School of Mathematical Sciences

THE AUSTRALIAN
NATIONAL UNIVERSITY

Canberra, Australia

Adaptive Regression and Model Selection in Data Mining Problems

by

Sergey Bakin

A thesis submitted for the degree of Doctor of Philosophly
of The Australian National University.

May 1999

Statement

I declare that this thesis is my own original work and all sources have been acknowledged.

Sergey Bakin

Acknowledgments

I would like to express my profound gratitude to my supervisor Professor Michael R. Osborne as well as to my advisor Doctor Markus Hegland for the time and efforts they have put into directing and supervising my PhD research, for the fruitful discussions we have had, for their patience in dealing with my written English, and for their invaluable assistance in helping me to get an academic appointment.

I would like to thank Doctor Steven Roberts for discussing various research problems with me as well as for writing a reference letter to support my job application. I would also like to thank the CRC for Advanced Computational Systems (ACSys) for the generous financial and technical assistance.

I must acknowledge the support given to me by my mother whose love and strength have enabled me to concentrate on my studies.

Special thanks to Jacki Wicks who has made the last month of my stay at ANU pleasant and unforgettable, as well as to Derek Holtby who has greatly influenced the formation of my spiritual outlook and has set an example of living as a Christian for me. I am also deeply grateful to James Rice and Csaba Schneider for their support and friendship.

I would like to thank all the students and staff of the School of Mathematical Sciences of the Australian National University, and members of the Advanced Computation Group in particular, for the friendly and stimulating working and social environment.

Finally, I owe many thanks to the anonymous examiners whose valuable comments helped me improve the quality of this thesis.

Abstract

Data Mining is a new and rapidly evolving area which deals with problems related to extracting structure from massive commercial and scientific data sets. Regression analysis is one of the major Data Mining techniques. The data sets encountered in the Data Mining area are often characterized by a large number of attributes (variables) as well as data records. This imposes two major requirements on the regression analysis tools used in Data Mining: first, in order to produce accurate and parsimonious models exhibiting the most important features of the problem in hand, they should be able to perform model selection adaptively and, second, the cost of running such tools has to be reasonably low. Most of the modern regression tools fail to meet the above requirements. This thesis is intended to contribute to the improvement of the existing methodologies as well as to propose new approaches.

We focus on two regression estimation techniques. The first one, called Probing Least Absolute Squares Modelling (PLASM), is a generalization of the Least Absolute Shrinkage and Selection Operator (LASSO) by R. Tibshirani which minimizes the residual sum of squares subject to the l_1-norm of the regression coefficients being less than a constant. LASSO has been shown to enjoy stability of the ridge regression coupled with the ability to carry out model selection. In our approach called PLASM, we replace the constraint employed in LASSO with a different constraint. PLASM allows for an arbitrary grouping of basis functions in a model and includes LASSO as a special case. The implication of using the new constraint is that PLASM is able to perform model selection in terms of groups of basis functions. This turns out to be very useful in a number of data analytic problems. For example, as far as additive modelling is concerned, the dimensionality of the PLASM minimization problem is much less than that of LASSO and is independent (at least explicitly) of the number of datapoints which makes it suitable for use in the Data Mining context.

The second tool we consider in this thesis is the Multivariate Adaptive Regression Splines (MARS) developed by J. Friedman. In our version of MARS called BMARS, we use B-splines instead of truncated power basis functions. The fact that B-splines have the compact support property allows us to introduce a new strategy whereby at any moment

the algorithm builds a model using B-splines of a certain scale only and it switches over to splines of smaller scale after the fitting ability of the current splines has been exhausted. Also, we discuss a parallel version of BMARS as well as an application of the algorithm to processing of a large commercial data set. The results of the numerical experiments demonstrate that, while being considerably more efficient, BMARS is able to produce models competitive with those of the original MARS.

Contents

List of Figures

List of Tables

Chapter 1
Introduction

Statistical regression models represent a convenient way to understand and summarize the structure of various kinds of data. However, each model is to achieve two, nearly always conflicting goals: on the one hand, it should follow trends in a data set closely and, on the other hand, it is often required to be simple. Not only does a simple model enable a researcher to gain a better insight into the data, but also (if carefully built) it is likely to command a greater predictive ability. The need for efficient regression modelling techniques became especially important with the appearance a few years ago of a new multidisciplinary field called Data Mining. Data Mining deals with extraction of useful information from massive scientific and commercial data sets and includes a large scale regression analysis as one of its components [3], [17]. The problems arising in Data Mining are characterized by a large number of data points as well as predictor variables and, therefore, the availability of scalable, adaptive nonparametric procedures is vital for the solution of Data Mining problems.

Fuelled by the increase in computing powers, the field of nonparametric regression analysis has seen an enormous growth in the past two decades. After providing a formal formulation of the problem of the regression estimation, we will give a brief overview of the most recent and profound achievements in the area.

1.1 Problem Formulation

Given data $\mathcal{D} = \{(\mathbf{x}_n, y_n),\ n = 1, \ldots, N\}$, where $\mathbf{x} = (x_1, \ldots, x_d)$ is a vector of *predictor variables* (independent variables) and y is a *response value* (dependent variable), we assume that the predictors and response are related in the following way:

$$y_n = f(\mathbf{x}_n) + \epsilon_n, \tag{1.1}$$

where $f(\cdot)$ is some smooth *regression function* which is to be estimated; $\{\epsilon_n\}$ are independent and identically distributed zero mean *noise variables* that have to be included due to, for instance, experimental errors. As we pointed out before, the model $\hat{f}(\mathbf{x})$ has to be an accurate approximation of the regression function $f(\mathbf{x})$ and, at the same time, it should be easy to interpret. For example, it may be required to depend only on those predictor variables (and their interactions) which exhibit the strongest effects. The traditional techniques such as linear multivariate (parametric) regression are likely to be inappropriate in this situation and, instead, one has to resort to the so-called *adaptive, nonparametric* procedures that do not rely on the models whose structure is prespecified up to several parameters to be estimated but rather select the most appropriate one based on the data [20].

In order to be able to compare models produced by various regression techniques, one has to define a measure of the distance between a regression function $f(\mathbf{x})$ and a model $\hat{f}(\mathbf{x})$. One of the measures that evaluates the behaviour of the model $\hat{f}(\mathbf{x})$ at a fixed point $\mathbf{x}$ is the *Mean Squared Error* (MSE)

$$\mathrm{MSE}(\mathbf{x}) = E[\hat{f}(\mathbf{x}) - f(\mathbf{x})^2];$$

where the expectation is taken over the joint distribution of the observations $(\mathbf{x}_n, y_n)$, $n = 1, \ldots, N$. To evaluate the global behaviour of the model, one can use the *Integrated Mean Squared Error* (IMSE)

$$\mathrm{IMSE} = \int_{\mathbf{x}} \mathrm{MSE}(\mathbf{x}) w(\mathbf{x}) d\mathbf{x}, \tag{1.2}$$

where the weight function $w(\mathbf{x})$ is often taken to be identical either to one or the marginal density of $\mathbf{x}$. A related quantity that is used in this work's simulation studies is called *Scaled Mean Squared Error* (SMSE):

$$\text{SMSE} = \frac{\text{IMSE}}{\text{Var}(f)}, \tag{1.3}$$

where $\text{Var}(f) = \int [f(\mathbf{x}) - \bar{f}]^2 w(\mathbf{x}) d\mathbf{x}$ and $\bar{f} = \int f(\mathbf{x}) w(\mathbf{x}) d\mathbf{x}$. Another measure closely related to IMSE is the *Prediction Error* of $\hat{f}(\mathbf{x})$ (PE) defined as

$$\text{PE} = E[y - \hat{f}(\mathbf{x})^2]. \tag{1.4}$$

Here the expectation is taken over the joint distribution of the data points $(\mathbf{x}_n, y_n)$, $n = 1, \ldots, N$ as well as the future independent observation $(\mathbf{x}, y)$. Assuming that $w(\mathbf{x})$ is equal to the marginal density of $\mathbf{x}$ and $\text{Var}(y|\mathbf{x}) = \sigma^2$ is independent of $\mathbf{x}$, the connection between IMSE and PE can be expressed as

$$\text{PE} = \text{MSE} + \sigma^2.$$

Unfortunately, the joint probability distributions used in the above definitions as well as the true regression function $f(\mathbf{x})$ are often unknown. Therefore, one has to consider techniques for computing approximations of the above measures. Below is the list of some of the popular approaches to approximate evaluation of the Prediction Error:

- *Cross-Validation* score (CV) [54] is defined as

$$\text{CV} = \frac{1}{N} \sum_{n=1}^{N} [y_n - \hat{f}_{-n}(\mathbf{x}_n)]^2, \tag{1.5}$$

  where $\hat{f}_{-n}(\mathbf{x})$ is a model of the regression function $f(\mathbf{x})$ estimated in the same way as $\hat{f}(\mathbf{x})$ using all but the n-th data point.

- *L-fold Cross-Validation* (CV_L) is a somewhat less expensive way to estimate PE compared to CV and it is based on splitting of the data set $\mathcal{D}$ into L parts $\mathcal{D}_1, \ldots, \mathcal{D}_L$ having approximately the same size

$$\mathrm{CV}_L = \frac{1}{N}\sum_{l=1}^{L}\sum_{n_l \in I_{\mathcal{D}_l}} [y_{n_l} - \hat{f}_{-\mathcal{D}_l}(\mathbf{x}_{n_l})]^2, \tag{1.6}$$

where $I_{\mathcal{D}_l}$ is a collection of indexes referencing data points in $\mathcal{D}_l$ and $\hat{f}_{-\mathcal{D}_l}(\mathbf{x})$ is a model estimated using $\mathcal{D}\backslash\mathcal{D}_l$ data points as well as the same regression estimation method as used to estimate $\hat{f}(\mathbf{x})$.

- *Generalized Cross-Validation* score (GCV) [13] can be computed as follows

$$\mathrm{GCV} = \frac{1}{N}\frac{\sum_{n=1}^{N}[y_n - \hat{f}(\mathbf{x}_n)]^2}{[1 - \mathrm{df}/N]^2} = \frac{1}{N}\frac{\mathrm{RSS}}{[1 - \mathrm{df}/N]^2}, \tag{1.7}$$

where df is the number of degrees of freedom used to estimate the model $\hat{f}(\mathbf{x})$. The definition of df depends on the context in which GCV is used.

It is worth noting that GCV is the least computationally expensive of the methods listed above and it is used quite extensively in this thesis. There are other approaches to estimating Prediction Errors of regression models such as *Jackknife* [40], *Bootstrap* [16] etc though they will not be considered here.

1.2 Modern Regression Modelling Procedures

This section provides a brief outline of the most recent methodologies in the area of regression analysis and highlights their advantages and disadvantages.

The Smoothing Interaction Splines algorithm produces models of the form of an expansion in low dimensional functions [7], [59]

$$\hat{f}(\mathbf{x}) = \sum_{j=1}^{M} \hat{f}_j(\mathbf{v_j}).$$

where $\hat{f}_j(\cdot)$ are some smooth functions to be determined and $\mathbf{v_j}$, $j = 1, ..., M$ represent small preselected subsets of the explanatory variables $x_1, ..., x_d$. Having selected the

subsets of variables $\mathbf{v_j}$, one obtains the corresponding function estimates $\{\hat{f}_j(\mathbf{v_j})\}_1^M$ via minimization of the following functional

$$J(f_1, \ldots, f_M) = \sum_{n=1}^{N} [y_n - \sum_{j=1}^{M} f_j(\mathbf{v_{jn}})]^2 + \sum_{j=1}^{M} \lambda_j P(f_j).$$

where $P(f_j)$'s are roughness penalty terms whose values increase with the increasing roughness of the functions f_j, $j = 1, ..., M$. The minimization of $J(f_1, \ldots, f_M)$ is performed over all f_j for which it is defined. The parameters λ_j's regulate the tradeoff between the roughness of f_j's and the level of deviations of the data points from the regression surface. For example, the possible choice for the functional P can be defined as follows

$$P(f_j) = \sum_{k=1}^{d_j} \sum_{l=1}^{d_j} \int \left| \frac{\partial^2 f}{\partial x_k \partial x_l} \right|^2 \mathrm{d}x,$$

where d_j is the dimensionality of the argument of the function f_j. This choice is appropriate for $d_j \leq 3$ leading to thin-plate splines. For $d_j > 3$, the general thin-plate spline penalty has a more complex form involving derivatives of higher order than two [59]. Despite the unquestionable practical value of the approach (see, for instance, [36], [59]), it has a number of serious limitations. First, it is not clear how to perform an efficient selection of the appropriate subsets $\mathbf{v_j}$ of the predictor variables. Second, there are M parameters λ_j, $j = 1, ..., M$ present in the functional J. Determination of the best values for those parameters involves multivariate optimization which is quite an involved and often computationally expensive exercise.

Another interesting procedure based on the same principles as the previous one is often referred to as Generalized Additive Models (GAM) [28],[51],[52]. Basically, it is a smooth extension of the ideas of Generalized Linear Models (GLM) [15], [37] and, therefore, is able to deal with more general regression estimation problems compared to that set out before: one no longer expects responses to have the same variance. In particular, it is assumed that the response values have distribution density from the *exponential* family:

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

where θ is the *natural* parameter and ϕ is the *scale* parameter. Also, it is assumed that the expectation of y, denoted by μ, is related to the set of covariates $x_1, ..., x_d$ by $g(\mu) = \eta$, where $\eta = \sum_{i=1}^{d} f_i(x_i)$. Here $f_i(x_i),\ i = 1, ..., d$ are some smooth univariate functions. The function $g(\cdot)$ is called *link* function because it links the systematic component of the model η with the random component represented by μ. The estimates of the univariate components $\hat{f}_i(x_i),\ i = 1, ..., d$ are normally found through maximization of the log-likelihood function

$$l(\mathbf{y}, \theta) = \sum_{n=1}^{N} \left[\frac{(y_n \theta_n - b(\theta_n)}{a(\phi)} + c(y_n, \phi) \right]$$

with respect to univariate functions $f_i(x_i),\ i = 1, ..., d$ subject to certain smoothness constraints determined by the algorithm used to estimate them. The minimization of the log-likelihood is carried out based on the *Local Scoring Algorithm* [28]:

$f_1^0(x_1) \leftarrow 0, \ldots, f_d^0(x_d) \leftarrow 0$
$m \leftarrow 0$
repeat
  $m \leftarrow m + 1$
  $\eta_n^{m-1} \leftarrow \sum_{j=1}^{d} f_j^{m-1}(x_{jn}),\ n = 1, ..., N$
  $\mu_n^{m-1} \leftarrow g^{-1}(\eta_n^{m-1}),\ n = 1, ..., N$
  $z_n^m \leftarrow \eta_n^{m-1} + (y_n - \mu_n^{m-1})(\partial\eta/\partial\mu)_n^{m-1},\ n = 1, ..., N$
  $w_n^m \leftarrow \{(\partial\mu/\partial\eta)_n^{m-1}\}^2 V_n^{-1},\ n = 1, ..., N$
  $(f_1^m(x_1), ..., f_d^m(x_d)) \leftarrow \text{BACKFIT}[\mathbf{z}^m, \mathrm{x}, \mathbf{w}^m]$
until fit fails to improve

In the above algorithm V_n is the *variance function* $b''(\theta)$ computed at the point $\theta_n = b'^{-1}(g^{-1}(\eta_n))$ and $\text{BACKFIT}[\mathbf{z}^m, \mathrm{x}, \mathbf{w}^m]$ stands for the weighted fit of an additive model to the adjusted responses $z_n^m,\ n = 1, ..., N$ with weights $w_n^m,\ n = 1, ..., N$ carried out via the *Backfitting Algorithm* [28]:

$f_1^{m,0}(x_1) \leftarrow 0, \ldots, f_d^{m,0}(x_d) \leftarrow 0$
$l \leftarrow 0$
repeat
  $l \leftarrow l + 1$

for $j = 1$ to d **do**

$r_n^j \leftarrow z_n^m - \sum_{k=1}^{j-1} f_k^{m,l}(x_{kn}) - \sum_{k=j+1}^{d} f_k^{m,l-1}(x_{kn}),\ n = 1, ..., N$

$f_j^{m,l}(x_j) \leftarrow \mathrm{WSM}[\mathbf{r}^j, x_j, \mathbf{w}^m]$

end for

until fit fails to improve

where $\mathrm{WSM}[\mathbf{r}^j, x_j, \mathbf{w}^m]$ stands for the weighted regression of the residuals $r_n^j,\ n = 1, ..., N$ on the covariate $x_{jn},\ n = 1, ..., N$ with weights $w_n^m,\ n = 1, ..., N$ obtained together with the responses $z_n^m,\ n = 1, ..., N$ at the m-th step of the Local Scoring Algorithm outlined before. To perform such regression one can use any of the known smoothers (e.g. running line smoother, regression splines etc [9], [56]). It should be noted that the idea of Generalized Additive Models can be extended to deal with situations where the distribution of the response variable no longer belongs to the exponential family [28].

Generalized Additive Models provide a flexible tool for dealing with various situations and have met with a considerable success in Data Mining Applications [38]. Unfortunately, additive models are not adequate in some cases and, although the same approach can be used to include interaction terms in a model, the problem of (automatic) model selection remains open.

