3	8	6.3961	1.3	3.1831	0.2107
1.4	8	6.3246	1.4	3.1123	0.2107
1.5	8	6.2559	1.5	3.045	0.2107
1.6	8	6.19	1.6	2.9812	0.2107
1.7	8	6.1267	1.7	2.9206	0.2107
1.8	8	6.066	1.8	2.863	0.2107
1.9	8	6.0078	1.9	2.8083	0.2107
2	8	5.952	2	2.7564	0.2107
2.1	8	5.8987	2.1	2.7072	0.2107
2.2	8	5.8478	2.2	2.6607	0.2107
2.3	8	5.7993	2.3	2.6167	0.2107
2.5	8	5.7094	2.5	2.5363	0.2107
2.6	8	5.668	2.6	2.4996	0.2107
2.7	8	5.6291	2.7	2.4654	0.2107
2.8	8	5.5925	2.8	2.4334	0.2107
2.9	8	5.5583	2.9	2.4038	0.2107
3	8	5.5265	3	2.3764	0.2107
3.1	8	5.4971	3.1	2.3512	0.2107
3.2	8	5.4702	3.2	2.3282	0.2107
3.3	8	5.4457	3.3	2.3074	0.2107
3.7	8	5.3723	3.7	2.2456	0.2107
3.8	8	5.3601	3.8	2.2354	0.2107
3.9	8	5.3504	3.9	2.2273	0.2107
4	8	5.3432	4	2.2213	0.2107

No. BF: Number of basis function, Denominator: Denominator of GCV.

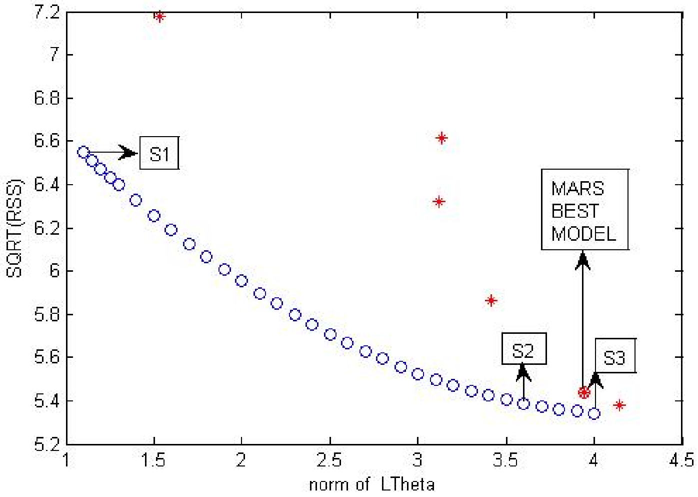

Figure 5.43: Norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for metal casting data (Rep1-CV1).

(*: MARS solutions; o: C-MARS solutions)

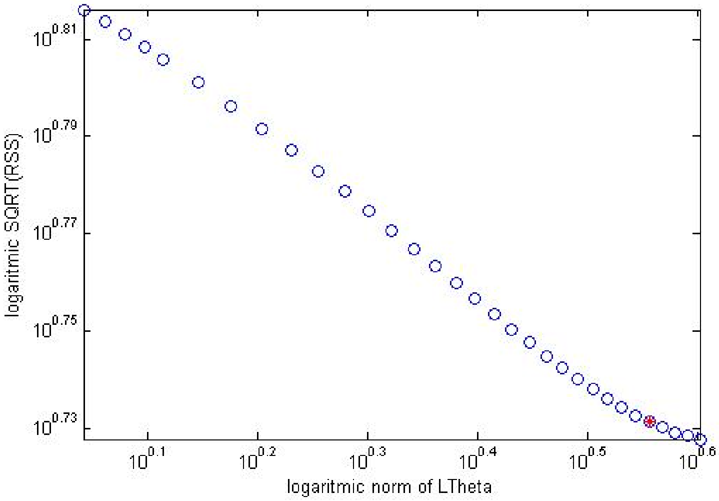

Figure 5.44: A log-log scale, the curve of norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for metal casting data (Rep1-CV1).

REPLICATION 1 TRAIN 2

Table 5.39: The results of Salford MARS for metal casting data (Rep1-CV2).

No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
1	7.0644	2.2026	0.9370	0.8732
2	6.6501	2.4088	0.9251	0.7837
3	6.4923	2.4987	0.9885	0.6990
4	6.2277	3.3534	1.0268	0.6192
5	5.9116	3.9765	1.0527	0.5442
6	5.4962	4.5832	1.0446	0.4741
7	5.4100	4.5695	1.1738	0.4088
8	5.1026	5.5400	1.2255	0.3483
9	4.8101	5.7042	1.2960	0.2927
10	4.5991	5.6061	1.4336	0.2419
11	4.4024	5.3767	1.6217	0.1959
12	4.1971	5.2198	1.8655	0.1548
13	3.9235	5.5032	2.1293	0.1185

No. BF: Number of basis function, Denominator: Denominator of GCV.

Table 5.40: The results of C-MARS for metal casting data (Rep1-CV2).