The Support Vector Machines (SVM) approach [57] allows one to perform regression analysis of high-dimensional data sets. The model for a regression function is constructed in the form of an expansion on a set of basis functions each of which is determined by a single element of the data set called the Support Vector. So, there are as many basis functions in the model as there are Support Vectors in the data and, therefore, one may say that the SVM algorithm performs compression of the data in the sense that the regression surface can be reconstructed using only Support Vectors. To be more specific, let us consider the simplest case, where the regression function $f(\mathbf{x})$ is modelled as a linear function

$$\hat{f}(\mathbf{x}) = \beta_0 + \sum_{j=1}^{d} \beta_j x_j.$$

According to the Support Vector Machine algorithm, one determines the coefficients (β_0, β) via minimization of the following functional:

$$R(\beta_0, \beta) = \frac{1}{N}\sum_{n=1}^{N} |y_n - \hat{f}(\mathbf{x}_n)|_\epsilon + \gamma||\beta||^2 \tag{1.8}$$

with respect to (β_0, β), where $||\beta||^2$ is a l_2 norm of the vector β, γ is some constant and

$$|y - \hat{f}(x)|_\epsilon = \begin{cases} \epsilon, & \text{if } |y - \hat{f}(x)| \leq \epsilon \\ |y - \hat{f}(x)|, & \text{otherwise,} \end{cases}$$

is the so-called *loss function with ϵ-insensitive zone.* It can be shown [57] that the β-component of the pair $(\hat{\beta}_0, \hat{\beta})$ minimizing (1.8) can be expressed as [24]

$$\hat{\beta} = \sum_{n=1}^{N} (\alpha_n^* - \alpha_n)\mathbf{x}_n, \tag{1.9}$$

where the coefficients $\{\alpha_n^*, \alpha_n\}_{n=1}^N$ are the ones that maximize the functional

$$W(\alpha^*, \alpha) = -\epsilon\sum_{n=1}^{N}(\alpha_n^* + \alpha_n) + \sum_{n=1}^{N} y_n(\alpha_n^* - \alpha_n) - \frac{1}{2}\sum_{n,m=1}^{N}(\alpha_n - \alpha_n^*)(\alpha_m - \alpha_m^*)(\mathbf{x}_n, \mathbf{x}_m)$$

subject to the constraints

$$\begin{aligned} \sum_{n=1}^{N} \alpha_n^* &= \sum_{n=1}^{N} \alpha_n, \\ 0 \leq \alpha_n^* \leq C, &\qquad n = 1, ..., N, \\ 0 \leq \alpha_n \leq C, &\qquad n = 1, ..., N. \end{aligned}$$

Here the value of C depends on the value of the parameter γ. So, the regression plane can be cast as

$$\hat{f}(\mathbf{x}) = \hat{\beta}_0 + \sum_{n=1}^{N}(\alpha_n^* - \alpha_n)(\mathbf{x}, \mathbf{x}_n). \tag{1.10}$$

It is expected (though not proved) that only a relatively small number of quantities $\alpha_n^* - \alpha_n$ will be distinct from zero. The predictor data vectors in (1.9) corresponding to nonzero $\alpha_n^* - \alpha_n$ are called the Support Vectors. Given the support vectors, the coefficient $\hat{\beta}_0$ in (1.10) can be computed according to the formula [24]

$$\hat{\beta}_0 = y_{l_{\text{supp}}} + \epsilon - \sum_{n=1}^{N}(\alpha_n^* - \alpha_n)(\mathbf{x}_{l_{\text{supp}}}, \mathbf{x}_n).$$

The vector $\mathbf{x}_{l_{\text{supp}}}$ appearing in the above formula is any of the Support Vectors for which $0 < |\alpha^*_{l_{\text{supp}}} - \alpha_{l_{\text{supp}}}| < C$. The parameter ϵ regulates the number of the Support Vectors (i.e. the complexity of the regression surface) while C determines the tradeoff between the bias and the variance of the estimate of the regression function given the level of its complexity defined by ϵ. Thus, as was pointed out earlier, $\hat{f}(\mathbf{x})$ is an expansion on a set of basis functions $(\mathbf{x}, \mathbf{x}_n)$, $n \in I_{\text{support}} \subset \{1, ..., N\}$ each of which is determined by a Support Vector. This procedure can be extended to the case of nonlinear regression. To achieve this, let us consider a mapping T that maps our original predictor space onto some infinite dimensional Hilbert space H chosen a priori and called *feature space* [57] according to the following rule:

$$T(\mathbf{x}) = \mathbf{z} = (\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \ldots)$$

where $\{\phi_i(\mathbf{x}) \in L_2(R^d)\}_{i=1}^{\infty}$ are some basis functions. Assume that the scalar product in H is defined in such a way that

$$(T(\mathbf{x}_1), T(\mathbf{x}_2)) = K(\mathbf{x}_1, \mathbf{x}_2) = \sum_{j=1}^{\infty}\xi_j\phi_j(\mathbf{x}_1)\phi_j(\mathbf{x}_2) \tag{1.11}$$

where ξ_j's are some positive quantities and $K(\mathbf{x}_1, \mathbf{x}_2)$ is a kernel function which satisfies

Mercer's Theorem of the Hilbert space theory [10]. Now, having mapped the original data into the feature space H: $(\mathbf{z}_n, y_n)$, $n = 1, ..., N$, one can utilize the methodology outlined above to build the regression plane in H. The coefficients determining the plane are

$$\begin{aligned}\hat{\beta} &= \sum_{n=1}^{N}(\alpha_n^* - \alpha_n)\mathbf{z}_n, \\ \hat{\beta}_0 &= y_{l_{\text{supp}}} + \epsilon - \sum_{n=1}^{N}(\alpha_n^* - \alpha_n)(\mathbf{z}_{l_{\text{supp}}}, \mathbf{z}_n), \quad 0 < |\alpha_{l_{\text{supp}}}^* - \alpha_{l_{\text{supp}}}| < C.\end{aligned}$$

Thus, the regression plane in H takes the form:

$$\hat{f}(\mathbf{z}) = \hat{\beta}_0 + \sum_{n=1}^{N}(\alpha_n^* - \alpha_n)(\mathbf{z}, \mathbf{z}_n).$$

Taking (1.11) into consideration, we arrive at the following model for the regression function defined over the original predictor space R^d:

$$\hat{f}(\mathbf{x}) = \hat{\beta}_0 + \sum_{n=1}^{N}(\alpha_n^* - \alpha_n)K(\mathbf{x}, \mathbf{x}_n),$$

where the coefficients $\{\alpha_n^*, \alpha_n\}_{n=1}^N$ are the ones that maximize the functional

$$W(\alpha^*, \alpha) = -\epsilon\sum_{n=1}^{N}(\alpha_n^* + \alpha_n) + \sum_{n=1}^{N} y_n(\alpha_n^* - \alpha_n) - \frac{1}{2}\sum_{n,m=1}^{N}(\alpha_n - \alpha_n^*)(\alpha_m - \alpha_m^*)K(\mathbf{x}_n, \mathbf{x}_m)$$

subject to the constraints

$$\begin{aligned}\sum_{n=1}^{N}\alpha_n^* &= \sum_{n=1}^{N}\alpha_n, \\ 0 \le \alpha_n^* \le C, &\qquad n = 1, ..., N, \\ 0 \le \alpha_n \le C, &\qquad n = 1, ..., N. \end{aligned} \tag{1.12}$$

Again, some of $(\alpha_n^* - \alpha_n),\ n = 1, ..., N$ may turn out to be zero. Thus, the regression surface estimated by the SVM algorithm is written as

$$\hat{f}(\mathbf{x}) = \hat{\beta}_0 + \sum_{n \in I_{\text{support}}} (\alpha_n^* - \alpha_n) K(\mathbf{x}, \mathbf{x}_n),$$

where

$$\hat{\beta}_0 = y_{l_{\text{supp}}} + \epsilon - \sum_{n \in I_{\text{support}}} (\alpha_n^* - \alpha_n) K(\mathbf{x}_{l_{\text{supp}}}, \mathbf{x}_n),\ \ 0 < |\alpha_{l_{\text{supp}}}^* - \alpha_{l_{\text{supp}}}| < C$$

and $I_{\text{support}} \subset \{1, ..., N\}$ is the set of indexes corresponding to nonzero quantities $(\alpha_n^* - \alpha_n),\ n = 1, ..., N$.

The SVM algorithm is claimed to be able to produce accurate models for multivariate regression functions [58]. However, from our point of view, it suffers from two major drawbacks. First, it does not perform model selection. The models produced by the SVM algorithm are of the "black box" type and, in this sense, similar to models produced by neural networks. Second, the dimensionality of the quadratic optimization problem to be solved in order to determine the coefficients $\alpha_n^*, \alpha_n,\ n = 1, ..., N$ is equal to the size of the data set. Thus, the algorithm is likely to be too costly to apply to the solution of Data Mining problems though some attempts have been made to overcome this deficiency [47].

The Projection Pursuit Algorithm [22] builds regression models of the form

$$\hat{f} = \sum_{j=1}^{M} \hat{f}_j(\mathbf{a}_j \cdot \mathbf{x})$$

that is, the model $\hat{f}(\cdot)$ is a sum of smooth univariate functions whose arguments are linear combinations of the predictor variables. These functions and the corresponding vectors of coefficients $\mathbf{a}_j$ are determined to produce a good fit to the data. The algorithm can be described as follows:

$M \leftarrow 0$

$r_n \leftarrow y_n, \quad n = 1, ..., N$

repeat

$\mathbf{a}_{M+1} \leftarrow \text{argmin}_{\mathbf{a}} I(\mathbf{a})$

$r_n \leftarrow r_n - S(\mathbf{a}_{M+1} \cdot \mathbf{x}_n; \mathbf{a}_{M+1}), \quad n = 1, ..., N$

$\hat{f}_{M+1}(\mathbf{a}_{M+1} \cdot \mathbf{x}) \leftarrow S(\mathbf{a}_{M+1} \cdot \mathbf{x}; \mathbf{a}_{M+1})$

$M \leftarrow M + 1$

until $I(\mathbf{a}_{M+1}) > \epsilon$

Here $I(\mathbf{a})$ is a *figure of merit* for a given vector $\mathbf{a}$ that is defined as

$$I(\mathbf{a}) = 1 - \frac{\sum_{n=1}^{N}(r_n - S(\mathbf{a} \cdot \mathbf{x}_n; \mathbf{a}))^2}{\sum_{n=1}^{N} r_n^2}.$$

The function $S(z; \mathbf{a})$ is obtained via the (smooth) regression of the current residuals r onto the covariate $z = (\mathbf{a} \cdot \mathbf{x})$. Thus, the figure of merit is a proportion of the variance of the residuals $r_n, \quad n = 1, ..., N$ unexplained after the smoother has been applied to the data $(z_n, r_n), \ z_n = (\mathbf{a} \cdot \mathbf{x}_n), \quad n = 1, ..., N$. The smoother proposed in [22] is a four-stage procedure based on the locally linear smoothing with varying bandwidth parameter.

The advantages of this approach are that it is able to overcome the sparsity limitations of kernel and nearest-neighbour techniques since the procedure is based on a univariate smoothing, and many classes of functions can be approximated quite well even for small to moderate values of M. Disadvantages of the projection pursuit are that there still exist some simple functions that require the large number of terms M in the model to ensure an adequate approximation, and the algorithm does not perform the model selection in terms of the original explanatory variables.

The Bayesian model selection algorithm has been introduced to estimate linear regression models with normal errors :

$$\mathbf{y} = X\beta + \epsilon,$$

where X is a model matrix and $\epsilon \sim N(0, I\sigma^2)$. There have been proposed many variations of the Bayesian model selection algorithm (see, for example, [8], [23], [50]). Here we will follow the one described in the paper [50]. Let γ be the vector of indicator variables with

the i-th element γ_i such that $\gamma_i = 0$ if $\beta_i = 0$ and $\gamma_i = 1$ otherwise. Given γ, we define β_γ as a vector consisting of all nonzero elements of β and X_γ as a matrix comprised of columns of X corresponding to those elements of γ that are equal to one. The following prior assumptions are generally made:

- Given γ and σ^2, the prior for β_γ is $\beta_\gamma \sim N\left(0, C\sigma^2(X'_\gamma X_\gamma)^{-1}\right)$, where C is a large positive scale factor. This corresponds to a very spread out prior for β_γ and emphasizes our lack of the prior knowledge concerning the true distribution of the model parameters.
- The prior of σ^2 given γ is $p(\sigma^2|\gamma) \propto 1/\sigma^2$.
- The γ_i are assumed to be a priori independent with $p(\gamma_i = 1) = \pi_i, 0 \leq \pi_i \leq 1$, $i = 1, \ldots, P$, where P is the number of the regression coefficients β.

For a given γ, let $q_\gamma = \sum_{i=1}^{P} \gamma_i$ be the number of nonzero elements of β and

$$S(\gamma) = (\mathbf{y}'\mathbf{y} + C \cdot \mathrm{SSR})/(C+1),$$

where SSR is the residual sum of squares corresponding to the least squares fit of the model determined by γ. It can be shown [50] that

$$p(\mathbf{y}|\gamma) \propto (1+C)^{q_\gamma/2} S(\gamma)^{-N/2}.$$

Therefore, the posterior distribution of γ is

$$p(\gamma|\mathbf{y}) \propto p(\mathbf{y}|\gamma)p(\gamma) \propto (1+C)^{q_\gamma/2} S(\gamma)^{-N/2} \prod_{i=1}^{P} \pi^{\gamma_i}(1-\pi)^{1-\gamma_i} \tag{1.13}$$

In order to sample from this distribution, one can use the *Gibbs Sampler Algorithm* [6]:

$\gamma^{[0]} \leftarrow (\gamma_1^{[0]}, \ldots, \gamma_P^{[0]})$ {Choose an initial value for γ}
for $j = 1$ to M **do**

for $i = 1$ to P **do**
  sample $\gamma_i^{[j]}$ from $p(\gamma_i|\mathbf{y}, \gamma_1^{[j]}, \ldots, \gamma_{i-1}^{[j]}, \gamma_{i+1}^{[j-1]}, \ldots, \gamma_P^{[j-1]})$
end for
end for

The conditional probability of γ_i can be obtained from (1.13):

$$p(\gamma_i|y, \gamma_{j\neq i}) \propto p(\mathbf{y}|\gamma)p(\gamma_i) \propto (1+C)^{q_\gamma/2} S(\gamma)^{-N/2} \pi^{\gamma_i} (1-\pi)^{1-\gamma_i}. \tag{1.14}$$

Since γ_i is a binary random variable, the conditional probability $p(\gamma_i|\mathbf{y}, \gamma_{j\neq i})$ is obtained by evaluating (1.14) for $\gamma_i = 0$ and $\gamma_i = 1$ and normalizing. The number of iterates M generated by the algorithm is determined based on the needs of the problem in hand.

The posterior distribution $p(\gamma|\mathbf{y})$ has support on a parameter space of the size 2^P making it difficult to find its mode by direct enumeration when P is large. Therefore, the mode of the posterior density is estimated based on the fact that the Gibbs iterates $\gamma^{[k]}$ are located in regions of high probability. The value of $\gamma^{[k]}$, $k = 1, \ldots, M$, maximizing $p(\gamma|\mathbf{y})$ is taken as an estimate of the posterior mode of $p(\gamma|\mathbf{y})$ and denoted by γ_{mod}. The regression parameters β are then estimated by the least squares fit based on the model corresponding to γ_{mod}.

The approach based on the Bayesian model selection is very flexible and allows one to produce a variety of models [8], [50]. For example, one can model the regression function $f(\mathbf{x})$ as a linear combination of some basis functions. In this case, each basis function can be treated as "predictor variable" and the selection of the best subset of the basis functions can be carried out using the above procedure. However, the cost of using this procedure is quite high and is roughly proportional to the number of columns of the model matrix X. This number is very large ($\sim 10^{10}$) for, for instance, such a popular choice of basis functions as a full set of tensor product basis functions.

The Regression Tree approach [4] is based on models $\hat{f}(\mathbf{x})$ of the form

$$\hat{f}(\mathbf{x}) = \sum_{t \in T} \beta_t I(\mathbf{x} \in t).$$

Here T is the set of disjoint subregions (called *terminal nodes*) representing a partition of the predictor domain. The algorithm uses the data to simultaneously estimate a good set of subregions T and the parameters $\{\beta_t\}$, $t \in T$. The procedure consists of two stages: the first one grows the so-called *binary tree* which, essentially, represents the history of the process of the recursive splitting of the data set. The set of terminal nodes of the tree defines the partition of the predictor domain into a number of disjoint subregions. The process run as follows. Initially, all the data is contained in one node called the *root* node. At each step the data is split by dividing it into two parts. The first part is made up of the data points defined by the value of a predictor variable being less than the split point and the second part is the remainder. The variable to be used for splitting and the split point itself are chosen to minimize the residual sum of squares. The same splitting rule is applied recursively to the resulting subdomains until a large tree containing only a few data points in each subregion has been grown. It should be noted that, in principle, more complex splits based on a linear combination of variables can be used to grow the tree.

Since the small number of observations in each node may lead to a very complex tree as well as to a high variance of the regression estimate, the recombination of nodes can improve the prediction and interpretation of the final model. So, during the second stage of the procedure known as *tree pruning*, a nested sequence of subtrees is obtained by removing some of the branches of the tree produced in the course of the first stage. To measure the performance of each of the subtrees, one can use the so-called *cost-complexity* measure defined as

$$C(\tilde{T}) = \sum_{t \in \tilde{T}} \sum_{\mathbf{x}_n \in t} (y_n - \beta_t)^2 + \alpha |\tilde{T}|,$$

where α can be interpreted as a penalty per terminal node in the tree, $|\tilde{T}|$ is the number of the terminal nodes in a tree and $\tilde{T}$ is a subtree of the tree T grown during the first stage. So, the subtree having minimal cost-complexity is chosen to represent the final model. Of course, the structure of the final model depends on the value of the parameter α. The best value for this parameter can be obtained through minimization of some estimate of the prediction error of the model (normally, the L-fold cross-validation criterion is used as the estimate).

The Regression Tree approach possesses a number of very appealing properties: models

are easily interpretable via a binary tree model representation and they are quite cheap to build. Nevertheless, the approach suffers from some limitations. In particular, the resulting regression function $\hat{f}(\mathbf{x})$ is discontinuous at the subregion boundaries which may result in quite poor accuracy of the fit. For example, it fails to approximate some simple functions such as certain types of linear functions. Also, in some cases the algorithm produces very complex trees which are difficult to interpret.

1.3 Overview of the Contents of the Thesis

As we saw in the previous section, there has been proposed a variety of techniques for performing regression analysis though most of them have various limitations that are likely to be hampering factors as far as Data Mining is concerned. In this thesis we will focus on two methodologies which, we believe, have a very bright future. The first of them, the Least Absolute Shrinkage and Selection Operator (LASSO) [55] was proposed by R. Tibshirani. It amounts to minimization of the residual sum of squares of a model subject to the l_1 norm of the regression coefficients being less than a constant. LASSO appears to enjoy the most favourable properties of both ridge regression and subset selection algorithms [33], [39], [43]. In Chapters 2 and 3, we propose and investigate the properties of the generalized version of LASSO called PLASM allowing for grouping of the regression coefficients. The issues related to numerical determination of the PLASM solutions are considered in Chapter 4. Chapter 5 introduces a modified version of PLASM which turns out to be closely related to the well-known Penalized Least Squares approach [26], while Chapters 6 and 7 are concerned with possible extensions of the ideas of PLASM to l_1 regression.

The second approach we will be concerned with in this thesis is the Multivariate Adaptive Regression Adaptive Splines (MARS) [20] algorithm by J. Friedman. It is one of the most successful large scale regression tools proposed so far. Basically, it utilizes the same recursive partitioning strategy as that used in the Regression Tree approach [4] though, unlike the latter, MARS produces continuous models. Due to the extremely flexible strategy of the algorithm, it is able to perform model selection as well as handle both continuous and categorical predictors. In Chapters 8 and 9, we will provide an in depth discussion of MARS and introduce a new version of this procedure called BMARS based on B-splines and on a somewhat different model building strategy. Also, we will discuss a parallel

implementation of BMARS (section 8.4) and its application to the solution of a more or less typical Data Mining problem (section 10.3). In spite of the success of MARS, so far there have been no publications intended to investigate the convergence properties of the algorithm. Chapter 11 is an attempt to carry out that sort of study. In this Chapter, we will introduce a relatively simple procedure based on the so-called *greedy* model building strategy [21] similar (to some extent) to the strategy of MARS and investigate its convergence properties.

In the conclusion (Chapter 12), we will recap on the main points of the thesis and outline directions for the future research.