$\sqrt{\bar{M}}$	No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
0.3	25	7.029	0.3	13.3949	0.0605
0.4	25	6.8546	0.4	12.7384	0.0605
0.5	25	6.6946	0.5	12.1506	0.0605
0.7	25	6.4027	0.7	11.1139	0.0605
1.1	25	5.8927	1.1	9.4141	0.0605
1.2	25	5.7772	1.2	9.0487	0.0605
1.4	25	5.5587	1.4	8.3771	0.0605
1.6	25	5.3555	1.6	7.7758	0.0605
1.7	25	5.2592	1.7	7.4987	0.0605
1.8	25	5.1662	1.8	7.236	0.0605
1.9	25	5.0765	1.9	6.9867	0.0605
2.1	25	4.906	2.1	6.5253	0.0605
2.3	25	4.7466	2.3	6.1083	0.0605
2.4	25	4.6708	2.4	5.9147	0.0605
2.5	25	4.5975	2.5	5.7304	0.0605
2.6	25	4.5264	2.6	5.5546	0.0605
2.8	25	4.3909	2.8	5.227	0.0605
2.9	25	4.3263	2.9	5.0743	0.0605
3.1	25	4.2028	3.1	4.7889	0.0605
3.4	25	4.0314	3.4	4.4061	0.0605
3.5	25	3.9776	3.5	4.2894	0.0605
3.7	25	3.875	3.7	4.0709	0.0605
3.8	25	3.826	3.8	3.9687	0.0605
4	25	3.7327	4	3.7775	0.0605

No. BF: Number of basis function, Denominator: Denominator of GCV.

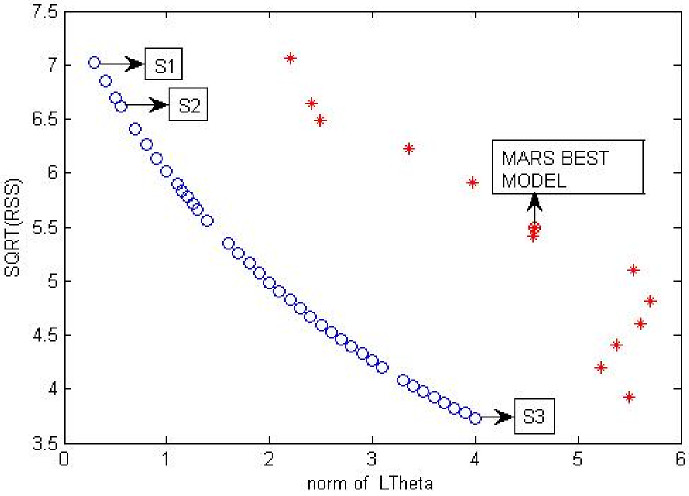

Figure 5.45: Norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for metal casting data (Rep1-CV2).

(*: MARS solutions; o: C-MARS solutions)

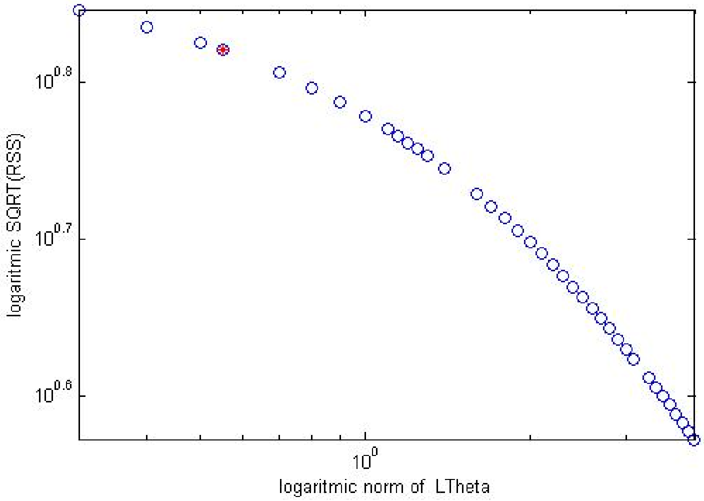

Figure 5.46: A log-log scale, the curve of norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for metal casting data (Rep1-CV2).

REPLICATION 1 TRAIN 3

Table 5.41: The results of Salford MARS for metal casting data (Rep1-CV3).

No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
1	7.4323	1.5131	1.0541	0.8452
2	6.8948	6.0459	1.0493	0.7307
3	5.8934	10.2215	0.8969	0.6246
4	5.4977	10.5587	0.9254	0.5268
5	5.0604	14.1571	0.9445	0.4373
6	4.6963	14.5926	0.9989	0.3561
7	4.4398	15.6615	1.1223	0.2833
8	4.0417	14.2006	1.2043	0.2188
9	3.8174	14.3799	1.4456	0.1626
10	3.6796	14.7833	1.9035	0.1147
11	3.5151	14.4868	2.6507	0.0752
12	3.3766	15.1821	4.1827	0.0440
13	3.2564	15.0237	8.1166	0.0211

No. BF: Number of basis function, Denominator: Denominator of GCV.

Table 5.42: The results of C-MARS for metal casting data (Rep1-CV3).

$\sqrt{\bar{M}}$	No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
0.265	20	6.8225	0.265	7.9941	0.0939
0.3	20	6.7907	0.3	7.9199	0.0939
0.4	20	6.7052	0.4	7.7216	0.0939
0.5	20	6.6252	0.5	7.5384	0.0939
0.6	20	6.5492	0.6	7.3664	0.0939
0.7	20	6.4763	0.7	7.2034	0.0939
1.15	20	6.1746	1.15	6.5478	0.0939
1.2	20	6.1431	1.2	6.4812	0.0939
1.4	20	6.0203	1.4	6.2246	0.0939
1.7	20	5.8446	1.7	5.8667	0.0939
1.8	20	5.7881	1.8	5.7539	0.0939
2.1	20	5.6243	2.1	5.4328	0.0939
2.2	20	5.5715	2.2	5.3313	0.0939
2.4	20	5.4685	2.4	5.1359	0.0939
2.5	20	5.4182	2.5	5.0419	0.0939
2.7	20	5.3201	2.7	4.861	0.0939
2.8	20	5.2722	2.8	4.7739	0.0939
3.1	20	5.1331	3.1	4.5253	0.0939
3.2	20	5.0882	3.2	4.4465	0.0939
3.4	20	5.0006	3.4	4.2947	0.0939
3.7	20	4.8743	3.7	4.0804	0.0939
3.9	20	4.7933	3.9	3.946	0.0939
4.2	20	4.6765	4.2	3.756	0.0939
4.3	20	4.6388	4.3	3.6957	0.0939
4.7	20	4.4934	4.7	3.4677	0.0939
5	20	4.3899	5	3.3098	0.0939

No. BF: Number of basis function, Denominator: Denominator of GCV.