Chapter 2
Probing Least Absolute Squares Modelling

This chapter starts the first part of thesis which is dedicated to the study of a new approach called Probing Least Absolute Squares Modelling (PLASM). The idea of PLASM was inspired by a paper on the Least Absolute Shrinkage and Selection Operator (LASSO) by R. Tibshirani [55].

Before we start our discussion of LASSO, we would like to give a formulation of the regression estimation problem once again. It does not differ form the formulation given in the introductory chapter 1 conceptually but, rather it is intended to emphasize the issues we will be concerned with in the second part of the thesis.

Assume we are given a dataset $(\mathbf{x}_n, y_n)$, $n = 1, 2, \ldots, N$, where $\mathbf{x}_n \in R^d$, $n = 1, \ldots, N$ are predictor vectors and y_n, $n = 1, \ldots, N$ are the corresponding response value. Also, assume that the response values are related to the predictors in the following way:

$$y_n = f(\mathbf{x}_n) + \epsilon_n,$$

where $f(\mathbf{x})$ is a regression function to be estimated based on the data, and ϵ_n, $n = 1, \ldots, N$ are independent identically distributed random variables such that

$$E(\epsilon_n) = 0, \ n = 1, \ldots, N.$$

We will be concerned with the following model for the regression function $f(\mathbf{x})$

$$f(\mathbf{x}) = \beta_0 + \sum_{j=1}^{P} B_j(\mathbf{x}_n)\beta_j. \tag{2.1}$$

Here $B_j(\mathbf{x})$, $j = 1, ..., P$ are some basis functions, β_0 is an *intercept*, and β_j, $j = 1, \ldots, P$ are *regression coefficients*. As was pointed out in the introductory chapter 1, we are interested in problems where the number of predictor variables d and the number of datapoints N are large (say, 40 and $1,000,000$ respectively). The large number of predictors implies that, in order to ensure that the space of all possible models of the form (2.1) is large enough to contain an adequate model, the number of basis functions P is likely to be large too. So, the easiest solution would be to estimate the regression coefficients of the model comprised of all basis functions $B_j(\mathbf{x})$, $j = 1, ..., P$. However, the final model is often required to be as simple as possible so that it would be easy to interpret. To achieve this, one would need a procedure that could select a reasonably accurate model containing only a relatively small subset of all basis functions. The second feature of our formulation (N is large) means that, in order to be practical, the selection procedure would have to have complexity linear in the number of datapoints N.

2.1 Least Absolute Shrinkage and Selection Operator

LASSO is a procedure intended to tackle the problem of the selection of accurate and interpretable models. According to the LASSO approach, one can estimate β_j's and β_0 via solution of the following optimization problem:

$$\begin{aligned} (\beta, \beta_0) \quad = \quad & \text{argmin } (\mathbf{y} - T\beta - \beta_0)^T(\mathbf{y} - T\beta - \beta_0) \\ & \text{subject to } \sum_{j=1}^{P} |\beta_j| \leq t, \end{aligned} \tag{2.2}$$

where T is a $N \times P$ full rank model matrix whose entries are computed as:

$$T_{nj} = B_j(\mathbf{x}_n),\ n = 1, ..., N,\ j = 1, ..., P,$$

and $t > 0$ is a free parameter of the procedure. Given the solution (β^*, β_0^*) of (2.2), β_0^* relates to β^* as follows:

$$\beta_0^* = \frac{1}{N}\sum_{n=1}^{N}(y_n - \sum_{i=1}^{P} B_j(\mathbf{x}_n)\beta_j^*).$$

Therefore, the optimization problem (2.2) can be reformulated in terms of β only:

$$\begin{aligned} \beta &= \text{argmin } (\mathbf{y} - \bar{\mathbf{y}} - A\beta)^T(\mathbf{y} - \bar{\mathbf{y}} - A\beta) \\ &\quad \text{subject to } \sum_{j=1}^{P}|\beta_j| \leq t. \end{aligned} \tag{2.3}$$

Here A is derived from T via centering the columns of the latter

$$A = (I_N - \frac{ee^T}{N})T,$$

where I_N is a $N \times N$ identity matrix and e is a vectors of ones. One can assume without any loss of generality that $\bar{\mathbf{y}} = 0$ and recast the LASSO optimization problem as

$$\begin{aligned} \beta &= \text{argmin } (\mathbf{y} - A\beta)^T(\mathbf{y} - A\beta) \\ &\quad \text{subject to } \sum_{j=1}^{P}|\beta_j| \leq t. \end{aligned} \tag{2.4}$$

Tibshirani demonstrated in [55] that LASSO enjoys some of the favourable properties of two other well-known regression modelling techniques: subset selection and ridge regression. Subset selection builds a model based on, for example, a forward or backward selection strategy [39]. Although they are able to produce relatively simple models, such subset selection algorithms suffer from excessive variability. The ridge regression procedure [30],[31] is, in a sense, the opposite of the subset selection: it does not perform a model selection but stabilizes the variance of the estimated parameters and, therefore, can generate reasonably accurate models which, unlike models produced via subset selection,

are stable with respect to small changes of data. Ridge regression estimates coefficients $\beta_j,\ j = 1, ..., P$ via minimization of the residual sum of squares subject to a l_2-norm of coefficients being less than a free parameter:

$$\begin{aligned} \beta &= \text{argmin } (\mathrm{y} - A\beta)^T(\mathrm{y} - A\beta) \\ &\quad \text{subject to } [\sum_{j=1}^{P} \beta_j^2]^{\frac{1}{2}} \leq t. \end{aligned} \tag{2.5}$$

It turns out that LASSO sets some of the regression coefficients to zero producing interpretable models (like subset selection) and displays the stability similar to that of ridge regression.

2.2 Introduction of PLASM

LASSO has proved to be quite efficient at building accurate and simple models. However, there are situations where it appears to be more natural (and often more advantageous) to perform model selection in terms of groups of regression coefficients rather than in terms of the individual ones. To clarify this point, let us consider the following model of the regression function:

$$f(\mathrm{x}) = f_1(x_1) + ... + f_d(x_d), \tag{2.6}$$

where

$$f_i(x_i) = \sum_{j=1}^{p_i} \beta_{ij} B_{ij}(x_i),\ i = 1, ..., d.$$

Here $B_{ij}(x_i),\ j = 1, ..., p_i$ are univariate basis functions of the predictor x_i. So, the model (2.6) is a sum of univariate functions f_i each of which is modelled as a linear combination of some univariate basis functions. Note that now we use two subscripts

to index the regression coefficients β_{ij}: the first one refers a predictor variable while the second subscript indexes univariate basis functions $B_{ij}(x_i)$ of that predictor. This type of model is called an additive model [28]. In this situation the simplicity of the model is determined by the number of univariate functions f_i present rather than by the number of individual basis functions. So, in this situation, it seems more appropriate to select the model in terms of functions f_i, $i = 1, ..., d$ or, in other words, in terms of groups of regression coefficients $\{\beta_{ij},\ j = 1, ..., p_i\}_{i=1}^{d}$. Therefore, we propose a procedure which performs this kind of model selection:

$$\begin{aligned} \beta &= \text{argmin } (\mathbf{y} - A\beta)^T(\mathbf{y} - A\beta) \\ &\quad \text{subject to } \sum_{i=1}^{d}[\beta_i^T\beta_i]^{\frac{1}{2}} \leq t. \end{aligned} \tag{2.7}$$

Here $\beta_i^T = (\beta_{i1}, \ldots, \beta_{ip_i})$ is a vector of coefficients of the i-th group, i.e. $\beta^T = (\beta_1^T, \ldots, \beta_d^T)$ and $\sum p_i = P$. As was pointed out by the examiners, it is feasible to choose other constraints. For example, one may consider the following expression:

$$\begin{aligned} \sum_j |\beta_{ij}| &\leq c_i, \quad i = 1, \ldots, d \\ \sum_i c_i &\leq t, \quad c_i \geq 0, \quad i = 1, \ldots, d, \end{aligned}$$

which corresponds to a constraint on the sum of supremum norms of groups of regression coefficients. However, as we show later in this thesis, the optimization problem (2.7) can be replaced with an alternative optimization problem of considerably lower dimensionality. To the best of the author's knowledge, whether the same trick is possible with other norms or not is an open question.

Note that both LASSO (2.4) and PLASM (2.7) contain a free parameter t which controls the extent of the influence of the respective constraints on the estimates of regression coefficients. Normally, the optimal value for t is determined via minimization of some estimate of the future predictive error of the resulting model. The range of the appropriate values for t is

$$(0, t_r), \quad \text{where } t_r = \sum_{i=1}^{d} [\beta_i^{oT} \beta_i^o]^{\frac{1}{2}} \tag{2.8}$$

and β^o is an unconstrained least squares solution.

It should be emphasized at this point that PLASM allows for arbitrary grouping of coefficients and the additive modelling considered above is just an example. This allows us to establish the fact that PLASM occupies an intermediate position between LASSO and ridge regression. Indeed, to link PLASM to LASSO, let us consider a *fine* grouping where each group contains only one regression coefficient, i.e. $p_i = 1,\ i = 1, ..., d$ and $d = P$. As can be seen the constraint of (2.7) takes the form $\sum_{j=1}^{P} [\beta_j^2]^{\frac{1}{2}} \leq t$ or $\sum_{j=1}^{P} |\beta_j| \leq t$ which coincides with the constraint of LASSO in (2.4). Now, let us consider the other extreme where all regression coefficients are grouped together in one group, i.e. $p_i = P,\ i = 1, ..., d$ and $d = 1$. Thus, the PLASM constraint becomes $[\sum_{j=1}^{P} \beta_j^2]^{\frac{1}{2}} \leq t$. This is a constraint of ridge regression (2.5). Thus, due to the above connections, one would expect that PLASM sets some of the groups of coefficients to zero while the others are estimated in the way similar to that of the ridge regression.

Obviously, without an efficient numerical procedure for solution of the optimization problem (2.7), our approach would be of very limited value. The straightforward approach to this problem would be to consider an algorithm based on the numerical solution of the corresponding first-order necessary conditions (Kuhn-Tucker conditions) [35],[18]. However, the Kuhn-Tucker conditions involve derivatives of the constraint in (2.7) which may not exist in the classical sense as PLASM sets some groups of the regression coefficients to zero. One could try to circumvent this difficulty by recasting the optimization problem (2.7) in the equivalent form

$$\begin{aligned} \beta \quad = \quad & \text{argmin } (\mathbf{y} - A\beta)^T (\mathbf{y} - A\beta) \\ & \text{subject to } \beta_i^T \beta_i = \tau_i^4, \quad i = 1, \ldots, d, \\ & \sum_{i=1}^{d} \tau_i^2 \leq t, \end{aligned} \tag{2.9}$$

where $\tau_i,\ i = 1, ..., d$ are auxiliary variables. The respective Lagrangian can be written as

$$L = (\mathbf{y} - A\beta)^T(\mathbf{y} - A\beta) + \sum_{i=1}^{d} \mu_i(\beta_i^T\beta_i - \tau_i^4) + \lambda(\sum_{i=1}^{d} \tau_i^2 - t),$$

where μ_i, $i = 1, \ldots, d$ and λ are Lagrange multipliers. The corresponding first-order necessary conditions are

$$\begin{aligned} A^TA\beta - A^T\mathbf{y} + M\beta &= 0, \\ -4\mu_i\tau_i^3 + 2\lambda\tau_i &= 0, \quad i = 1, \ldots, d, \\ \beta_i^T\beta_i - \tau_i^4 &= 0, \quad i = 1, \ldots, d, \\ (\sum_{i=1}^{d} \tau_i^2 - t)\lambda &= 0. \end{aligned} \tag{2.10}$$

Here M is a $P \times P$ diagonal matrix

$$M = \begin{pmatrix} \mu_1 & & & & & & \\ & \ddots & & & & & \\ & & \mu_1 & & 0 & & \\ & & & \ddots & & & \\ & & & & \ddots & & \\ & & 0 & & \mu_d & & \\ & & & & & \ddots & \\ & & & & & & \mu_d \end{pmatrix}. \tag{2.11}$$

The diagonal of the matrix is made up of blocks each block having p_i identical entries equal to μ_i. It can be seen that the Kuhn-Tucker conditions (2.10) may have no solution and an appropriate example is the so-called Orthogonal Design case where A^TA is a unit matrix. Assume that one of the groups $\beta_{i_0} = 0$. It follows that, in this situation, the first equation in (2.10) holds if and only if the corresponding group of entries of the vector $A^T\mathbf{y}$ is equal to zero too which, generally, is not the case. One should note, however, that this does not imply that the problem (2.9) has no solution. Rather, (2.9) has a unique solution due to its equivalence to (2.7) which is a convex problem. The trouble is that if there are

groups of variables at zero level the minimum point of (2.9) may not be a Kuhn-Tucker point and, therefore, one cannot rely on the equations (2.10) to obtain the solution of (2.9) numerically simple because these equations may not hold.

2.3 Regularized PLASM

As the discussion in the previous section showed the Kuhn-Tucker equations cannot be used to solve (2.7). To fix this deficiency we propose to consider a regularized version of PLASM. We would like to point out that this is a temporary measure intended to make further theoretical investigation possible and we will return to the original formulation (2.7) later. So, the regularized PLASM can be formulated as follows

$$\beta = \text{argmin } (\mathrm{y} - A\beta)^T(\mathrm{y} - A\beta)$$

$$\text{subject to } \sum_{i=1}^{d}[\beta_i^T\beta_i + \alpha]^{\frac{1}{2}} \leq t, \tag{2.12}$$

where α is a small parameter. The problem is convex since the objective function as well as each of the terms $[\beta_i^T\beta_i + \alpha]^{\frac{1}{2}}$ in the constraint are convex. Moreover, due to A being a full rank matrix, the objective function is strictly convex and, therefore, the problem has a unique solution $\beta(\alpha)$. As the following Proposition shows, $\beta(\alpha)$ is close to the solution of the original PLASM (2.7).

Proposition 2.3.1 *$\beta(\alpha) \to \beta(0)$ as $\alpha \to 0$, where $\beta(\alpha)$ and $\beta(0)$ are solution of (2.12) and (2.7) respectively corresponding to $t \in (0, t_r)$[1], t_r being defined in (2.8).*

Proof. For the sake of convenience, let us introduce the following notation:

$$\rho[\beta]_\alpha = \sum_{i=1}^{d}[\beta_i^T\beta_i + \alpha]^{\frac{1}{2}}.$$

[1] This condition implies that solutions of both (2.12) and (2.7) are located on the boundaries of the respective feasible regions defined by their constraints.

To prove the Proposition let us assume that the converse holds, that is, $\beta(\alpha)$ does not converge to $\beta(0)$. This implies that there exists $\epsilon > 0$ and $\{\alpha_k\}_{k=1}^{\infty}, \alpha_k \to 0$ such that $||\beta(\alpha_k) - \beta(0)|| > \epsilon, \quad k = 1, 2...$ (here $||\cdot||$ denotes the ordinary Euclidian norm). Due to the fact that all $\beta(\alpha_k)$ are located in a compact region, there exists a subsequence $\{\beta(\alpha_{k_l})\}_{l=1}^{\infty}$ of the sequence $\{\beta(\alpha_k)\}_{k=1}^{\infty}$ such that $\beta(\alpha_{k_l}) \to \beta^*$, where $\beta(0) \neq \beta^*$. Note that $\rho[\beta^*]_0 = t$ and, as $\beta(0)$ is the solution of (2.7),

$$f(\beta^*) > f(\beta(0)), \tag{2.13}$$

where $f(\cdot)$ is the objective function in (2.7) (and (2.12)).

There exists a scalar γ_α such that $\rho[\gamma_\alpha \beta(0)]_\alpha = t$ and $\gamma_\alpha \to 1$ as $\alpha \to 0$. So, $\gamma_\alpha \beta(0) \to \beta(0)$ and $f(\gamma_\alpha \beta(0)) \to f(\beta(0))$. Now, since $\beta(\alpha_{k_l})$ is a solution of (2.12) with $\alpha = \alpha_{k_l}$, one concludes that $f(\beta(\alpha_{k_l})) \leq f(\gamma_{\alpha_{k_l}} \beta(0))$. Therefore, $f(\beta^*) \leq f(\beta(0))$ which contradicts (2.13). Thus, our assumption that $\beta(\alpha)$ does not converge to $\beta(0)$ is wrong and the statement of the Proposition holds. □

While dealing with the regularized PLASM we will assume that t is chosen from the following range: $t \in (t_l^\alpha, t_r^\alpha)$, t_l^α and t_r^α being defined as

$$\begin{aligned} t_l^\alpha &= d\alpha^{\frac{1}{2}}, \\ t_r^\alpha &= \sum_{i=1}^{d} [\beta_i^{oT} \beta_i^o + \alpha]^{\frac{1}{2}}. \end{aligned} \tag{2.14}$$

Here β^o is an unconstrained least squares solution. This assumption ensures that the constraint in (2.12) is active. As can be seen, $(t_l^\alpha, t_r^\alpha) \to (0, t_r)$ when $\alpha \to 0$, where t_r is defined in (2.8). The following technical result will be needed for our future investigations.

Lemma 2.3.1 *Let $\lambda(t, \alpha)$ be a Lagrange multiplier corresponding to the inequality constraint in the regularized PLASM optimization problem (2.12) with $\alpha > 0$ and $t \in (t_l^\alpha, t_r)$, t_l^α and t_r being defined in (2.14) and (2.8) respectively. Then, the limit of $\lambda(t, \alpha)$, as $\alpha \to 0$, exists and $\lim_{\alpha \to 0} \lambda(t, \alpha) = \lambda_0 > 0$.*

Proof. The Kuhn-Tucker conditions for the problem (2.12) are:

$$\mathbf{a}_{ij}^T(A\beta(t,\alpha)-\mathbf{y})+\frac{1}{2}\frac{\lambda(t,\alpha)\beta_{ij}(t,\alpha)}{[\beta_i(t,\alpha)^T\beta_i(t,\alpha)+\alpha]^{\frac{1}{2}}}=0,\quad j=1,...,p_i,\ i=1,...,d. \tag{2.15}$$

Note that $\mathbf{a}_{ij}$ denotes the ij-th column of the matrix A. As was proved before, $\beta(t,\alpha)\to\beta(t,0)$, $\alpha\to 0$. Let $\beta_{i_0j_0}(t,0)$ be any nonzero component in the vector $\beta(t,0)$. It follows from (2.15) that

$$\lambda(t,\alpha)=-\frac{2[\beta_{i_0}(t,\alpha)^T\beta_{i_0}(t,\alpha)+\alpha]^{\frac{1}{2}}\mathbf{a}_{i_0j_0}^T(A\beta(t,\alpha)-\mathbf{y})}{\beta_{i_0j_0}(t,\alpha)}.$$

The limit of the right-hand side, as $\alpha\to 0$, is well defined and, therefore, so is the limit of $\lambda(t,\alpha)$. The fact that $\lambda(t,\alpha)$ converges to a positive value can be deduced from (2.15). Indeed, assume that the converse holds: $\lambda(t,\alpha)\to 0$. Then, the second term of left-hand side in (2.15) tends to zero for all i,j as $\alpha\to 0$. Consequently,

$$A^T(A\beta(t,0)-\mathbf{y})=0.$$

In other words, $\beta(t,0)$ is the unconstrained least squares solution and this contradicts to the condition of the Lemma that $t<t_r$. Thus, our assumption that $\lambda(t,\alpha)\to 0$, $\alpha\to 0$ is wrong and, therefore, the statement of the Lemma holds. □

To continue our investigation of the regularized PLASM, let us recast (2.12) as

$$\begin{aligned}\beta \;=\;& \text{argmin } (\mathbf{y}-A\beta)^T(\mathbf{y}-A\beta)\\ & \text{subject to } \beta_i^T\beta_i+\alpha=\tau_i^4,\quad i=1,\ldots,d,\\ & \qquad\qquad \sum_{i=1}^{d}\tau_i^2\le t.\end{aligned} \tag{2.16}$$

Note that the τ_i^2's are strictly positive in the feasible region. Now we will show that the solution of the problem (2.16) is a Kuhn-Tucker point. Indeed, let us compute the

gradients of the constraints in (2.16):

$$\nabla(\beta_i^T\beta_i + \alpha - \tau_i^4) = \mathbf{g}_i = (0,\ldots,2\beta_i,\ldots,0,0,\ \ldots,-4\tau_i^3,\ldots,0),\ \ i = 1,\ldots,d,$$

and

$$\nabla(\sum_{i=1}^{d}\tau_i^2 - t) = \mathbf{h} = (0,\ldots,\ 0\ \ ,\ldots,0,2\tau_1,\ldots,2\tau_i,\ldots,2\tau_d).$$

The gradients are linearly independent if $\sum\tau_i^2 = t$, which is the case for all boundary points. Indeed, a linear combination of the gradients is

$$\sum_{i=1}^{d}\theta_i\mathbf{g}_i + \theta_{d+1}\mathbf{h} = (2\theta_1\beta_1,\ldots,2\theta_d\beta_d,-4\theta_1\tau_1^3 + 2\theta_{d+1}\tau_1,\ldots,-4\theta_d\tau_d^3 + 2\theta_{d+1}\tau_d).$$

If this combination is equal to zero, then, considering that $\sum\tau_i^2 = t$ as well as (2.14) hold, all of the coefficients $\{\theta_i\}_{i=1}^{d+1}$ have to be equal to zero as well. Thus, all boundary points of the feasible region in (2.16) are Kuhn-Tucker points [18]. By the earlier assumption the solution of (2.16) is located on the boundary and, therefore, it is a Kuhn-Tucker point.

2.4 Kuhn-Tucker Conditions for the Regularised PLASM

According to the results of the previous section, the solution of (2.16) satisfies the following equations:

$$\begin{aligned} A^TA\beta - A^T\mathbf{y} + M\beta &= 0, \\ -4\mu_i\tau_i^3 + 2\lambda\tau_i &= 0,\ \ i = 1,\ldots,d, \\ \beta_i^T\beta_i + \alpha - \tau_i^4 &= 0,\ \ i = 1,\ldots,d, \\ (\sum_{i=1}^{d}\tau_i^2 - t)\lambda &= 0. \end{aligned} \tag{2.17}$$

Now we will show that the system (2.17) can be expressed in an equivalent form involving

only $d+1$ unknown variables as opposed to $P+2d+1$ unknowns in (2.17). To achieve that, let us introduce new variables

$$v_i^2 = \frac{2\tau_i^2}{\lambda}, \quad i = 1, \ldots, d. \tag{2.18}$$

According to (2.17)

$$\mu_i = \frac{1}{v_i^2}, \quad i = 1, \ldots, d,$$

and, consequently

$$\beta = (A^T A + V^{-2})^{-1} A^T \mathbf{y}, \tag{2.19}$$

where V has the same structure as the matrix in (2.11) with μ_i's replaced with v_i^{-2}'s. Now, the equality constraints of the problem (2.16) can be rewritten as

$$\beta^T I_i \beta + \alpha = \frac{\lambda^2}{4} v_i^4, \quad i = 1, \ldots, d, \tag{2.20}$$

where I_i is a diagonal matrix with unities in the entries corresponding to the i-th block and zeros elsewhere:

$$I_i = \begin{pmatrix} 0 & & & & & \\ & \ddots & & & & \\ & & 1 & & 0 & \\ & & & \ddots & & \\ & 0 & & 1 & & \\ & & & & \ddots & \\ & & & & & 0 \end{pmatrix}.$$

One can insert the expression for β (2.19) into (2.20) and obtain the system of equations in terms of $d+1$ variables v_i^2, $i = 1, ..., d$ and λ:

$$\begin{aligned} \mathbf{y}^T A(A^T A + V^{-2})^{-1} I_i (A^T A + V^{-2})^{-1} A^T \mathbf{y} &= \frac{\lambda^2}{4} v_i^4 - \alpha, \quad i = 1, \ldots, d \\ \sum_{i=1}^{d} v_i^2 &= \frac{2t}{\lambda}. \end{aligned} \tag{2.21}$$

This system has a unique solution in terms of v_i^2's and λ.

Lemma 2.4.1 *For each value for the parameter $t \in (t_l^\alpha, t_r^\alpha)$, t_l^α and t_r^α being defined in (2.14), the system (2.21) has a unique solution for v_i^2's and λ.*

Proof. The Lagrangian for the original problem (2.12) is

$$L = (\mathbf{y} - A\beta)^T (\mathbf{y} - A\beta) + \tilde{\lambda} \sum_{i=1}^{p} ([\beta_i^T \beta_i + \alpha]^{\frac{1}{2}} - t),$$

and, consequently, the first-order necessary conditions take the form

$$A^T A\beta - A^T \mathbf{y} + \tilde{V}^{-2}\beta = 0, \tag{2.22}$$

where $\tilde{V}$ has the same structure as (2.11) and $\tilde{v}_i^{-2}$'s are introduced as a short notation for the more complex expressions

$$\tilde{v}_i^{-2} = \frac{\tilde{\lambda}}{2} [\beta_i^T \beta_i + \alpha]^{-\frac{1}{2}}, \quad i = 1, \ldots, d. \tag{2.23}$$

According to (2.22) and (2.23), $\tilde{v}_i^2$'s as well as $\tilde{\lambda}$ satisfy the following system of equations

$$\mathbf{y}^T A(A^T A + \tilde{V}^{-2})^{-1} I_i (A^T A + \tilde{V}^{-2})^{-1} A^T \mathbf{y} = \frac{\tilde{\lambda}^2}{4} \tilde{v}_i^4 - \alpha, \quad i = 1, \ldots, d,$$

$$\sum_{i=1}^{p} \tilde{v}_i^2 = \frac{2t}{\tilde{\lambda}}. \tag{2.24}$$

Note that $\tilde{\lambda}$ is strictly positive since, otherwise the optimal point would coincide with the unconstrained least squares solution for β which, by the assumption $t \in (t_l^\alpha, t_r^\alpha)$, is impossible. Because (2.12) is a strictly convex optimization problem, the system (2.24) has a unique solution for v_i^2 and λ and it has exactly the same form as the system (2.21). Therefore, (2.21) has a unique solution too. □

Note that if the system (2.21) is solved the regression coefficients β can be obtained by (2.19). In the next chapter we will show how this observation can be exploited to produce more insight as well as numerically tractable formulations of both the regularized (2.12) and the original (2.7) PLASMs.

Chapter 3
New Formulation of PLASM

3.1 New Regularised PLASM

In this chapter we will continue our investigation of the PLASM approach and we will start with the introduction of a new optimization problem which can be solved instead of the regularized PLASM (2.12). The reason for pursuing this goal is that it eventually leads to an equivalent formulation (we will use the term "new formulation" from now on) of the original PLASM (2.7). This new formulation will help us understand the nature of the PLASM approach and develop an efficient numerical algorithm. Consider the following optimization problem

$$\begin{aligned} \underset{\mathbf{u}}{\text{minimize}} \quad & -\mathbf{y}^T A(A^T A + U^{-1})^{-1} A^T \mathbf{y} + \sum_{i=1}^{d} \frac{\alpha}{u_i} \\ \text{subject to} \quad & \sum_{i=1}^{d} u_i \leq t', \\ & u_i \geq 0, \quad i = 1, \ldots, d, \end{aligned} \tag{3.1}$$

where t' is a free parameter, α is the same small parameter introduced in the previous chapter, and U is a diagonal matrix having the same structure as M in (2.11) with μ_i's replaced with u_i's. The problem has at least one solution and the Lagrangian associated with it is

$$L = -\mathbf{y}^T A(A^T A + U^{-1})^{-1} A^T \mathbf{y} + \sum_{i=1}^{d} \frac{\alpha}{u_i} + \xi(\sum_{i=1}^{d} u_i - t'), \tag{3.2}$$

where ξ is a Lagrange multiplier, and the Kuhn-Tucker conditions take the form

$$\begin{aligned} \mathbf{y}^T A(A^TA+U^{-1})^{-1}I_i(A^TA+U^{-1})^{-1}A^T\mathbf{y}+\alpha &= \xi u_i^2, \quad i=1,\ldots,d, \\ \xi(\sum_{i=1}^{d} u_i - t') &= 0, \\ u_i &\geq 0, \quad i=1,\ldots,d. \end{aligned} \tag{3.3}$$

To derive these equations, the well-known formula for a derivative of an inverse of a matrix is used:

$$(S^{-1})' = -S^{-1}S'S^{-1}.$$

Note that the positivity constraints $u_i \geq 0$ were disregarded in (3.2) and (3.3) since, due to the term $\sum \alpha/u_i$, no point on the boundaries $u_i = 0,\ i = 1,...,d$ can be a solution. Below we will show that the objective function of (3.1) is strictly convex which, combined with the convexity of its feasible region, will imply that (3.1) has a unique solution.

Proposition 3.1.1 *The objective function of the optimization problem (3.1) is strictly convex.*

Proof. To prove the Proposition, we will show that the first term

$$f = -\mathbf{y}^T A(A^TA+U^{-1})^{-1}A^T\mathbf{y} \tag{3.4}$$

in the objective function is a convex function. This fact along with the strict convexity of the second term for $u_i > 0,\ i = 1,...,d$

$$\sum_{i=1}^{d} \frac{\alpha}{u_i} \tag{3.5}$$

will enable us to deduce that the objective function (the sum of these two terms) is strictly

convex. In order to prove that (3.4) is convex it suffices to demonstrate [35] that the Hessian F of the function f is a positive semidefinite matrix for all $(u_1, \ldots, u_d) : u_i > 0$ and $\sum u_i \leq t'$. Denoting $(A^T A + U^{-1})^{-1}$ by B, one can obtain the first order derivatives of f with respect to u_i, $i = 1 \ldots, d$:

$$F_i = -\frac{1}{u_i^2} \mathbf{y}^T A B I_i B A^T \mathbf{y}.$$

Similarly, the second order derivatives, $j \neq i$:

$$F_{ij} = f''_{u_i u_j} = -\frac{2}{u_i^2 u_j^2} \mathbf{y}^T A B I_i B I_j B A^T \mathbf{y},$$

and $i = j$:

$$F_{ii} = f''_{u_i^2} = \frac{2}{u_i^3} \mathbf{y}^T A B I_i B A^T \mathbf{y} - \frac{2}{u_i^4} \mathbf{y}^T A B I_i B I_i B A^T \mathbf{y}.$$

Now, let $\tilde{\mathbf{x}}$ be a d-dimensional vector. The quadratic form $\tilde{\mathbf{x}}^T F \tilde{\mathbf{x}}$ can be expressed as

$$\begin{aligned} \tilde{\mathbf{x}}^T F \tilde{\mathbf{x}} &= 2\mathbf{y}^T A B \tilde{X}^2 U^{-3} B A^T \mathbf{y} - 2\mathbf{y}^T A B \tilde{X} U^{-2} B \tilde{X} U^{-2} B A^T \mathbf{y} \\ &= 2\mathbf{y}^T A B \tilde{X} U^{-2} [U - B] \tilde{X} U^{-2} B A^T \mathbf{y}. \end{aligned} \tag{3.6}$$

Here $\tilde{X}$ has the same structure as U. Note that $\tilde{\mathbf{x}}$ is a d-dimensional vector whereas $\tilde{X}$ is a $P \times P$ matrix, and $\tilde{X} U = U \tilde{X}$. If $U - B = U - (A^T A + U^{-1})^{-1}$ is a positive definite matrix, the quadratic form (3.6) is nonnegative and the proof is complete. The positive definiteness of $U - B$ can be established based on the following equality

$$U - (A^T A + U^{-1})^{-1} = U^{\frac{1}{2}} [I - (U^{\frac{1}{2}} A^T A U^{\frac{1}{2}} + I)^{-1}] U^{\frac{1}{2}}.$$

Since $U^{\frac{1}{2}} A^T A U^{\frac{1}{2}}$ is a symmetric and positive definite matrix the eigenvalues of $(U^{\frac{1}{2}} A^T A U^{\frac{1}{2}} + I)^{-1}$ are less than unity and, therefore, eigenvalues of $I - (U^{\frac{1}{2}} A^T A U^{\frac{1}{2}} + I)^{-1}$ are greater than zero which establishes the positive definiteness of $U - B$. This, as was pointed out be-

fore, means that the Hessian of the function f in (3.4) is positive semidefinite throughout the feasible region of (3.1) and, therefore, f is a convex function which inevitably entails the strict convexity of the whole objective function. □

Remark. It is seen that the constraint $\sum u_i \leq t'$ is always active since the gradient of the objective function is nonzero (in fact, all of its components are negative) anywhere in the region $u_i > 0,\ i = 1, \ldots, d$. Also, from the Sensitivity Theorem (see, for instance, [35]), it follows that the Lagrange multiplier ξ is always positive.

Now, having established the uniqueness of the solution of our new optimization problem (3.1), we will establish the connection between its solutions and solutions of the Kuhn-Tucker conditions (2.21) for the regularized PLASM (2.12).

Proposition 3.1.2 *There is a one-to-one correspondence between solutions of the system (2.21), $t \in (t_l^\alpha, t_r^\alpha)$ (t_l^α and t_r^α being defined by (2.14)), and solutions of (3.1), $t' \in (0, \infty)$.*

Proof. Given the value for the parameter $t \in (t_l^\alpha, t_r^\alpha)$ and the corresponding solution of (2.21) $v_i^2(t)$ and $\lambda(t)$ one can obtain a solution $u_i(t')$ and $\xi(t')$ of the system (3.3) (Kuhn-Tucker conditions for the optimization problem (3.1)) corresponding to $t' = 2t/\lambda(t)$ according to the following formulas:

$$\begin{aligned} u_i(t') &= v_i^2(t), \quad i = 1, \ldots, d, \\ \xi(t') &= \frac{\lambda(t)^2}{4}. \end{aligned} \tag{3.7}$$

If $t_1 \neq t_2$, then $t_1' \neq t_2'$ since, otherwise that would mean that the system (3.3) has two solutions for some $t' = t_1' = t_2'$ which, as was proved earlier, is impossible. Thus, for every solution $v_i^2(t)$ and $\lambda(t)$, $t \in (t_l^\alpha, t_r^\alpha)$ of the system (2.21) one can construct a solution for the optimization problem (3.1) with $t' = 2t/\lambda(t)$ according to the formula (3.7). Also, solutions of (2.21) obtained for different values for the parameter t correspond to different solutions for the problem (3.1).

Conversely, given the value for the parameter $t' \in (0, \infty)$ and the corresponding solution $u_i(t'),\ i = 1, \ldots, d$ of (3.1) together with the Lagrange multiplier $\xi(t')$ for the constraint $\sum u_i \leq t'$ one can construct the solution $v_i^2(t)$ and $\lambda(t)$

$$\begin{aligned} v_i^2(t) &= u_i(t'), \quad i = 1, \ldots, d, \\ \lambda(t) &= 2\sqrt{\xi(t')} \end{aligned} \tag{3.8}$$

of the system (2.21) corresponding to $t = t'\sqrt{\xi(t')}$. Now, $\beta(t) = (A^TA + v^{-2}(t))^{-1}A^T\mathbf{y}$ satisfies the Kuhn-Tucker conditions (2.22), (2.23) and, therefore, is a unique solution of the optimization problem (2.12) (regularized PLASM). As can be seen, $\beta(t)$ is distinct from the unconstrained least squares solution. Consequently, it is located on the boundary of the feasible region of the problem (2.12). This implies that $t \in (t_l^\alpha, t_r^\alpha)$. If $t_1' \neq t_2'$ then $t_1 \neq t_2$ since, otherwise that would mean that the system (2.21) has two solutions for some $t = t_1 = t_2$ which is impossible due to the uniqueness of the solution of the regularized PLASM (2.12). Thus, for every solution $u_i(t')$, $t' \in (0, \infty)$ of the optimization problem (3.1) one can construct a solution for (2.21) with $t = t'\sqrt{\xi(t')}$ ($\xi(t')$ being a Lagrange multiplier of (3.1) corresponding to the constraint $\sum u_i \leq t'$) according to the formula (3.8), and solutions of (3.1) obtained for different values for the parameter t' correspond to different solutions for the system (2.21).

Thus, we demonstrated that there is a one-to-one correspondence between the solutions of the system (2.21) and the optimization problem (3.1) defined by the formulas (3.8) and (3.7). □

Recognizing that solution of the regularized PLASM (2.12) is equivalent to the solution of the system of equations (2.21), we arrive at the conclusion that, instead of estimating the regression coefficients β based on the regularized PLASM (2.12) with t from the range $t \in (t_l^\alpha, t_r^\alpha)$, one can obtain the estimates of β according to the formula

$$\beta(t') = (A^TA + U^{-1}(t'))^{-1}A^T\mathbf{y} = U(t')(A^TAU(t') + I)^{-1}A^T\mathbf{y}. \tag{3.9}$$

for $t' \in (0, \infty)$, where $U(t')$ is obtained via solution of the optimization problem (3.1). In the next few sections, we will see that this important result can be extended to the original formulation of PLASM (2.7).

3.2 New Form of the Original PLASM

As was pointed out before, the introduction of the regularization parameter α is a technical trick intended to overcome the lack of regularity of the original problem (2.7). We proved in the Proposition 2.3.1 that the solution $\beta(\alpha)$ of the regularized PLASM (2.12) converges to the solution β of the original PLASM (2.7) as α vanishes to zero. Also, in the previous section we found out that the solution of $\beta(\alpha)$, $t \in (t_l^\alpha, t_r^\alpha)$ of (2.12) can be obtained by solving the alternative optimization problem (3.1). The natural question would be if (3.1) with $\alpha = 0$ could be solved instead of the original PLASM (2.7). In this section we will show that the answer is positive.