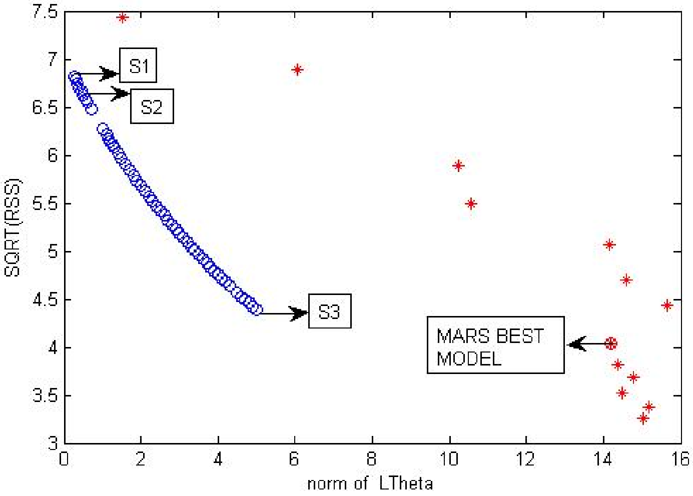

Figure 5.47: Norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for metal casting data (Rep1-CV3).

(*: MARS solutions; o: C-MARS solutions)

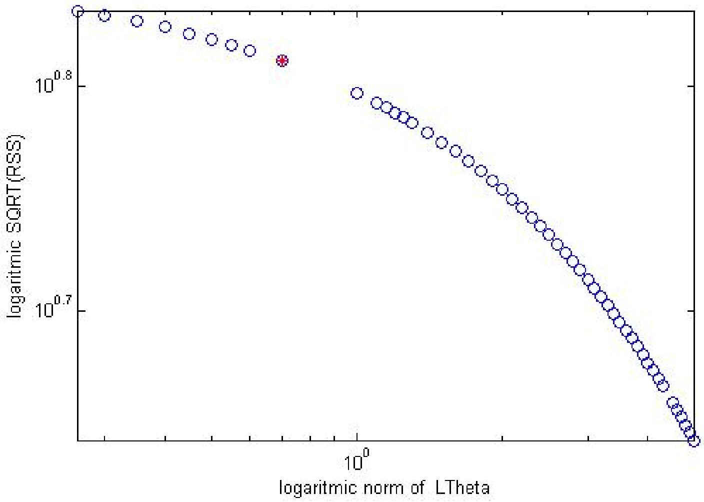

Figure 5.48: A log-log scale, the curve of norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for metal casting data (Rep1-CV3).

REPLICATION 2 TRAIN 1

Table 5.43: The results of Salford MARS for metal casting data (Rep2-CV1).

No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
1	6.5817	0.6636	0.8426	0.8428
2	6.1204	1.1380	0.8450	0.7267
3	5.8245	1.2348	0.8982	0.6192
4	5.5058	1.4275	0.9551	0.5203
5	5.0590	1.5539	0.9757	0.4300
6	4.7548	1.6627	1.0641	0.3483
7	4.4040	2.8720	1.1554	0.2752
8	4.1614	2.9592	1.3474	0.2107
9	4.0405	2.8530	1.7290	0.1548
10	3.8848	2.9586	2.3015	0.1075
11	3.8347	3.1875	3.5039	0.0688

No. BF: Number of basis function, Denominator: Denominator of GCV.

Table 5.44: The results of C-MARS for metal casting data (Rep2-CV1).

$\sqrt{\bar{M}}$	No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
0.265	19	6.7223	0.265	306.2872	0.0024
0.3	19	6.6064	0.3	295.8123	0.0024
0.45	19	6.1427	0.45	255.7402	0.0024
0.6	19	5.7308	0.6	222.5966	0.0024
0.8	19	5.2571	0.8	187.3175	0.0024
1.1	19	4.6901	1.1	149.0901	0.0024
1.3	19	4.3933	1.3	130.8208	0.0024
1.4	19	4.2654	1.4	123.3148	0.0024
1.6	19	4.0441	1.6	110.8514	0.0024
1.8	19	3.861	1.8	101.0383	0.0024
2.1	19	3.6408	2.1	89.8427	0.0024
2.2	19	3.579	2.2	86.8182	0.0024
2.4	19	3.4696	2.4	81.5906	0.0024
2.6	19	3.3766	2.6	77.2764	0.0024
3	19	3.232	3	70.7975	0.0024
3.1	19	3.2034	3.1	69.5527	0.0024
3.3	19	3.1547	3.3	67.4526	0.0024
3.4	19	3.1343	3.4	66.5845	0.0024
3.5	19	3.1165	3.5	65.8306	0.0024
3.6	19	3.1012	3.6	65.1866	0.0024
3.7	19	3.0884	3.7	64.6483	0.0024
3.8	19	3.078	3.8	64.2125	0.0024
3.9	19	3.0699	3.9	63.876	0.0024
4	19	3.0641	4	63.636	0.0024

No. BF: Number of basis function, Denominator: Denominator of GCV.