However, before we tackle this problem we will establish several technical results. The first one is concerned with uniqueness of the solution of the optimization problem

$$\begin{aligned} \underset{\mathbf{u}}{\text{minimize}} \quad & -\mathbf{y}^T A(A^T A + U^{-1})^{-1} A^T \mathbf{y} \\ \text{subject to} \quad & \sum_{i=1}^{d} u_i \leq t', \\ & u_i \geq 0, \quad i = 1, \ldots, d. \end{aligned} \tag{3.10}$$

In order to avoid any possible confusion, we would like to note that the objective function is well-defined even if some of the diagonal elements of the matrix U are equal to zero. Indeed, one can cast it in an alternative form: $-\mathbf{y}^T A(A^T A + U^{-1})^{-1} A^T \mathbf{y} = -\mathbf{y}^T A U^{1/2}(U^{1/2} A^T A U^{1/2} + I)^{-1} U^{1/2} A^T \mathbf{y}$. Since the matrix $U^{1/2} A^T A U^{1/2}$ is at least positive semidefinite the inverse of the matrix $(U^{1/2} A^T A U^{1/2} + I)$ always exists which proves our point. We prefer to use the objective function in the form (3.10) because this will make most of the subsequent formulas look sonsiderably more neat. However, any numerical algorithm intended to solve the abovecited optimization problem would clearly be based on the alternative form of the objective function.

Lemma 3.2.1 *Given that all of the components of the vector $A^T \mathbf{y}$ are distinct from zero, the optimization problem (3.10) has a unique solution for any $t' \in (0, \infty)$.*

Proof. The problem has a solution since the objective function is continuous over the

compact feasible region. To prove the uniqueness assume the converse. Let $\mathbf{u}_1$ and $\mathbf{u}_2$ be the solutions of (3.10) and let the index set σ point to those components of $\mathbf{u}$ which are zero in both solutions. Now let us introduce a reduced optimization problem in terms of variables $\hat{u}_i,\ i = 1, \ldots, \hat{d}$ which are not pointed to by σ

$$\begin{aligned} \underset{\mathbf{u}}{\text{minimize}} \quad & -\mathbf{y}^T \hat{A}(\hat{A}^T\hat{A} + \hat{U}^{-1})^{-1}\hat{A}^T\mathbf{y} \\ \text{subject to} \quad & \sum_{i=1}^{\hat{d}} \hat{u}_i \leq t', \\ & \hat{u}_i \geq 0, \quad i = 1, \ldots, \hat{d}, \end{aligned} \tag{3.11}$$

where $\hat{A}$ is obtained from A by an appropriate reduction. Note that $\hat{\mathbf{u}}$ is a vector while $\hat{U}$ is a matrix. It can be seen that the reduced vectors $\hat{\mathbf{u}}_1$ and $\hat{\mathbf{u}}_2$ are the solutions of the reduced problem. Since the objective function of (3.11) and the feasible region are convex (as was proved in the Proposition 3.1.1), the points

$$\hat{\mathbf{u}}_s = \hat{\mathbf{u}}_1 + s\hat{\mathbf{v}}, \quad s \in (0, 1),$$

are solutions of the reduced optimization problem. Here $\hat{\mathbf{v}} = \hat{\mathbf{u}}_2 - \hat{\mathbf{u}}_1$. Thus $\hat{\mathbf{u}}_{0.5}$ is a solution of (3.11) and all of its components are positive. Now, denoting $f(\hat{\mathbf{u}}_s)$ (f is the objective function) by $f(s)$ one has $f''(0.5) = 0$ which implies that

$$\hat{\mathbf{v}}^T F \hat{\mathbf{v}} = 0,$$

where F is the Hessian of the objective function f. From the previous results (see formula (3.6)) we know that

$$\hat{\mathbf{v}}^T F \hat{\mathbf{v}} = 2\mathbf{y}^T \hat{A}\hat{B}\hat{V}\hat{U}_{0.5}^{-2}[\hat{U}_{0.5} - \hat{B}]\hat{V}\hat{U}_{0.5}^{-2}\hat{B}\hat{A}^T\mathbf{y}, \tag{3.12}$$

where $\hat{B} = (\hat{A}^T\hat{A} + \hat{U}_{0.5}^{-1})^{-1}$. Again, $\hat{\mathbf{v}}$ is a vector whereas $\hat{V}$ is a matrix having the same

structure as $\hat{U}_{0.5}$. It was shown (see the proof of the Proposition 3.1.1) that the matrix $(\hat{u}_{0.5} - \hat{B})$ is positive definite and, therefore,

$$\hat{V}\hat{U}_{0.5}^{-2}\hat{B}\hat{A}^T\mathbf{y} = 0. \tag{3.13}$$

Since $\hat{\mathbf{U}}_{0.5}$ is the solution of (3.11) the Kuhn-Tucker conditions hold at this point

$$\mathbf{y}^T\hat{A}(\hat{A}^T\hat{A} + \hat{U}_{0.5}^{-1})^{-1}I_i\frac{1}{\hat{U}_{0.5i}^2}(\hat{A}^T\hat{A} + \hat{U}_{0.5}^{-1})^{-1}\hat{A}^T\mathbf{y} = \xi, \quad i = 1, \ldots, \hat{d}. \tag{3.14}$$

Note that, since none of $u_{0.5i}$ is equal to zero, the Lagrange multipliers corresponding to the positivity constraints in (3.11) are equal to zero and, therefore, do not appear in the equations (3.14). As $\hat{\mathbf{v}} \neq 0$, it follows from (3.13) and (3.14) that ξ is equal to zero and

$$\hat{B}\hat{A}^T\mathbf{y} = 0,$$

which is impossible since the matrix $\hat{B}$ is nonsingular and, by assumption, the vector $\hat{A}^T\mathbf{y}$ is nonzero. Thus the assumption that (3.10) has more than one solution led us to a contradiction and, therefore, the statement of the Lemma holds. □

The second result relates solutions of (3.1) and (3.10) when $\alpha \to 0$.

Lemma 3.2.2 *If $t'(\alpha) \to t'_0$ as $\alpha \to 0$, the solution of the problem*

$$\begin{aligned} \underset{\mathbf{u}}{\text{minimize}} \quad & -\mathbf{y}^T A(A^TA + U^{-1})^{-1}A^T\mathbf{y} + \sum_{i=1}^{d}\frac{\alpha}{u_i} \\ \text{subject to} \quad & \sum_{i=1}^{d} u_i \leq t'(\alpha), \\ & u_i \geq 0, \quad i = 1, \ldots, d, \end{aligned} \tag{3.15}$$

as well as the Lagrange multiplier $\xi(\alpha)$ corresponding to the constraint $\sum_{i=1}^{d} u_i \leq t'(\alpha)$

converge to the solution and the Lagrange multiplier respectively of (3.10) with $t' = t'_0$.

Proof. Since all solutions $u_i(\alpha)$, $i = 1, \ldots, d$ of (3.15) are contained in the compact region and (3.10) has a unique solution, it follows that, in order to prove the statement of the Lemma, it suffices to show that the limit $\mathbf{u}_0$ of any convergent sequence of solutions of (3.15) $\mathbf{u}_k = \mathbf{u}(\alpha_k) \to \mathbf{u}_0$, $k = 1, 2, \ldots$, where $\alpha_k \to 0$ as $k \to \infty$, is the solution of (3.10) with $t' = t'_0$. Note that $\sum u_{0i} = t'_0$. Assume that the converse is true, that is, $\mathbf{u}_0$ is not a solution of (3.10) and let $\mathbf{u}_s$ be a solution of (3.10). Now, $f(\mathbf{u}_s) < f(\mathbf{u}_0)$, where f is the objective function of the problem (3.10). By continuity, there exist a vector $\tilde{\mathbf{u}}$ in the vicinity of the solution $\mathbf{u}_s$ and positive constants δ and K_1 such that all components of $\tilde{\mathbf{u}}$ are strictly positive, $\sum \tilde{u}_i \leq t'(\alpha_k)$ for $k > K_1$ and $f(\mathbf{u}_k) - f(\tilde{\mathbf{u}}) > \delta$, $k > K_1$. Since all of $\tilde{u}_i$ are positive there exists a constant K_2 such that

$$\sum_{i=1}^{d} \frac{\alpha_k}{\tilde{u}_i} < \frac{\delta}{3}, \quad k > K_2.$$

Therefore, for all $k > \max(K_1, K_2)$ we have

$$f(\mathbf{u}_k) + \sum_{i=1}^{d} \frac{\alpha_k}{u_i(\alpha_k)} > f(\tilde{\mathbf{u}}) + \sum_{i=1}^{d} \frac{\alpha_k}{\tilde{u}_i}.$$

This inequality contradicts to the fact that $\mathbf{u}_k$ is the solution of the optimization problem (3.15) with $t'(\alpha_k)$. Thus, the initial assumption that $\mathbf{u}_0 \neq \mathbf{u}_s$ is wrong and, therefore, the statement of the Lemma holds.

Having proved that $\mathbf{u}(\alpha) \to \mathbf{u}_s$ as $\alpha \to 0$, one can prove that $\xi(\alpha) \to \xi_s$, where $\xi(\alpha)$ and ξ_s are Lagrange multipliers corresponding to the constraints $\sum u_i \leq t(\alpha)'$ and $\sum u_i \leq t'_0$ respectively. Indeed, let u_{si_0} be a nonzero component of $\mathbf{u}_s$. As $u_{i_0}(\alpha) \to u_{si_0}$, there exists an $\alpha^* > 0$ such that $u_{i_0}(\alpha) > 0$ for all $\alpha < \alpha^*$. Thus, the i_0-th equation of the Kuhn-Tucker conditions for the problem (3.15) is as follows:

$$\mathbf{y}^T A(A^T A + U^{-1}(\alpha))^{-1} \frac{I_{i_0}}{u^2_{i_0}(\alpha)} (A^T A + U^{-1}(\alpha))^{-1} A^T \mathbf{y} + \frac{\alpha}{u^2_{i_0}(\alpha)} = \xi(\alpha). \tag{3.16}$$

Analogously, the i_0-th equation of the Kuhn-Tucker conditions for the problem (3.10) is:

$$\mathbf{y}^T A(A^T A + U_s^{-1})^{-1} \frac{I_{i_0}}{u_{si_0}^2} (A^T A + U_s^{-1})^{-1} A^T \mathbf{y} = \xi_s. \tag{3.17}$$

As $\mathbf{u}(\alpha) \to \mathbf{u}_s$, the left-hand side of the equation (3.16) converges to the left-hand side of the equation (3.17). From this it immediately follows that $\xi(\alpha) \to \xi_s$ as $\alpha \to 0$. □

Remark. Considering that the constraint $\sum u_i \leq t'$ is always active (see Remark on the page 35), the above Lemma implies that the same constraint is always active in (3.10) as well.

Thus, the existence as well as uniqueness of the solution of the optimization problem (3.10) have been established. Also, solutions of (3.10) have been related to solutions of the problem (3.1). The next technical result states that the solution of (3.10) is a continuous function of the parameter $t' \in (0, \infty)$.

Lemma 3.2.3 *Let $u(t')$, $i = 1, ..., d$ and $\xi(t')$ be the solution and the Lagrange multiplier (corresponding to the inequality constraint $\sum u_i \leq t'$) respectively of the optimization problem (3.10). Then, both $u(t')$, $i = 1, ..., d$ and $\xi(t')$ are continuous functions of the parameter t'.*

Proof. The continuity of the components $u(t')$, $i = 1, ..., d$ can be proved by a straightforward adaptation of the approach used in the proof of Proposition 2.3.1. The continuity of $\xi(t')$ at any point $t'_0 \in (0, \infty)$ can be demonstrated as follows. Let $u_{i_0}(t'_0)$ be a nonzero component of the solution of (3.10). Due to continuity of the function $u_{i_0}(t')$, one deduces that $u_{i_0}(t')$ is nonzero in some neighbourhood Ω of the point t'_0. Therefore, for every point $t' \in \Omega$, the i_0-th equation of the Kuhn-Tucker conditions takes the form

$$\mathbf{y}^T A(A^T A + U^{-1}(t'))^{-1} \frac{I_{i_0}}{u_{i_0}^2(t')} (A^T A + U^{-1}(t'))^{-1} A^T \mathbf{y} = \xi(t'). \tag{3.18}$$

Note that, according to the complementarity condition $u_{i_0}\sigma_{i_0} = 0$, the Lagrange multiplier $\sigma_{i_0}(t')$ corresponding to the positivity constraint $u_{i_0} \geq 0$ is zero for $t' \in \Omega$ as the component $u_{i_0}(t')$ is assumed to be nonzero. Now, due to the continuity of the left-hand side of the

equation (3.18), $\xi(t')$ is a continuous function of $t' \in \Omega$ as well. Since the point t'_0 was an arbitrary point from the range $(0, \infty)$, we conclude that $\xi(t')$ is a continuous function over the whole interval $(0, \infty)$. □

The next step will be to prove the main result of this section: the optimization problem (3.10) can be solved instead of the original PLASM (2.7). Let us introduce two sets of estimates of regression coefficients. The first one is defined as

$$\mathcal{B}_1 = \{\beta(t) : t \in (0, t_r)\}.$$

Here $\beta(t)$ is the solution of the PLASM optimization problem (2.7) for a given value of the parameter t, where t_r is defined in (2.8). The second set is

$$\mathcal{B}_2 = \{\tilde{\beta}(t') : t' \in (0, \infty)\},$$

where

$$\tilde{\beta}(t') = (A^T A + U^{-1}(t'))^{-1} A^T \mathbf{y} = U(t')(A^T A U(t') + I)^{-1} A^T \mathbf{y}, \tag{3.19}$$

$u_i(t')$, $i = 1, \ldots, d$ being a solution of (3.10).

Proposition 3.2.1 *The sets $\mathcal{B}_1$ and $\mathcal{B}_2$ are identical.*

Proof. Let $t \in (0, t_r)$ and $\beta(t)$ be the corresponding solution of PLASM (2.7). Then, there exists α^* such that $t \in (t_l^\alpha, t_r^\alpha)$ for $\alpha < \alpha^*$, where t_l^α, t_r^α are defined by (2.14). Let $v_i^2(t, \alpha)$, $i = 1, \ldots, d$, $\lambda(t, \alpha)$ be a solution of the system (2.21) with the same value for the parameter t and $\alpha < \alpha^*$. Then, due to Proposition 3.1.2 and formula (2.22),

$$\begin{aligned} u_i(t'_\alpha) &= v_i^2(t, \alpha), \quad i = 1, \ldots, d, \\ \xi(t'_\alpha) &= \frac{\lambda(t, \alpha)^2}{4}, \end{aligned}$$

are the solution and the Lagrange multiplier respectively of the optimization problem (3.1) with $t'_\alpha = 2t/\lambda(t,\alpha)$ and

$$\beta(t,\alpha) = (A^TA + U^{-1}(t'_\alpha))^{-1}A^T\mathbf{y} = U(t'_\alpha)(A^TAU(t'_\alpha) + I)^{-1}A^T\mathbf{y} \tag{3.20}$$

is the solution of the regularized PLASM (2.12) with the above values for the parameters t and α. Note that, according to the Lemma 2.3.1, $\lambda(t,\alpha) \to \lambda(t,0) > 0$ and, consequently, $t'_\alpha \to t'_0 = 2t/\lambda(t,0)$, $t'_0 \in (0,\infty)$. Thus, by Lemma 3.2.2:

$$\begin{aligned} \mathbf{u}(t'_\alpha) &\to \mathbf{u}(t'_0), \\ \xi(t'_\alpha) &\to \xi(t'_0), \\ t'_\alpha &\to t'_0, \end{aligned}$$

as $\alpha \to 0$. Here, $\mathbf{u}(t'_0)$ is the solution of (3.10) with $t' = t'_0$ and $\xi(t'_0)$ is the Lagrange multiplier corresponding to the constraint $\sum u_{\leq} t'_0$. Also, note that

$$\xi(t'_0) = \frac{\lambda(t,0)^2}{4}. \tag{3.21}$$

So, $\beta(t,\alpha) \to \tilde{\beta}(t'_0) = U(t'_0)(A^TAU(t'_0) + I)^{-1}A^T\mathbf{y}$. On the other hand, by Proposition 2.3.1, $\beta(t,\alpha)$ defined in (3.20) converges to the solution of the original PLASM $\beta(t)$ as $\alpha \to 0$. Thus, we conclude that

$$\beta(t) = \tilde{\beta}(t'_0),$$

and $\beta(t)$ belongs to $\mathcal{B}_2$ which proves that $\mathcal{B}_1 \subset \mathcal{B}_2$. Note that the mapping $(0,t_r) \to (0,\infty)$ defined by

$$t'(t) = 2t/\lambda(t,0) \tag{3.22}$$

maps any two distinct t_1 and t_2 onto the distinct t_1' and t_2'. Indeed, assume the converse: there exist $t_1 \neq t_2$ such that $t_1' = t'(t_1) = t_2' = t'(t_2)$. Then, $\xi(t_1') = \xi(t_2')$, and it follows from (3.21) that $\lambda(t_1, 0) = \lambda(t_2, 0)$. According to (3.22), this implies that $t_1 = t_2$ which contradicts to the assumption that $t_1 \neq t_2$. Thus, distinct values for t are mapped onto the distinct values of t'.

Now we will show that $\mathcal{B}_2 \subset \mathcal{B}_1$ also holds. Let $t' \in (0, \infty)$ and $u_i(t')$, $i = 1, \ldots, d$, $\xi(t')$ be the solution and the Lagrange multiplier of (3.10) and

$$\tilde{\beta}(t') = U(t')(A^T A U(t') + I)^{-1} A^T \mathbf{y}$$

be a vector of the regression coefficients $\tilde{\beta}(t') \in \mathcal{B}_2$. Also, let $u_i(t', \alpha)$, $i = 1, \ldots, d$, $\xi(t', \alpha)$ be the solution and Lagrange multiplier of (3.1) for the same value of the parameter t' and some small value for the parameter α. According to the Lemma 3.2.2, we have

$$\begin{aligned} \mathbf{u}(t', \alpha) &\rightarrow \mathbf{u}(t', 0) = \mathbf{u}(t'), \\ \xi(t', \alpha) &\rightarrow \xi(t', 0) = \xi(t'), \end{aligned} \tag{3.23}$$

as $\alpha \rightarrow 0$. Now, by Proposition 3.1.2 and equation (2.22), the vector

$$\beta(t', \alpha) = \beta(t_\alpha) = U(t', \alpha)(A^T A U(t', \alpha) + I)^{-1} A^T \mathbf{y}$$

is a solution of the regularized PLASM (2.12) with $t_\alpha = t'\sqrt{\xi(t', \alpha)}$. We have that

$$\beta(t', \alpha) \rightarrow \beta(t', 0) = \beta(t_0) = \beta(t'), \quad \text{as} \quad \alpha \rightarrow 0,$$

where $\beta(t_0)$ is a solution of the original PLASM (2.7) with $t_0 = t'\sqrt{\xi(t', 0)}$. Note that $t_0 \in (0, t_r)$ since the $u_i(t')$, $i = 1, \ldots, d$ are finite and at least some are nonzero. Thus we conclude that $\beta(t') \in \mathcal{B}_1$ and $\mathcal{B}_2 \subset \mathcal{B}_1$. If $t_1' \neq t_2'$, then $t_1 \neq t_2$, where $t_i = t(t_i') = t_i'\sqrt{\xi(t_i', 0)}$, $i = 1, 2$. Indeed, assume that the converse holds: $t(t_1') = t(t_2')$. Using the same kind of argument as was utilized to derive the formula (3.21) one can conclude that

$\xi(t'_1) = \xi(t'_2)$. This in turn implies that $t'_1 = t'_2$ as $t = t'\sqrt{\xi(t')}$. The obtained contradiction proves that $t(t') : (0, \infty) \to (0, t_r)$ maps different t' onto different t.

In summary, we have showed that the sets $\mathcal{B}_1$ and $\mathcal{B}_2$ introduced above are identical. In fact, the identical members of these sets can be identified via relationship between the parameter t of the original PLASM (2.7) and the parameter t' of the optimization problem (3.10). This relationship is a one-to-one correspondence between the respective intervals ($t \in (0, t_r)$ and $t' \in (0, \infty)$) and can be expressed as follows:

$$t = t'\sqrt{\xi(t')}$$

or, recognizing the equality (3.21):

$$t' = \frac{2t}{\lambda(t)},$$

where the quantities $\xi(t')$ and $\lambda(t)$ were introduced above. Thus, $\mathcal{B}_1 = \{\beta(t) : t \in (0, t_r)\} = \{\tilde{\beta}(2t/\lambda(t)) : t \in (0, t_r)\} = \{\beta(t'\sqrt{\xi(t')}) : t' \in (0, \infty)\} = \{\tilde{\beta}(t') : t' \in (0, \infty)\} = \mathcal{B}_2$. □

The next Proposition develops some of the properties of the function $t'(t) = 2t/\lambda(t)$, $t \in (0, t_r)$.

Proposition 3.2.2 *Given that all of the components of the vector $A^T\mathbf{y}$ are distinct from zero, the function $t' = 2t/\lambda(t)$, $t \in (0, t_r)$ is a continuous, monotonic function such that $\lim_{t\to 0} t'(t) = 0$ and $\lim_{t\to t_r} t'(t) = \infty$. Here t_r is defined in (2.8), $\lambda(t) = \lim_{\alpha\to 0} \lambda(t, \alpha)$, and $\lambda(t, \alpha)$, $\xi(t')$ are the Lagrange multipliers of the problems (2.12) and (3.10) respectively.*

Proof. We established in the previous Proposition that $t'^{-1}(t) = t(t') = t'\sqrt{\xi(t')}$ and both $t'(t)$ and $t(t')$ define a one-to-one correspondence between the intervals $(0, t_r)$ and $(0, \infty)$. Now, the continuity of $t(t')$ follows immediately from the continuity of $\xi(t')$ which was demonstrated in the Lemma 3.2.3. This implies that $t(t')$ is a monotonic function and, therefore, $t'(t)$ is continuous and monotonic as well. Our next step will be to prove that $\lim_{t\to 0} t'(t) = 0$. This can be done by showing that $1/\lambda(t)$ remains bounded from above as $t \to 0$. We have that $\lambda(t) = \lim_{\alpha\to 0} \lambda(t, \alpha)$, where $\lambda(t, \alpha)$ is a Lagrange multiplier of the regularized PLASM (2.12). Note that $\lambda(t)$ is not a Lagrange multiplier of the original

PLASM (2.7) since, as was shown in section 2.2, the Kuhn-Tucker conditions may not hold for that optimization problem. Now, the Kuhn-Tucker conditions for the problem (2.12) are:

$$\mathbf{a}_{ij}^T(A\beta(t,\alpha)-\mathbf{y})+\frac{1}{2}\frac{\lambda(t,\alpha)\beta_{ij}(t,\alpha)}{[\beta_i(t,\alpha)^T\beta_i(t,\alpha)+\alpha]^{\frac{1}{2}}}=0,\quad j=1,...,p_i,\ i=1,...,d,$$

where $\beta(t,\alpha)$ is the solution of (2.12) and $\mathbf{a}_{ij}$ is a (ij)-th column of the matrix A. Therefore, $1/\lambda(t,\alpha)$ can be expressed as

$$1/\lambda(t,\alpha)=-\frac{\beta_{i_0 1}(t,\alpha)}{2[\beta_{i_0}(t,\alpha)^T\beta_{i_0}(t,\alpha)+\alpha]^{\frac{1}{2}}}\cdot\frac{1}{\mathbf{a}_{i_0 1}^T(A\beta(t,\alpha)-\mathbf{y})},$$

where i_0 is the number of the group of regression coefficients such that $\beta_{i_0}(t,\alpha)^T\beta_{i_0}(t,\alpha)\to\beta_{i_0}(t)^T\beta_{i_0}(t)\neq 0$, where $\beta_{i_0}(t)$ is the i_0-th group of regression coefficients as estimated by the original PLASM (2.7). It is worth mentioning that, generally, i_0 is a function of the parameter t. Thus,

$$1/\lambda(t)=-\frac{\beta_{i_0 1}(t)}{2[\beta_{i_0}(t)^T\beta_{i_0}(t)]^{\frac{1}{2}}}\cdot\frac{1}{\mathbf{a}_{i_0 1}^T(A\beta(t)-\mathbf{y})}.$$

Note that $\beta_{i_0 1}(t)/[\beta_{i_0}(t)^T\beta_{i_0}(t)]^{\frac{1}{2}}\leq 1$ for $t\in(0,t_r)$ and $A^T(A\beta(t)-\mathbf{y})\to -A^T\mathbf{y}$ as $t\to 0$. Remembering that, by assumption, none of the components of the vector $A^T\mathbf{y}$ is equal to zero we conclude that

$$1/\lambda(t)\leq\frac{1}{\min_{ij}|\mathbf{a}_{ij}^T\mathbf{y}|}$$

for small t. Therefore, $\lim_{t\to 0}2t/\lambda(t)=0$. This implies that $t'(t)=2t/\lambda(t)$ is an increasing function and, due to the fact that it maps the interval $(0,t_r)$ onto the interval $(0,\infty)$, $\lim_{t\to t_r}t'(t)=\infty$. □

Now we will summarize what we have achieved in this chapter. We started with consideration of the PLASM formulation:

$$\beta(t)=\text{argmin }(\mathbf{y}-A\beta)^T(\mathbf{y}-A\beta)$$

$$\text{subject to } \sum_{i=1}^{d}[\beta_i^T\beta_i]^{\frac{1}{2}} \leq t,$$

$$\text{where } t \in (0, t_r), \quad t_r = \sum_{i=1}^{d}[\beta_i^{T^o}\beta_i^o]^{\frac{1}{2}}, \tag{3.24}$$

which defines the set of estimates of regression coefficients $\mathcal{B}_1 = \{\beta(t) : t \in (0, t_r)\}$. In order to produce the best model, one has to minimize an estimate of the future predictive error of the model determined by $\beta(t)$ over this set. The main result of our investigation is that $\mathcal{B}_1$ can be parameterized in a different way $\mathcal{B}_1 = \{\tilde{\beta}(t') : t' \in (0, \infty)\}$ based on the following formulation of PLASM:

$$\tilde{\beta}(t') = U(t')(A^TAU(t') + I)^{-1}A^T\mathbf{y},$$

$$\begin{aligned}\mathbf{u}(t') = \text{argmin} \quad & -\mathbf{y}^TA(A^TA + U^{-1})^{-1}A^T\mathbf{y} \\ \text{subject to} \quad & \sum_{i=1}^{d} u_i \leq t', \\ & u_i \geq 0, \quad i = 1, \ldots, d,\end{aligned}$$

$$\text{where } t' \in (0, \infty). \tag{3.25}$$

Thus, either of these formulations can be used to produce the set of the PLASM estimates of regression coefficients $\mathcal{B}_1$. However, there are two advantages of using (3.25) instead of (3.24). First, as can be seen, the second formulation requires the solution of an optimization problem involving as many unknowns, d, as there are groups of regression coefficients. This is in contrast with the first formulation where the parameters to be determined are regression coefficients and there are $P = d \cdot \bar{p}$ of them, where $\bar{p}$ is an average number of regression coefficients per group. So, the dimensionality of the optimization problem in (3.25) is, generally, much lower than that of (3.24). The second advantage is that formulation (3.25) has a simpler structure in the sense that it involves only a simple linear constraint as well as positivity constraints as opposed to the nonlinear constraint of the first formulation.

In this chapter we confined ourselves to consideration of additive models. However, the similar approach can be used to build more complex models involving interactions between variables where all basis functions modelling interaction between particular variables would have to be combined into a group.

Finally, the new formulation of PLASM sheds some light on the properties of the PLASM solutions and these issues will be considered in the next chapter.

Chapter 4
Investigation of PLASM & Numerical Issues

In the previous two chapters we introduced the Probing Least Absolute Squares Modelling approach (2.7) and derived an alternative formulation of PLASM (3.25) which provides a number of advantages over the previous one. In this chapter the new formulation (3.25) is used to obtain a deeper understanding of the nature of PLASM. Our study begins with consideration of the special situation called the Orthogonal Design case where analytical solution of the optimization problem in (3.25) is possible.

4.1 The Orthogonal Design Case

The Orthogonal Design case corresponds to the situation where $A^TA = I$. Given this, the optimization problem in (3.25) takes the form

$$\begin{aligned} \underset{\mathbf{u}}{\text{minimize}} \quad & -\mathbf{y}^T A(I + U^{-1})^{-1} A^T \mathbf{y} \\ \text{subject to} \quad & \sum_{i=1}^{d} u_i \leq t', \\ & u_i \geq 0, \quad i = 1, \ldots, d, \end{aligned} \tag{4.1}$$

and the corresponding Kuhn-Tucker necessary conditions are as follows:

$$\mathbf{y}^T A(I + U^{-1})^{-1} \frac{1}{u_i^2} I_i (I + U^{-1})^{-1} A^T \mathbf{y} + \sigma_i = \xi, \quad i = 1, \ldots, d,$$

$$\begin{aligned} u_i &\geq 0, \quad i = 1, \ldots, d, \\ \sigma_i &\geq 0, \quad i = 1, \ldots, d, \\ u_i \sigma_i &= 0, \quad i = 1, \ldots, d, \\ \sum_{i=1}^{d} u_i &= t', \ \xi \geq 0, \end{aligned} \tag{4.2}$$

where ξ and $\sigma_i,\ i = 1, \ldots, d$ are Lagrange multipliers. These equations can be recast as

$$\begin{aligned} \frac{B_i^2}{(1+u_i)^2} + \sigma_i &= \xi, \quad i = 1, \ldots, d, \\ u_i &\geq 0, \quad i = 1, \ldots, d, \\ \sigma_i &\geq 0, \quad i = 1, \ldots, d, \\ u_i \sigma_i &= 0, \quad i = 1, \ldots, d, \\ \sum_{i=1}^{d} u_i &= t', \ \xi \geq 0, \end{aligned} \tag{4.3}$$

where $B_i^2 = \mathbf{y}^T A I_i A^T \mathbf{y},\ i = 1, \ldots, d$ which we will call *group* coefficients in our future considerations. Examination of the above equations indicate that a component u_i is distinct from zero if and only if the corresponding group coefficient B_i^2 is greater than ξ. Taking this observation into account and given the value for the parameter ξ, we can write the solution of the system (4.3) for quantities $u_i,\ i = 1, \ldots, d$:

$$u_i = \begin{cases} B_i/\xi^{\frac{1}{2}} - 1, & B_i^2 > \xi, \\ 0, & B_i^2 \leq \xi, \end{cases}$$

and the Lagrange multipliers $\sigma_i,\ i = 1, \ldots, d$:

$$\sigma_i = \begin{cases} 0, & B_i^2 > \xi, \\ \xi - B_i^2, & B_i^2 \leq \xi, \end{cases}$$

The value for the parameter ξ is chosen such as to satisfy the equality constraint $\sum_{i=1}^{d} u_i =$

t'. This is possible since the sum $\sum u_i$ is a continuous function of the parameter ξ and it varies from 0 to ∞ as the value of ξ varies from ∞ to 0. Given u_i's, the regression coefficients β can be computed according to the formula in (3.25):

$$\beta_{ij} = \begin{cases} [1 - \xi^{\frac{1}{2}}/B_i]\beta^o_{ij}, & B_i^2 > \xi, \\ 0, & B_i^2 \leq \xi, \end{cases}$$

where β^o_{ij}, $j = 1, \ldots, p_i$, $i = 1, \ldots, d$ are components of the vector of the unconstrained least squares regression coefficients $A^T\mathbf{y}$. As can be seen, the groups of unconstrained coefficients exhibiting the strongest effects (determined by the group coefficients β_i, $i = 1, ..., d$) are shrunk and retained in the model while the other groups are set to zero. This result supports our earlier conjecture (see page 23) that PLASM sets some of the groups of coefficients to zero while the others are estimated in the way similar to that of the ridge regression.

4.2 Highly Constrained PLASM Solutions

In this section we will investigate another special situation corresponding the small values for the parameter t' which implies that the estimates of the regression coefficients will be highly influenced by the constraint. It turns out that, in this situation, one is able to obtain an (approximate) analytical solution of PLASM (3.25). The following results holds:

Proposition 4.2.1 *Assume that the group coefficients $B_i^2 = \mathbf{y}^T A I_i A^T \mathbf{y}$, $i = 1, ..., d$ are all distinct. Then, if t' is small enough, there is only one component distinct from zero in the solution of the optimization problem in (3.25).*

Proof. Since t' is small, u_i, $i = 1, \ldots, d$ are small too and, therefore, one can expand the objective function in (3.25) with respect to $\mathbf{u}$ and omit all powers of u_i, $i = 1, ..., d$ of the second and higher orders:

$$-\mathbf{y}^T A(A^T A + U^{-1})^{-1} A^T \mathbf{y} \approx -\sum_{i=1}^{d} B_i^2 u_i.$$

The optimization problem becomes:

$$\begin{aligned} \underset{\mathbf{u}}{\text{minimize}} \quad & -\sum_{i=1}^{d} B_i^2 u_i \\ \text{subject to} \quad & \sum_{i=1}^{d} u_i \leq t', \\ & u_i \geq 0, \quad i = 1, \ldots, d. \end{aligned} \tag{4.4}$$

The Kuhn-Tucker conditions for this problem take the form:

$$\begin{aligned} -B_i^2 + \xi - \sigma_i &= 0, \quad i = 1, ..., d \\ u_i \sigma_i &= 0, \quad i = 1, ..., d \\ (\sum_{i=1}^{d} u_i - t')\xi &= 0, \end{aligned} \tag{4.5}$$

where $\xi \geq 0$ and $\sigma_i \geq 0$, $i = 1, ..., d$ are Lagrange multipliers. We claim that this system has only one solution expressed by the formula:

$$\begin{aligned} u_{i_0} &= t', \\ u_i &= 0, \quad i \neq i_0, \end{aligned} \tag{4.6}$$

where $i_0 = \text{argmin}_{i=1,...,d} B_i^2$. We will justify this claim in two steps. First, we will show than no more than one component u_i can be distinct from zero in the solution of (4.5). Indeed, assume that there are two or more non-zero components: $u_{i_1} > 0$ and $u_{i_2} > 0$. According to (4.5), $\sigma_{i_1} = 0$ and $\sigma_{i_2} = 0$ and, consequently,

$$\begin{aligned} -B_{i_1}^2 + \xi &= 0, \\ -B_{i_2}^2 + \xi &= 0. \end{aligned}$$

This pair of equalities implies that $B_{i_1}^2 = B_{i_2}^2$ which contradicts the condition in the

statement of the Proposition that all B_i^2, $i = 1, \ldots, d$ are distinct.

Finally, we will demonstrate that only u_{i_0} can be distinct from zero, where $i_0 = \text{argmin } B_i^2$. Assume that the converse holds: $u_{i_1} > 0$, $i_1 \neq i_0$. The following two equalities follow immediately from (4.5):

$$\begin{aligned} -B_{i_0}^2 + \xi - \sigma_{i_0} &= 0, \\ -B_{i_1}^2 + \xi &= 0. \end{aligned}$$

Here $\sigma_{i_0} \geq 0$. It follows from the second equation that $\xi = B_{i_1}^2$ and the first equation yields $-B_{i_0}^2 + B_{i_1}^2 - \sigma_{i_0} = 0$ which is impossible since $i_0 = \text{argmin } B_i^2$. Therefore, (4.6) is indeed the only solution of (4.5).

Given the solution (4.6), it follows from the formula for the regression coefficients β in (3.25) that β_{i_0} is the only group distinct from zero:

$$\beta_{i_0} = (A_{i_0}^T A_{i_0} + t'^{-1} I)^{-1} A_{i_0}^T \mathbf{y},$$

where A_{i_0} is an appropriate submatrix of the matrix A. □

So, the Proposition above shows that for t' small enough, the PLASM solution contains only one non-zero group of the coefficients. As t' varies from 0 to ∞ the number of groups present in the model will, generally, vary from 1 to d, since PLASM solutions approach unconstrained least squares solutions as $t' \to \infty$. Therefore, intermediate values for t' are likely to result in selection of models containing only a few groups.

4.3 PLASM as a GCV Minimizer

In the previous sections we provided some evidence that, given the value for the parameter t', PLASM does perform model selection. However, the question is: is the selected model optimal in terms of the level of the Prediction Error (1.4)? In this section it is shown that PLASM solutions minimize an estimate of the Prediction Error called Generalized

Cross-Validation score (1.7)

$$\mathrm{GCV} = \frac{1}{N} \cdot \frac{\mathrm{RSS}}{[1 - \mathrm{df}/N]^2},$$

where RSS is the residual sum of squares for a model and df is the number of degrees of freedom used to fit the model. In our case, RSS equals to

$$\mathrm{RSS}(\mathbf{u}) = (\mathbf{y} - A\beta)^T(\mathbf{y} - A\beta), \quad \beta = (A^TA + U^{-1})^{-1}A^T\mathbf{y}, \tag{4.7}$$

and the number of degrees of freedom can be estimated as [55]

$$\mathrm{df}(\mathbf{u}) = \mathrm{tr}(H)(\mathbf{u}) = \mathrm{tr}(A(A^TA + U^{-1})^{-1}A^T), \tag{4.8}$$

where $H = A(A^TA + U^{-1})^{-1}A^T$ is called a *hat* matrix. In the above definitions we emphasize the dependence of RSS and df on the quantities u_i, $i = 1, \ldots, d$ for reasons that will become clear shortly.