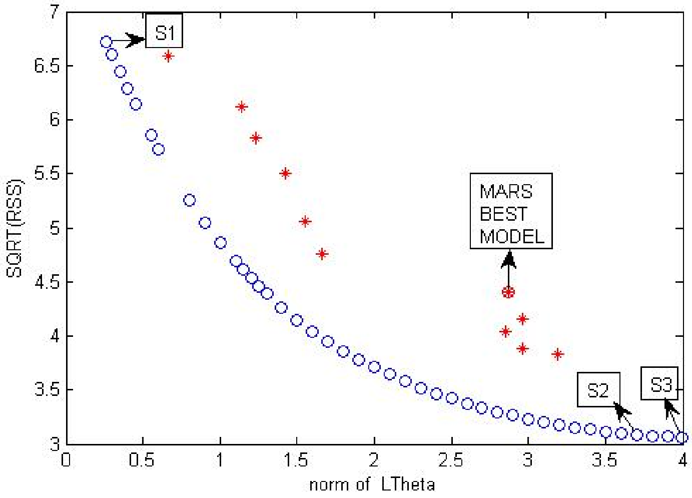

Figure 5.49: Norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for metal casting data (Rep2-CV1).

(*: MARS solutions; o: C-MARS solutions)

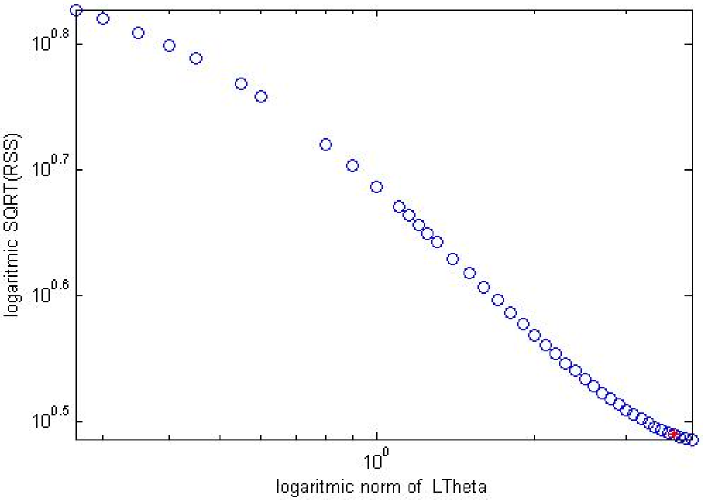

Figure 5.50: A log-log scale, the curve of norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for metal casting data (Rep2-CV1).

REPLICATION 2 TRAIN 2

Table 5.45: The results of Salford MARS for metal casting data (Rep2-CV2).

No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
1	7.2697	0.7656	0.9922	0.8732
2	6.7578	1.4278	0.9553	0.7837
3	6.6558	1.4537	1.0389	0.6990
4	6.4723	2.0849	1.1091	0.6192
5	6.2116	2.6176	1.1623	0.5442
6	6.0370	2.6932	1.2603	0.4741

No. BF: Number of basis function, Denominator: Denominator of GCV.

Table 5.46: The results of C-MARS for metal casting data (Rep2-CV2).

$\sqrt{\bar{M}}$	No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
0.265	13	7.2641	0.265	7.2988	0.1185
0.3	13	7.2072	0.3	7.185	0.1185
0.4	13	7.0532	0.4	6.8812	0.1185
0.5	13	6.9112	0.5	6.6068	0.1185
0.6	13	6.7802	0.6	6.3589	0.1185
0.7	13	6.6596	0.7	6.1347	0.1185
0.8	13	6.5484	0.8	5.9314	0.1185
1.1	13	6.2619	1.1	5.4238	0.1185
1.2	13	6.1797	1.2	5.2823	0.1185
1.3	13	6.1029	1.3	5.1519	0.1185
1.5	13	5.9641	1.5	4.9202	0.1185
1.7	13	5.8423	1.7	4.7213	0.1185
1.8	13	5.787	1.8	4.6324	0.1185
2.1	13	5.6408	2.1	4.4013	0.1185
2.3	13	5.5581	2.3	4.2731	0.1185
2.6	13	5.4536	2.6	4.1139	0.1185
2.8	13	5.396	2.8	4.0275	0.1185
2.9	13	5.3706	2.9	3.9897	0.1185
3.2	13	5.3076	3.2	3.8967	0.1185
3.3	13	5.2909	3.3	3.8721	0.1185
3.4	13	5.2761	3.4	3.8506	0.1185
3.5	13	5.2634	3.5	3.832	0.1185
3.6	13	5.2527	3.6	3.8164	0.1185
3.7	13	5.244	3.7	3.8037	0.1185
3.8	13	5.2372	3.8	3.7939	0.1185
4	13	5.2292	4	3.7824	0.1185

No. BF: Number of basis function, Denominator: Denominator of GCV.

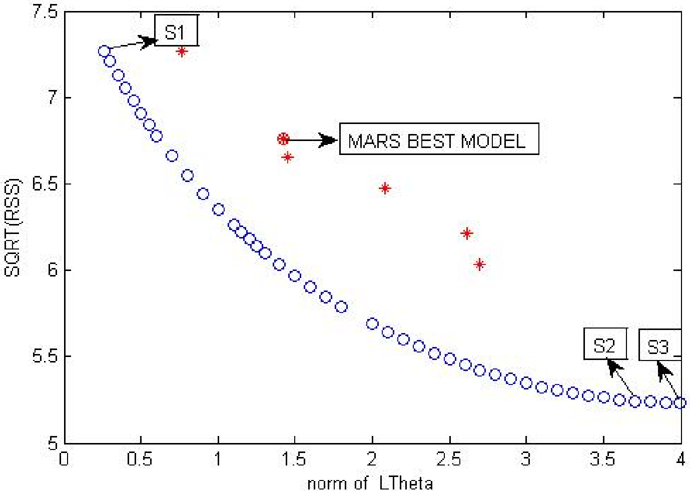

Figure 5.51: Norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for metal casting data (Rep2-CV2).

(*: MARS solutions; o: C-MARS solutions)

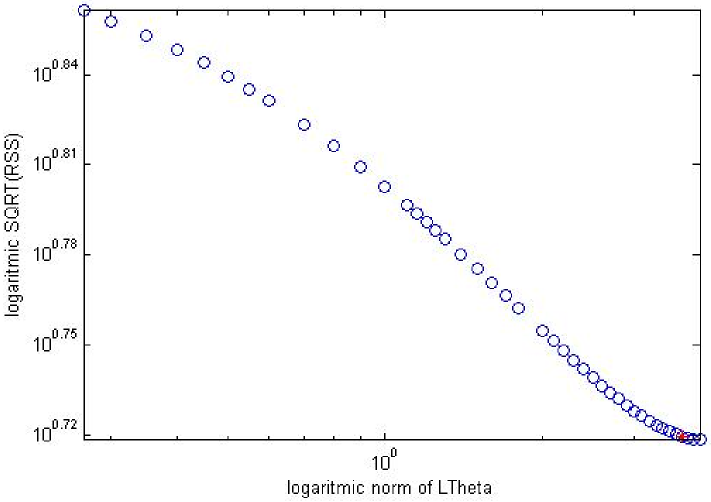

Figure 5.52: A log-log scale, the curve of norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for metal casting data (Rep2-CV2).

REPLICATION 2 TRAIN 3

Table 5.47: The results of Salford MARS for metal casting data (Rep2-CV3).

No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
1	7.0562	2.3324	0.9177	0.8751
2	6.4940	5.0266	0.9308	0.7307
3	5.8214	5.4261	0.8751	0.6246
4	5.4023	5.1742	0.8936	0.5268
5	5.0928	4.8150	0.9566	0.4373
6	4.7547	4.5057	1.0238	0.3561
7	4.4918	4.2899	1.1487	0.2833
8	4.2063	4.2933	1.3044	0.2188
9	3.7562	5.7064	1.3996	0.1626
10	3.6399	5.6697	1.8626	0.1147
11	3.5250	5.4367	2.6658	0.0752
12	3.4059	5.4215	4.2557	0.0440

No. BF: Number of basis function, Denominator: Denominator of GCV.

Table 5.48: The results of C-MARS for metal casting data (Rep2-CV3).

$\sqrt{\bar{M}}$	No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
0.265	18	6.6042	0.265	22.3487	0.0315
0.3	18	6.4805	0.3	21.5192	0.0315
0.35	18	6.3116	0.35	20.4118	0.0315
0.4	18	6.152	0.4	19.3925	0.0315
0.5	18	5.8611	0.5	17.6019	0.0315
0.6	18	5.6073	0.6	16.1105	0.0315
0.8	18	5.1992	0.8	13.851	0.0315
0.9	18	5.036	0.9	12.995	0.0315
1.1	18	4.769	1.1	11.6538	0.0315
1.2	18	4.6586	1.2	11.1201	0.0315
1.6	18	4.3192	1.6	9.5589	0.0315
1.8	18	4.1915	1.8	9.0023	0.0315
1.9	18	4.1349	1.9	8.7605	0.0315
2.1	18	4.0326	2.1	8.3326	0.0315
2.2	18	3.9861	2.2	8.1416	0.0315
2.3	18	3.9423	2.3	7.9635	0.0315
2.6	18	3.8239	2.6	7.4925	0.0315
2.8	18	3.7543	2.8	7.2221	0.0315
2.9	18	3.7219	2.9	7.098	0.0315
3.1	18	3.6615	3.1	6.8694	0.0315
3.4	18	3.581	3.4	6.5707	0.0315
3.5	18	3.5567	3.5	6.4818	0.0315
3.7	18	3.5117	3.7	6.3189	0.0315
3.8	18	3.491	3.8	6.2446	0.0315
4	18	3.453	4	6.1094	0.0315

No. BF: Number of basis function, Denominator: Denominator of GCV.

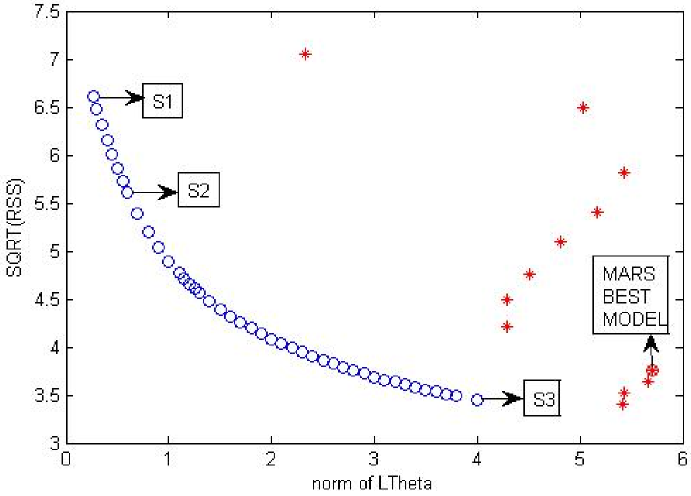

Figure 5.53: Norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for metal casting data (Rep2-CV3).