In this section we will consider an alternative method of estimating the regression coefficients which differs from the PLASM approach (3.25) in that one obtains u_i, $i = 1, \ldots, d$ via minimization of the Generalized Cross-Validation score instead of the objective function $-\mathbf{y}^TA(A^TA + U^{-1})^{-1}A^T\mathbf{y}$. Thus, this approach can be formulated as follows

$$\begin{aligned}
\hat{\beta}(t') &= U(t')(A^TAU(t') + I)^{-1}A^T\mathbf{y}, \\
\mathbf{u}(t') &= \operatorname{argmin} \frac{1}{N} \cdot \frac{\mathrm{RSS}(\mathbf{u})}{[1 - \mathrm{tr}(H)(\mathbf{u})/N]^2} \\
&\quad \text{subject to } \sum_{i=1}^{d} u_i = t', \\
&\quad u_i \geq 0, \quad i = 1, \ldots, d,
\end{aligned} \tag{4.9}$$

where $t' \in (0, \infty)$. Here $\mathrm{tr}(H)(\mathbf{u})$ and $RSS(\mathbf{u})$ are defined in (4.8) and (4.7) respectively. The following Proposition relates estimates of the regression coefficients obtained via PLASM (3.25) with those obtained according to (4.9).

Proposition 4.3.1 *Given that t' is small and the matrix A is normalized in such a way that $\gamma_i = \sum_{n=1}^{N} \sum_{j=1}^{p_i} a_{n(ij)}^2$, $i = 1, \ldots, d$ are all identical, the regression coefficients $\hat{\beta}$ obtained via (4.9) are approximately equal to β obtained by (3.25).*

Remark. Note that the optimization problem in (4.9) involves equality constraint $\sum u_i = t'$ as opposed to the inequality constrained $\sum u_i \leq t'$ in PLASM (3.25). However, as was shown earlier (see Remark on the page 41), the constraint $\sum u_i \leq t'$ in (3.25) is always active and, therefore, it can be replaced with the equality constraint.

Proof. Because the formulas for estimating regression coefficients are the same in both cases, it suffices to show that solutions of the respective optimization problems for u_i, $i = 1, \ldots, d$ are approximately equal. To show this, we will demonstrate that, for small values for the parameter t', the objective functions of the optimization problems are approximately proportional.

First, we will consider the objective function of the optimization problem in (4.9). The numerator of the function (RSS) can be transformed as follows

$$\begin{aligned} \mathrm{RSS}(\mathbf{u}) &= \mathbf{c}^T A^T A \mathbf{c} + \mathbf{y}^T \mathbf{y} - 2\mathbf{c}^T A^T \mathbf{y} = \\ & -2\mathbf{c}^T A^T \mathbf{y} + \mathbf{c}^T (A^T A + U^{-1} - U^{-1})\mathbf{c} + \\ & \mathbf{y}^T \mathbf{y} = -\mathbf{c}^T A^T \mathbf{y} + \mathbf{c}^T U^{-1} \mathbf{c} + \mathbf{y}^T \mathbf{y} \end{aligned} \tag{4.10}$$

where $\mathbf{c} = (A^T A + U^{-1})^{-1} A^T \mathbf{y}$. Taking into account that the parameter t' is small, one can expand this expression in powers of $\mathbf{u}$ and neglect terms of the second order and higher: $\mathrm{RSS} \approx -2\mathbf{y}^T A U A^T \mathbf{y} + \mathbf{y}^T \mathbf{y}$. The trace of the hat matrix in the denominator of the objective function in (4.9) can be transformed in the same way:

$$\mathrm{tr}(A(A^T A + u^{-1})^{-1} A^T) \approx \mathrm{tr}(AUA^T) = \sum_{n=1}^{N} \sum_{i=1}^{d} u_i \sum_{j=1}^{p_i} a_{n(ij)}^2 = \sum_{i=1}^{d} u_i \sum_{n=1}^{N} \sum_{j=1}^{p_i} a_{n(ij)}^2 = \sum_{i=1}^{d} u_i \gamma_i,$$

where d is the number of groups of basis functions, p_i is the number basis functions in the i-th group, N is the size of the dataset and $\gamma_i = \sum_{n=1}^{N} \sum_{k=1}^{s_i} a_{n(ik)}^2$. According to the formulation of the Proposition, $\gamma_i,\ i = 1, \ldots, d$ are all identical. Therefore,

$$\mathrm{tr}(H) = \mathrm{const} \cdot t',$$

and, consequently, the trace is constant. Thus, we conclude that the objective function in (4.9) can be written as

$$\frac{1}{N} \cdot \frac{\mathrm{RSS}(\mathbf{u})}{[1 - \mathrm{tr}(H)(\mathbf{u})/N]^2} = \mathrm{const}_1(-\mathbf{y}^T AUA^T\mathbf{y}) + \mathrm{const}_2.$$

Similarly, for small values for the parameter t', the objective function of the optimization problem in (3.25) simplifies to $-\mathbf{y}^T AUA^T\mathbf{y}$.

Thus, for small values for the parameter t', the objective function of the optimization problem in (4.9) is approximately proportional to the objective function of the optimization problem in (3.25). Considering the Remark made after the formulation of this Proposition, one can conclude that the solutions $u_i,\ i = 1, \ldots, d$ of both optimizations problems are approximately equal. □

Thus, at least for the small values for the parameter t', estimation of the regression coefficients according to PLASM is approximately equivalent to minimization of the Generalized Cross-Validation criterion over the class of linear models whose regression coefficients are determined as $\hat{\beta}(t') = U(t')(A^T AU(t') + I)^{-1} A^T\mathbf{y}$ with quantities $u_i,\ i = 1, \ldots, d$ subject to the constraint $\sum u_i = t'$.

4.4 Bayesian Formulation of PLASM

It was shown in [55] that the LASSO estimates can be derived as a Bayesian posterior mode under certain prior distributions for the regression coefficients β. In this section, we will demonstrate that it is possible to derive PLASM using Bayesian approach as well though under different prior assumptions. First, we note that minimization of the residual sum of squares subject to the constraint $\sum_{i=1}^{d} [\beta_i^T \beta_i]^{\frac{1}{2}} \leq t$ is equivalent to the unconstrained

minimization of the following function:

$$\beta = \text{argmin } (\mathbf{y} - A\beta)^T(\mathbf{y} - A\beta) + \gamma \sum_{i=1}^{d} [\beta_i^T \beta_i]^{\frac{1}{2}}, \tag{4.11}$$

where γ is a parameter related in a certain way to the parameter t of the original PLASM. This reformulation of PLASM will allow us to derive PLASM estimates as a Bayes posterior mode under a special prior for β.

Let us consider the following linear regression model:

$$\mathbf{y} = A\beta + \epsilon,$$

where $\mathbf{y}$ is vector of response values, A is a model matrix, ϵ is an error vector with variance matrix $\sigma^2 I$, and β is a vector of regression coefficients. We assume that the coefficients β are split into d groups. We make the following prior assumptions:

1. Given the regression coefficients β, the distribution of the response values $\mathbf{y}$ is normal with the vector of the mean values $E(\mathbf{y}) = A\beta$ and the covariance matrix $\sigma^2 I$.

2. The prior distributions for different groups of β_i and β_j are independent and have the following form:

$$p(\beta_i) \sim \exp\{-\lambda[\beta_i^T \beta_i]^{\frac{1}{2}}\}, \ i = 1, \ldots, d. \tag{4.12}$$

The posterior distribution for the regression coefficients can be obtained using the Bayes formula:

$$p(\beta|\mathbf{y}) = \frac{p(\beta, \mathbf{y})}{p(\mathbf{y})} = \frac{p(\mathbf{y}|\beta)p(\beta)}{p(\mathbf{y})}.$$

The Bayes estimates of β is the maximizer of the posterior probability distribution function. Since the denominator in the above formula does not depend on β it can be dis-

regarded. Therefore, considering our assumptions concerning the prior distributions, we have:

$$p(\beta|\mathbf{y}) \sim \exp\{-\frac{1}{2\sigma^2}(\mathbf{y}-A\beta)^T(\mathbf{y}-A\beta)\} \cdot \prod_{i=1}^{d} \exp\{-\lambda[\beta_i^T\beta_i]^{\frac{1}{2}}\}.$$

Maximization of this function with respect to β is equivalent to minimization of $-\log(p(\beta|\mathbf{y}))$:

$$-\log(p(\beta|\mathbf{y})) \sim (\mathbf{y}-A\beta)^T(\mathbf{y}-A\beta) + \tilde{\lambda}\sum_{i=1}^{d}[\beta_i^T\beta_i]^{\frac{1}{2}},$$

where $\tilde{\lambda} = 2\sigma^2\lambda$. As can be seen this expression has the same form as (4.11) with $\gamma = \tilde{\lambda}$. This proves that PLASM estimates coincide with the Bayes posterior mode under the special prior for the regression coefficients (4.12).

4.5 Numerical Solution of PLASM

The rest of this chapter we will dedicate to consideration of various numerical issues associated with obtaining PLASM estimates according to (3.25). Basically, there are two problems to be solved: the first one is a numerical solution of the optimization problem in (3.25) for any given value for the parameter $t' \in (0, \infty)$ and this will become the subject of this section; the second problem is related to selection of the optimal value for the parameter t'. The latter issue will be discussed in the next section.

To obtain solutions of the PLASM optimization problem in (3.25) we utilized a form of the very well-known algorithm called Sequential Quadratic Programming (SQP) procedure proposed by M.J.D. Powell [48]. The basic idea of SQP is to construct a sequence of special quadratic optimization problems whose solutions converge to the solution of an original nonlinear problem. Since there is an extensive literature dedicated to various aspects and modifications of SQP (see, for instance [18]), only a brief outline of this algorithm is given. Consider the (general) nonlinear optimization problem:

$$\underset{\mathbf{x}}{\text{minimize}} \quad F(\mathbf{x})$$

$$\begin{aligned} \text{subject to} \quad & c_i(\mathbf{x}) = 0,\ i = 1, \ldots, d', \\ & c_i(\mathbf{x}) \leq 0,\ i = (d'+1), \ldots, d, \end{aligned} \tag{4.13}$$

where $F(\mathbf{x})$, $c_i(\mathbf{x})$, $i = 1, \ldots, d$ are smooth functions of d variables representing objective function and constrains of the optimization problem. The SQP procedure consists of the following steps:

1. Start with an initial point $\mathbf{x}_0$ and an initial $d \times d$ positive definite matrix B_0. Set $k = 0$.

2. Solve the following quadratic optimization problem for the vector $\mathbf{s}$ and determine the respective Lagrange multipliers λ_i, $i = 1, \ldots, d$

$$\begin{aligned} \underset{\mathbf{s}}{\text{minimize}} \quad & F_k + \mathbf{s}'\nabla F_k + \mathbf{s}' B_k \mathbf{s} \\ \text{subject to} \quad & c_{ik} + \mathbf{s}'\nabla c_{ik} = 0,\ i = 1, \ldots, d', \\ & c_{ik} + \mathbf{s}'\nabla c_{ik} \leq 0,\ i = (d'+1), \ldots, d, \end{aligned} \tag{4.14}$$

   where F_k, c_{ik}, ∇F_k and ∇c_{ik} are values and gradients respectively of the functions $F(\mathbf{x})$ and $c_i(\mathbf{x})$ evaluated at the current approximation $\mathbf{x}_k$. If the solution of this quadratic problem is $\mathbf{s} = 0$, terminate the process and $\mathbf{x}$ is the solution of the problem (4.13).

3. With $\mathbf{s}$ found above, perform a line search in the direction $\mathbf{s}$, using the so-called absolute value penalty function defined as:

$$\Psi(\mathbf{x}) = F(\mathbf{x}) + \sum_{i=1}^{d'} \mu_i |c_i(\mathbf{x})| + \sum_{i=d'+1}^{d} \mu_i |\min[0, c_i(\mathbf{x})]|, \tag{4.15}$$

   where $\mu_i = \max[|\lambda_i|, 0.5(\tilde{\mu}_i + |\lambda_i|)]$, $i = 1, \ldots, d$ and $\tilde{\mu}_i$, $i = 1, \ldots, d$ are the values for μ_i used on the previous iteration. The next approximation to the solution is determined as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{s}.$$

Here

$$\alpha_k = \min_{\alpha>0} \Psi(\mathbf{x}_k + \alpha \mathbf{s})$$

was found by the line search.

4. Update B_k according to

$$B_{k+1} = B_k - \frac{B_k \mathbf{p}_k \mathbf{p}_k' B_k}{\mathbf{p}_k' B_k \mathbf{p}_k} + \frac{\mathbf{r}_k \mathbf{r}_k'}{\mathbf{p}_k' \mathbf{r}_k},$$

where

$$\begin{aligned} \mathbf{p}_k &= \mathbf{x}_{k+1} - \mathbf{x}_k, \\ \mathbf{q}_k &= \nabla F_{k+1} - \nabla F_k + \sum_{i=1}^{p} \lambda_i (\nabla c_{i(k+1)} - \nabla c_{ik}), \\ \mathbf{r}_k &= \theta_k \mathbf{q}_k + (1 - \theta_k) B_k \mathbf{p}_k. \end{aligned}$$

The value of the parameter θ_k is determined according to the formula

$$\theta_k = \begin{cases} 1, & \text{if } \mathbf{p}_k' \mathbf{q}_k \geq 0.2 \mathbf{p}_k' B_k \mathbf{p}_k, \\ 0.8 \mathbf{p}_k' B_k \mathbf{p}_k / (\mathbf{p}_k' B_k \mathbf{p}_k - \mathbf{p}_k' \mathbf{q}_k), & \text{if } \mathbf{p}_k' \mathbf{q}_k < 0.2 \mathbf{p}_k' B_k \mathbf{p}_k. \end{cases}$$

5. Return to the step (2) unless a convergence criterion is satisfied.

As can be seen from the above description, there are two subproblems to be solved to produce each next iterate: solution of the quadratic optimization problem (step 2) and performing the line search (step 3). These are quite well studied problems and a wide range of techniques have been developed for dealing with them [18],[35]. In our implementation of PLASM based on the MATLAB computing environment, an Active Set method [35] is utilized to solve the quadratic optimization problem, and the simple algorithm based on the Armijo's stopping rule described in [48] is used to perform the line search.

4.6 Optimal Value for the Parameter t'

Like any other nonparametric algorithm, PLASM contains a free parameter t' whose value is likely to affect the level of the Prediction Error (1.4) of the resulting mode. Therefore, the issue of selection of the optimal value for t' has to be considered.

We propose to select the optimal value for t' based on minimization of an estimate of the Prediction Error of a model. We used the Generalized Cross-Validation (GCV) (1.7) as an estimate of that error:

$$\mathrm{GCV}(\mathbf{u}) = \frac{1}{N}\frac{\mathrm{RSS}(\mathbf{u})}{[1-\mathrm{df}(\mathbf{u})/N]^2}, \tag{4.16}$$

where

$$\mathrm{RSS}(\mathbf{u}) = (\mathbf{y}-A\beta)^T(\mathbf{y}-A\beta), \quad \beta = (A^TA+U^{-1})^{-1}A^T\mathbf{y}, \tag{4.17}$$

and

$$\mathrm{df}(\mathbf{u}) = \mathrm{tr}(H)(\mathbf{u}) = \mathrm{tr}(A(A^TA+U^{-1})^{-1}A^T). \tag{4.18}$$

The parameters $\mathbf{u}$ appearing in these formulas are obtained via solution of the optimization problem (3.25) with a particular value for t': $\mathbf{u} = \mathbf{u}_{t'}$. The optimal value for the parameter t' would be the one which corresponds to the lowest level of the GCV criterion. The simplest way to carry out the minimization of GCV is to evaluate it for a number of evenly-spaced values for the parameter t' and select such t' which results in the lowest level of GCV. This would however require the PLASM optimization problem (3.25) to be solved a number of times. An alternative (and less expensive) way for minimizing the GCV score could be as follows: first, one obtains the solution $\mathbf{u}_{t'_0}$ for (3.25) with the parameter t'_0 set to a certain small value. Then, given an orbitrary t', we approximate the

corresponding solution $\mathbf{u}_{t'}$ for (3.25) by the vector

$$\hat{\mathbf{u}}_{t'} = \frac{t' \cdot \mathbf{u}_{t'_0}}{\sum_{i=1}^{d} u_{t'_0 i}}.$$

Based on $\hat{\mathbf{u}}_{t'}$, the GCV score (4.16) can be computed. Thus, $\mathbf{u}_{t'_0}$ plays the role of a direction in the space of the PLASM solutions along which minimization of GCV occurs. This approach to GCV minimization would require the PLASM optimization problem to be solved only once. We used this technique in our experiments discussed in the next two sections.

It should be noted that the only objects involved in (3.25) which depend on the data are the cross-product matrix $A^T A$ and vector $A^T \mathbf{y}$. The cost of their evaluation is linear in the number of datapoints which is the largest parameter in most Data Mining problems. So, PLASM appears to be very well suited for dealing with large scale regression analysis.

4.7 Experiments with PLASM involving Synthetic Data

As was explained in section 2.2, ridge regression and LASSO are specific cases of PLASM corresponding to two most extreme types of grouping of basis functions. Therefore, the techniques employed by the PLASM approach can be used to estimate regression models via LASSO or ridge regression as well.