(*: MARS solutions; o: C-MARS solutions)

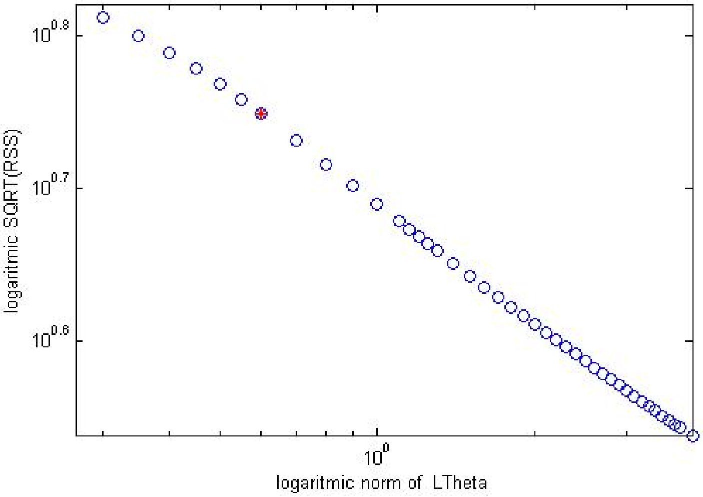

Figure 5.54: A log-log scale, the curve of norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for metal casting data (Rep2-CV3).

REPLICATION 3 TRAIN 1

Table 5.49: The results of Salford MARS for metal casting data (Rep3-CV1).

No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
1	7.0223	2.2872	0.9089	0.8751
2	6.5239	8.1793	0.9394	0.7307
3	6.2921	16.3229	1.0223	0.6246
4	5.1404	77.7190	0.8090	0.5268
5	4.7758	77.9845	0.8412	0.4373
6	4.4926	75.8218	0.9141	0.3561
7	4.2336	79.0825	1.0204	0.2833
8	3.9818	78.6799	1.1688	0.2188
9	3.7713	78.6466	1.4109	0.1626
10	3.5925	82.1292	1.8144	0.1147
11	3.4485	80.4764	2.5512	0.0752

No. BF: Number of basis function, Denominator: Denominator of GCV.

Table 5.50: The results of C-MARS for metal casting data (Rep3-CV1).

$\sqrt{\bar{M}}$	No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
0.265	15	6.6538	0.265	10.7222	0.0666
0.3	15	6.5774	0.3	10.4775	0.0666
0.4	15	6.372	0.4	9.8335	0.0666
0.55	15	6.0928	0.55	8.9905	0.0666
0.6	15	6.0065	0.6	8.7376	0.0666
1	15	5.4232	1	7.1231	0.0666
1.25	15	5.146	1.25	6.4134	0.0666
1.4	15	5.008	1.4	6.0741	0.0666
1.5	15	4.9268	1.5	5.8788	0.0666
1.7	15	4.7878	1.7	5.5517	0.0666
1.8	15	4.7289	1.8	5.416	0.0666
1.9	15	4.6764	1.9	5.2963	0.0666
2.1	15	4.5882	2.1	5.0985	0.0666
2.2	15	4.5516	2.2	5.0174	0.0666
2.3	15	4.5193	2.3	4.9465	0.0666
2.4	15	4.491	2.4	4.8846	0.0666
2.5	15	4.4662	2.5	4.8308	0.0666
2.9	15	4.3957	2.9	4.6795	0.0666
3.2	15	4.3644	3.2	4.6132	0.0666
3.3	15	4.3565	3.3	4.5966	0.0666
3.4	15	4.3495	3.4	4.5817	0.0666
3.5	15	4.3431	3.5	4.5682	0.0666
3.7	15	4.3318	3.7	4.5444	0.0666
3.8	15	4.3266	3.8	4.5337	0.0666
3.9	15	4.3218	3.9	4.5235	0.0666
4	15	4.3171	4	4.5138	0.0666

Nu. BF: Number of basis function, Denominator: Denominator of GCV.

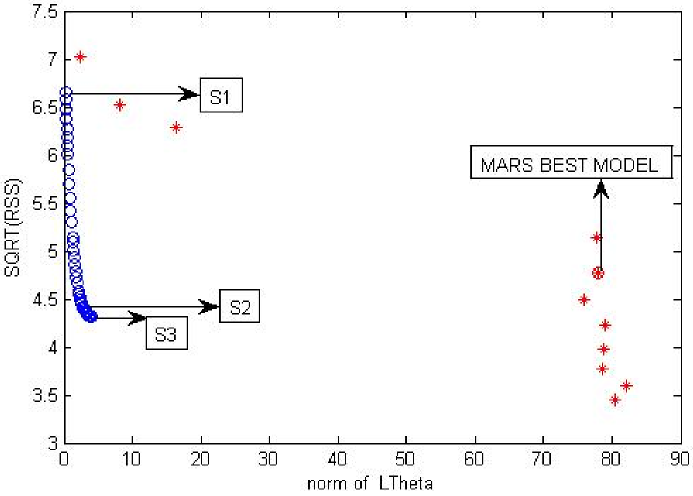

Figure 5.55: Norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for metal casting data (Rep3-CV1).

(*: MARS solutions; o: C-MARS solutions)

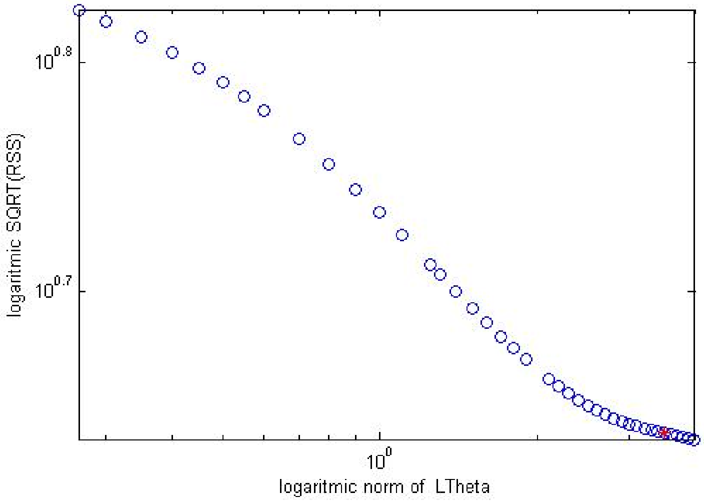

Figure 5.56: A log-log scale, the curve of norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for metal casting data (Rep3-CV1).

REPLICATION 3 TRAIN 2

Table 5.51: The results of Salford MARS for metal casting data (Rep3-CV2).

No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
1	7.2050	2.0840	0.9568	0.8751
2	6.4565	6.0437	0.9201	0.7307
3	6.0275	6.1080	0.9381	0.6246
4	5.9827	6.0987	1.0959	0.5268
5	5.4422	6.8918	1.0924	0.4373
6	5.2494	6.8850	1.2480	0.3561
7	4.8478	7.2842	1.3380	0.2833
8	4.4765	7.5388	1.4773	0.2188
9	3.9808	7.8957	1.5720	0.1626
10	3.6757	8.1823	1.8995	0.1147
11	3.3712	8.0593	2.4382	0.0752
12	3.2118	8.1696	3.7844	0.0440
13	2.9580	7.8355	6.6972	0.0211
14	2.8363	8.1693	19.9505	0.0065
15	2.7010	8.2864	452.3170	0.0003

No. BF: Number of basis function, Denominator: Denominator of GCV.

Table 5.52: The results of C-MARS for metal casting data (Rep3-CV2).

$\sqrt{\bar{M}}$	No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
0.265	22	6.8545	0.265	3.996	0.1896
0.35	22	6.7063	0.35	3.825	0.1896
0.4	22	6.6289	0.4	3.7372	0.1896
0.5	22	6.4888	0.5	3.581	0.1896
0.55	22	6.4244	0.55	3.5102	0.1896
0.8	22	6.1375	0.8	3.2037	0.1896
1	22	5.9341	1	2.9948	0.1896
1.15	22	5.7908	1.15	2.852	0.1896
1.3	22	5.6535	1.3	2.7183	0.1896
1.4	22	5.5646	1.4	2.6335	0.1896
1.7	22	5.3086	1.7	2.3967	0.1896
1.8	22	5.2262	1.8	2.323	0.1896
1.9	22	5.1452	1.9	2.2515	0.1896
2	22	5.0655	2	2.1822	0.1896
2.2	22	4.9095	2.2	2.0499	0.1896
2.3	22	4.8331	2.3	1.9866	0.1896
2.7	22	4.5381	2.7	1.7515	0.1896
2.8	22	4.4669	2.8	1.697	0.1896
2.9	22	4.3966	2.9	1.644	0.1896
3.1	22	4.2589	3.1	1.5426	0.1896
3.2	22	4.1915	3.2	1.4942	0.1896
3.4	22	4.0595	3.4	1.4016	0.1896
3.5	22	3.9949	3.5	1.3573	0.1896
3.7	22	3.8687	3.7	1.2729	0.1896
3.8	22	3.8071	3.8	1.2327	0.1896
4	22	3.6868	4	1.156	0.1896

No. BF: Number of basis function, Denominator: Denominator of GCV.

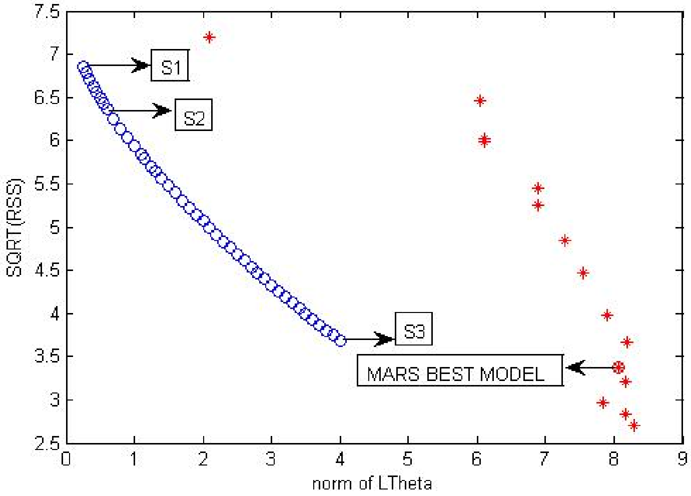

Figure 5.57: Norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for metal casting data (Rep3-CV2).

(*: MARS solutions; o: C-MARS solutions)

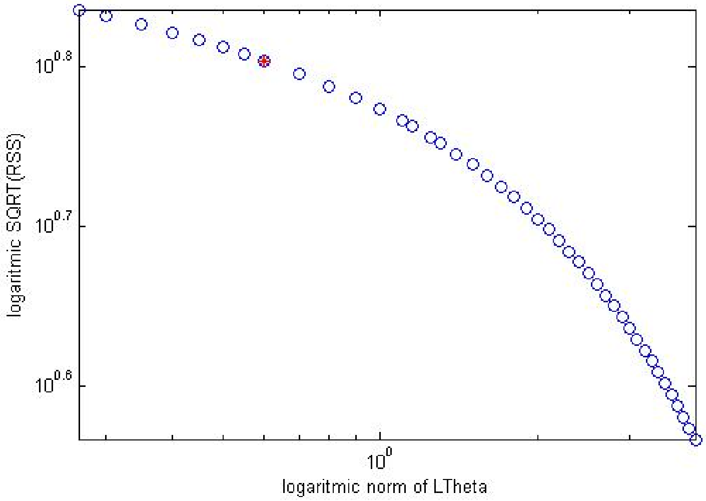

Figure 5.58: A log-log scale, the curve of norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for metal casting data (Rep3-CV2).

REPLICATION 3 TRAIN 3

Table 5.53: The results of Salford MARS for metal casting data (Rep3-CV3).

No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
1	7.4349	0.7789	1.0188	0.8751
2	7.0438	1.0514	1.0169	0.7869
3	6.6701	1.7130	1.0201	0.7034
4	6.1898	2.1061	0.9894	0.6246
5	6.0581	2.0746	1.0754	0.5505
6	5.7934	2.3300	1.1255	0.4810
7	5.5325	2.6714	1.1861	0.4162
8	5.3211	2.8815	1.2823	0.3561
9	5.0425	3.0039	1.3637	0.3007
10	4.7659	3.4233	1.4654	0.2500
11	4.5774	3.4558	1.6570	0.2040
12	4.4411	3.6363	1.9565	0.1626

No. BF: Number of basis function, Denominator: Denominator of GCV.

Table 5.54: The results of C-MARS for metal casting data (Rep3-CV3).

$\sqrt{\bar{M}}$	No. BF	SQRT(RSS)	norm of $\boldsymbol{L\theta}$	GCV	Denominator
0.7	25	6.3766	0.7	12.8621	0.051
0.8	25	6.2119	0.8	12.2064	0.051
1	25	5.903	1	11.0225	0.051
1.15	25	5.687	1.15	10.2307	0.051
1.2	25	5.6177	1.2	9.9829	0.051
1.3	25	5.483	1.3	9.5098	0.051
1.4	25	5.3531	1.4	9.0647	0.051
1.6	25	5.1072	1.6	8.251	0.051
1.7	25	4.9909	1.7	7.8793	0.051
1.9	25	4.7708	1.9	7.1997	0.051
2	25	4.6669	2	6.8896	0.051
2.1	25	4.5671	2.1	6.5981	0.051
2.2	25	4.4714	2.2	6.3244	0.051
2.4	25	4.2921	2.4	5.8275	0.051
2.6	25	4.1293	2.6	5.3937	0.051
2.9	25	3.9166	2.9	4.8523	0.051
3	25	3.8543	3	4.6992	0.051
3.1	25	3.7964	3.1	4.5591	0.051
3.2	25	3.743	3.2	4.4317	0.051
3.3	25	3.6941	3.3	4.3168	0.051
3.4	25	3.6498	3.4	4.2138	0.051
3.6	25	3.5752	3.6	4.0432	0.051
3.7	25	3.5449	3.7	3.975	0.051
3.9	25	3.4984	3.9	3.8716	0.051
4	25	3.4823	4	3.836	0.051

No. BF: Number of basis function, Denominator: Denominator of GCV.

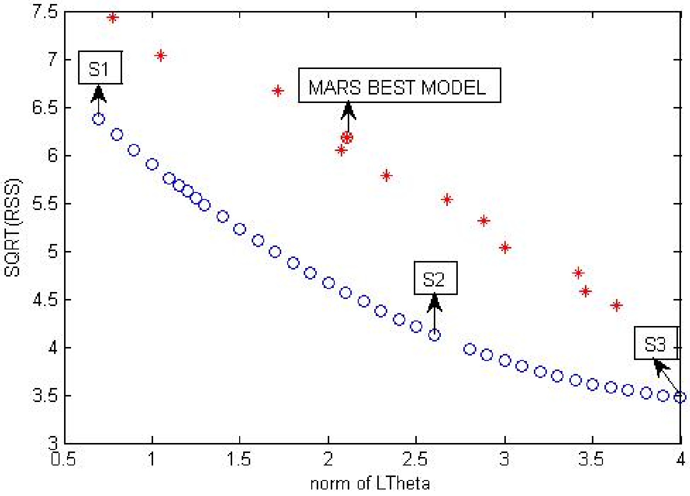

Figure 5.59: Norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for metal casting data (Rep3-CV3).

(*: MARS solutions; o: C-MARS solutions)

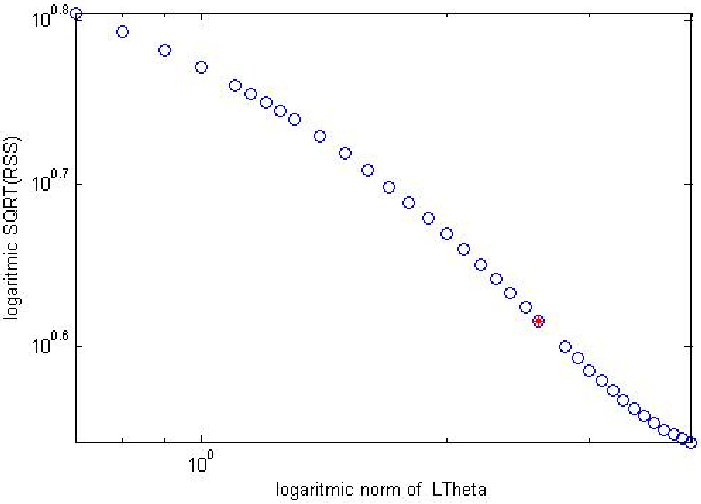

Figure 5.60: A log-log scale, the curve of norm of $\boldsymbol{L\theta}$ vs. SQRT(RSS) for metal casting data (Rep3-CV3).