This section is intended to compare PLASM with LASSO, ridge regression and ordinary unconstrained least squares. To compare PLASM with the abovementioned techniques, we chose to model the following three functions involving different number of predictor variables:

$$\begin{aligned} f_1(\mathbf{x}) &= \log(x_1 + 0.01), \\ f_2(\mathbf{x}) &= \log(x_1 + 0.01) + x_2 + \cos(\pi x_3), \\ f_3(\mathbf{x}) &= \log(x_1 + 0.01) + x_2 + \cos(\pi x_3) - x_4 + \sin(\pi x_5), \end{aligned} \tag{4.19}$$

level of noise	PLASM	LASSO	Ridge Regression	Unconstraine d LSQ
10:3	0.36(0.03)	0.40(0.03)	0.41(0.04)	0.47(0.07)
2:1	0.52(0.03)	0.56(0.03)	0.59(0.05)	0.67(0.08)
10:8	0.93(0.10)	1.02(0.52)	1.03(0.10)	1.21(0.20)

Table 4.1: Prediction Errors along with the corresponding standard deviations (in parentheses) of models of the function $f_1(\mathbf{x})$ in (4.19).

level of noise	PLASM	LASSO	Ridge Regression	Unconstrained LSQ
10:3	0.38(0.05)	0.37(0.01)	0.40(0.05)	0.46(0.10)
2:1	0.61(0.05)	0.61(0.02)	0.63(0.05)	0.74(0.12)
10:8	0.94(0.11)	0.92(0.02)	1.00(0.18)	1.25(0.30)

Table 4.2: Prediction Errors along with the corresponding standard deviations (in parentheses) of models of the function $f_2(\mathbf{x})$ in (4.19).

where the argument $\mathbf{x}$ of each function is assumed to vary over the unit 5-dimensional hypercube. Using these functions we generated nine groups of datasets $\mathcal{D}_i^j$, $i = 1, 2, 3$, $j = 1, 2, 3$. Each group was made up of 50 independently sampled datasets of 1000 datapoints. The response values of a dataset from the group $\mathcal{D}_i^j$ were computed according to the formula

$$y_n^{ij} = f_i(\mathbf{x}) + \epsilon_n^j, \ n = 1, \ldots, 1000,$$

where $f_i(\mathbf{x})$ is one of the functions in (4.19), and ϵ^j represents a normally distributed random noise variable with a variance corresponding to the j-th level of the signal-to-noise ratio. We have used three levels of signal-to-noise ratio, namely $10:3$, $2:1$, $10:8$. The vectors of the values of predictor variables of all datasets were scattered across a unit 5-dimensional hypercube and computed according to the following procedure. The first variable x_1 was sampled from a uniform distribution while the rest of the variables were evaluated using the values of x_1: for $i = 2, \ldots, 5$

level of noise	PLASM	LASSO
10:3	15	6
2:1	10	0
10:8	6	0

Table 4.3: Results (percentage of models having the correct structure) of modelling the function $f_1(\mathrm{x})$ in (4.19).

level of noise	PLASM	LASSO
10:3	25	13
2:1	21	8
10:8	17	5

Table 4.4: Results (percentage of models having the correct structure) of modelling the function $f_2(\mathrm{x})$ in (4.19).

$$x_i = \begin{cases} 0, & \text{if } x_1 + \delta_i \leq 0 \\ 1, & \text{if } x_1 + \delta_i \geq 1 \\ x_1 + \delta_i, & \text{otherwise,} \end{cases}$$

where δ_i, $i = 2, \ldots, 5$ are independent zero mean normally distributed variables with variance 0.64. This ensured a certain degree of correlation between predictors. We used the following regression model in our experiments (note that all possible univariate components were initially present in the model so that the ability of the regression procedures to perform model selection could be tested):

$$f(\mathrm{x}) = f_1(x_1) + \ldots + f_5(x_5), \tag{4.20}$$

where

level of noise	PLASM	LASSO	Ridge Regression	Unconstraine d LSQ
10:3	0.51(0.08)	0.50(0.02)	0.52(0.08)	0.60(0.14)
2:1	0.70(0.03)	0.69(0.03)	0.71(0.05)	0.79(0.09)
10:8	0.97(0.10)	0.94(0.03)	0.99(0.09)	1.19(0.20)

Table 4.5: Prediction Errors along with the corresponding standard deviations (in parentheses) of models of the function $f_3(\mathbf{x})$ in (4.19).

$$f_i(x_i) = \sum_{j=1}^{7} \beta_{ij} B_{ij}(x_i), \quad i = 1, \ldots, 5. \tag{4.21}$$

The univariate basis functions were taken to be truncated powers

$$\{B_{ij}(x_i)\}_{j=1}^{7} = \{x_i, x_i^2, [x_i - 0.2(j-3)]_+^3, \ j = 3, \ldots, 7\}.$$

In order to produce the model of the type set out in (4.20) and (4.21), one has to estimate the regression coefficients β_{ij}, $i = 1, \ldots, 5$, $j = 1, \ldots, 7$. We estimated them using PLASM, LASSO, ridge regression, and ordinary least squares procedures based on the synthetic datasets generated as was described earlier. When running PLASM, we grouped the regression coefficients β into 5 groups: $\{\beta_{ij}, \ j = 1, \ldots, 7\}$, $i = 1, \ldots, 5$ whereas LASSO and ridge regression were implemented using 35 groups (one basis function per group) and 1 group (all basis functions in one group) respectively. The optimal values for the parameter t' in all three cases were estimated based on minimization of the GCV score as outlined in section 4.6 (we used $t'_0 = 100$ and $t'_0 = 15$ to obtain the directions in the spaces of the PLASM and LASSO solutions respectively). The results of the simulations are presented in the tables 4.1–4.5. The tables 4.1,4.2 and 4.5 contain values of the Prediction Error (PE) (1.4) of models for the functions $f_1(\mathbf{x})$, $f_2(\mathbf{x})$ and $f_3(\mathbf{x})$ (4.19) respectively. The estimated standard deviations for PE values are indicated in parentheses. The tables 4.3 and 4.4 display percentages of models by PLASM and LASSO of $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ in (4.19) respectively where the relevant predictors were correctly identified. These tables do not contain figures for ridge regression and unconstrained least squares as they

do not perform model selection.

From the results of the simulations it appears that PLASM demonstrates the best performance in the situations where only a small number of predictors are relevant. As the number of the relevant predictors increases, LASSO seems to perform better than PLASM though the difference is marginal. Both PLASM and LASSO outperform ridge regression as well as the unconstrained least squares procedure. As far as model selection is concerned, PLASM outperforms LASSO (least squares and ridge regression do not perform model selection and, therefore are not considered).

As was pointed out on the page 47 (chapter 3) PLASM involves solution of the optimization problem whose dimensionality is generally much lower compared with that of the LASSO approach. This is likely to result in a considerable reduction of the computational cost which is confirmed by the results of numerical experiments: in the series of runs based on the synthetic datasets outlined above, the average ratio of execution times of LASSO and PLASM was about 8.

4.8 Application of PLASM to Real Data

In this section we apply PLASM to modelling real data. The data was obtained from the StatLib Datasets Archive found at `http://lib.stat.cmu.edu/datasets/` (the file called `Bodyfat`). It is small but involves a fair number of covariates. The data has come from the medical field: it is often required to estimate the percentage of the fat in a person's body. There are costly techniques which allow one to obtain such estimates (i.e. underwater weighing). Therefore, it would be convenient to build a predictive equation whereby the fat percentage could be determined given a set of the relatively easily obtainable measurements. The dataset contains records for 252 people, each record being comprised of the values for the following measurements (covariates):

1. Age (years)
2. Weight (lbs)
3. Height (inches)
4. Neck circumference (cm)

5. Chest circumference (cm)
6. Abdomen 2 circumference (cm)
7. Hip circumference (cm)
8. Thigh circumference (cm)
9. Knee circumference (cm)
10. Ankle circumference (cm)
11. Biceps (extended) circumference (cm)
12. Forearm circumference (cm)
13. Wrist circumference (cm)

as well as a response value equal to the percentage of fat in the body of a person. We discarded 43 outlying datapoints from the data thereby getting the dataset comprised of 209 records. We applied LASSO, PLASM and ridge regression to regressing the percentage of bodyfat onto the covariates (measurements on the various parts of a body). In all three cases the data was split into two parts: the first one (142 records) was used to build additive models vis solution of the optimization problem (3.25) using different types of grouping. The optimal value for the parameter t' in each case was determined via minimization of the Generalized Cross Validation score as described in section 4.6. We used $t'_0 = 1.5$ $t'_0 = 10$ to obtain the directions in the spaces of the PLASM and LASSO solutions respectively. The quality of the models did not seem to be strongly dependent on the values for t'_0. The second part (67 records) was used to estimate the future prediction error of a model. Prior to the analysis the covariate vectors were translated as well as scaled to be contained in a unit 13-dimensional hypercube.

The graphs of the univariate components of the additive model produced by PLASM are shown in the Figure 4.1. An interesting conclusion can be made: abdomen circumference appears to be the most influential variable. The future Prediction Errors estimates in each case were: LASSO – 0.70, PLASM – 0.67 and ridge regression – 0.71. We would like to note that this analysis of the data is incomplete because we have not demonstrated that an additive model would be adequate in this case. Ideally, one would have to try to fit more complex models involving additive components as well as interaction terms

and, then, based on the results, draw conclusions regarding the structure of the optimal models. Unfortunately, our present software allows one to fit additive models only.

4.9 PLASM and Second-Order Cone Programming

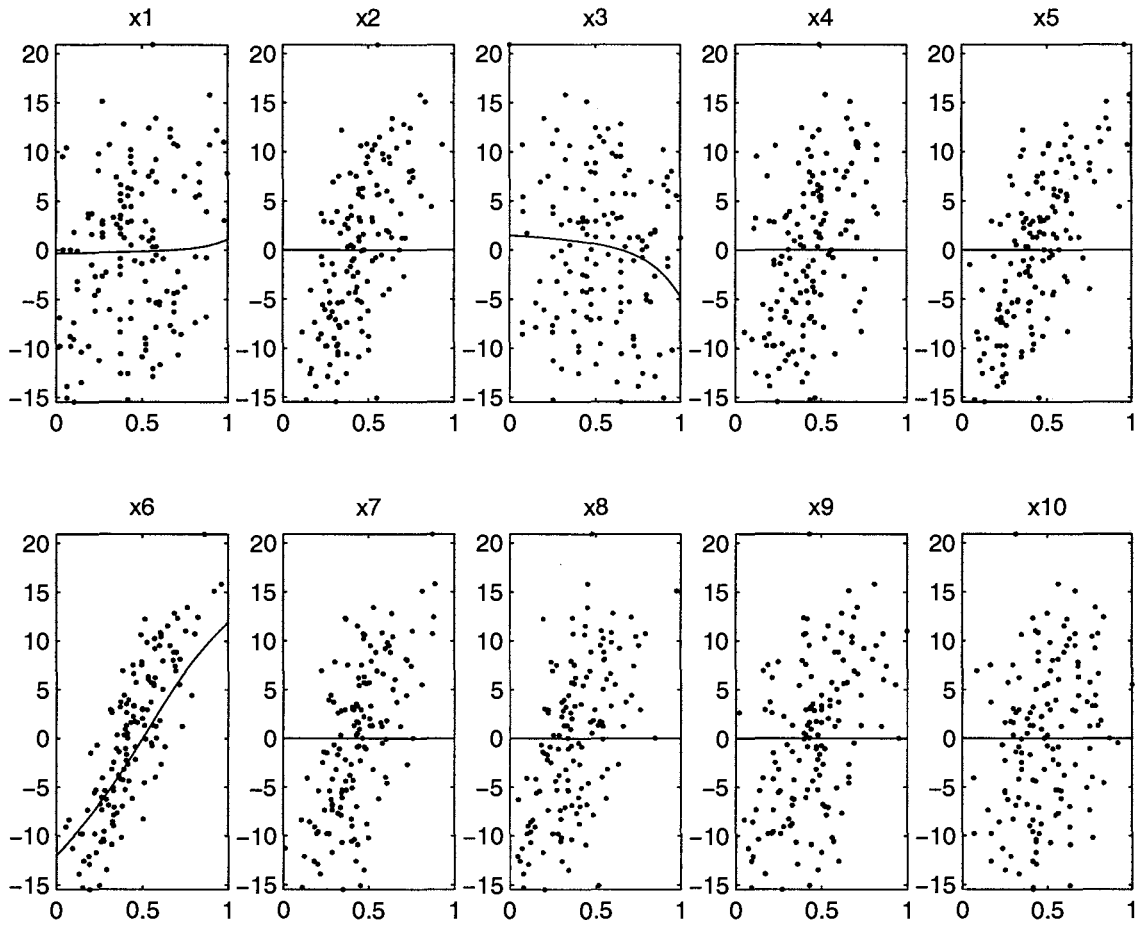
To finish off consideration of the numerical issues related to PLASM, we would to mention another approach that can be used to solve the optimization problem (2.7) *directly*. As one of the anonymous referees noticed, the problem (2.7) falls into the class of the Second-Order Cone Programs [34]. A general program of this type is formulated as follows:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \mathbf{q}^T\mathbf{x} \\ \text{subject to} \quad & ||P_i\mathbf{x} + \mathbf{r}_i|| \leq \mathbf{c}_i\mathbf{x} + d_i, \quad i = 1, \ldots, M. \end{aligned} \tag{4.22}$$

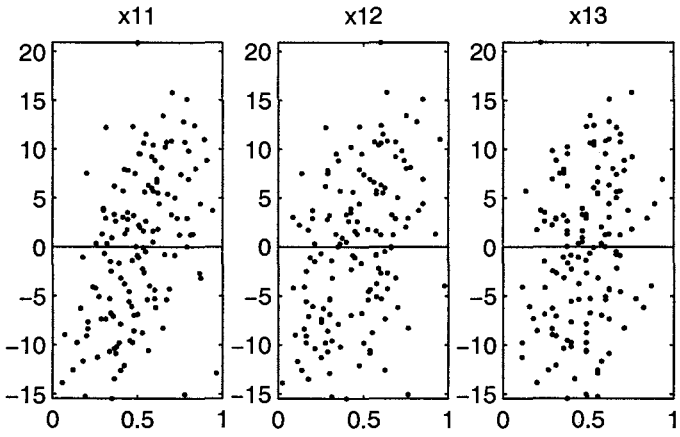
Here $\mathbf{x} \in R^m$ is the optimization variable; $\mathbf{q} \in R^m$ is the problem parameter; $P_i \in R^{(m_i-1)\times m}$, $i = 1, \ldots, M$, $\mathbf{r}_i \in R^{(m_i-1)}$, $i = 1, \ldots, M$, $\mathbf{c}_i \in R^m$, $i = 1, \ldots, M$ and $d_i \in R$, $i = 1, \ldots, M$. The PLASM optimization problem (2.7) can be recast in the same form:

$$\begin{aligned} \underset{\beta,\tau,\mathbf{s}}{\text{minimize}} \quad & s \\ \text{subject to} \quad & ||A\beta - \mathbf{y}|| \leq s, \\ & 0 \leq t - \sum_{i=1}^{d} \tau_i, \\ & ||\beta_i|| \leq \tau_i, \quad i = 1, \ldots, d. \end{aligned} \tag{4.23}$$

There have been developed interior-point methods aimed at solving the Second-Order Cone Programs [34],[41]. Thus, (2.7) can be solved via application of the abovecited techniques. However, it should be noted that the dimensionality of (4.23) is generally considerably higher that that of the new PLASM optimization problem (3.25). Therefore, it is not clear if the formulation of PLASM as a Second-Order Cone Program would be a better option compared with the approach discussed in this thesis and, obviously, more research has to be done to answer this question.



(a) variables x1-x10



(b) variables x11-x13

Figure 4.1: Graphs of (centered) univariate terms of additive model by PLASM in real data example (datapoints are shown by dots).

Chapter 5
PLASM and Penalized Least Squares

In the previous two chapters it was shown that the following formula can be used to compute the PLASM estimates of regression coefficients

$$\beta_{\text{plasm}} = (A^T A + U^{-1})^{-1} A^T \mathbf{y}, \tag{5.1}$$

where U is a diagonal matrix whose diagonal is comprised of groups of identical values. These values are determined via solution of the special optimization problem in (3.25). Now let us consider the following method of estimation of regression coefficients:

$$\beta_{\text{b-ridge}} = \text{argmin } (\mathbf{y} - A\beta)^T(\mathbf{y} - A\beta) + \sum_{i=1}^{d} \lambda_i \beta_i^T \beta_i. \tag{5.2}$$

Here the vector β is split into d groups

$$\beta = (\beta_1, \ldots, \beta_d),$$

and the second term is a weighted sum of squared norms of subvectors β_i, $i = 1, \ldots, d$. This approach would be equivalent to the ridge regression (2.5) if all λ_i, $i = 1, \ldots, d$ were set to be equal to each other. Thus, one could regard (5.2) as a block-ridge procedure. Simple calculations allow one to conclude that the solution of the block-ridge optimization

problem can be obtained by the following formula

$$\beta_{\text{b-ridge}} = (A^T A + \Lambda)^{-1} A^T \mathbf{y}, \tag{5.3}$$

where

$$\Lambda = \begin{pmatrix} \lambda_1 & & & & & \\ & \ddots & & & & \\ & & \lambda_1 & & 0 & \\ & & & \ddots & & \\ & & & & \ddots & \\ & & 0 & & \lambda_d & \\ & & & & & \ddots \\ & & & & & & \lambda_d \end{pmatrix}. \tag{5.4}$$

As can be seen, if one sets $\lambda_i = 1/u_i$, $i = 1, \ldots, d$, the formulae (5.1) and (5.3) for regression coefficients will coincide. Thus, PLASM (3.24) can be regarded as a procedure for obtaining optimal values for the parameters λ_i, $i = 1, \ldots, d$ in (5.2).

It can be shown that the procedure (5.2) is a special case of the more general methodology called *Penalised Least Squares* (PLS) [5],[26]. The discussion above suggests that it might be possible to devise a modified version of PLASM related to PLS in the same way as the original PLASM is related to the block-ridge procedure (5.2). In this chapter we will introduce such version of PLASM. Before doing so, we would like to outline the Penalised Least Squares approach to estimation of regression functions.

5.1 Penalized Least Squares

In order to introduce the Penalised Least Squares approach, we will consider estimation of additive models. Specifically, let us consider the following regression model

$$f(\mathbf{x}) = f_1(x_1) + ... + f_d(x_d), \tag{5.5}$$

where

$$f_i(x_i) = \sum_{j=1}^{p_i} \beta_{ij} B_{ij}(x_i), \ i = 1, ..., d, \tag{5.6}$$

and $B_{ij}(x_i)$, $j = 1, ..., p_i$ are univariate basis functions of the predictor x_i. According to the Penalized Least Squares methodology, one can estimate the regression coefficients β_{ij}, $j = 1, \ldots, p_i$, $i = 1, \ldots, d$ as follows

$$\beta \ = \ \text{argmin} \ (\mathbf{y} - A\beta)^T(\mathbf{y} - A\beta) + \sum_{i=1}^{d} \lambda_i \beta_i^T K_i \beta_i, \tag{5.7}$$

where K_i, $i = 1, \ldots, d$ are some symmetric positive semidefinite matrices and parameters λ_i, $i = 1, \ldots, d$ control relative weights of the penalty terms $\beta_i^T K_i \beta_i$, $i = 1, \ldots, d$ in the functional (5.7). As can be seen, the situation where $\lambda_i = 0$, $i = 1, \ldots, d$ corresponds to the unconstrained least squares estimation. The choice for the matrices K_i, $i = 1, \ldots, d$ may be dictated by various reasons. One example where they appear quite naturally is estimation of additive models (5.5) via smoothing splines [5]: the univariate components $f_1(x_1), \ldots, f_d(x_d)$ are taken to be the ones which provide solution for the following optimization problem

$$(f_1, \ldots, f_d) \ = \ \text{argmin} \ ||\mathbf{y} - \sum_{i=1}^{d} f_i(x_i)||^2 + \sum_{i=1}^{d} \lambda_i \int [f_i''(x_i)]^2 \, \mathrm{d}x_i, \tag{5.8}$$

where minimization is performed over a space of smooth functions. Here the penalty terms $\int [f_i''(x_i)]^2 \, \mathrm{d}x_i$, $i = 1, \ldots, d$ measure roughness of the univariate functions f_i, $i = 1, \ldots, d$. Generally, this is an infinite dimensional optimization problem and a possible way to reduce

(5.8) to a finite dimensional problem is to confine $f_i,\ i = 1, \ldots, d$ to a finite dimensional space of functions. Specifically, one can model $f_i(x_i),\ i = 1, \ldots, d$ as linear combinations of some (smooth enough) basis functions $B_{ij}(x_i),\ j = 1, \ldots, p_i,\ i = 1, \ldots, d$. In this case

$$\int [f_i''(x_i)]^2 \, \mathrm{d}x_i = \sum_{j_1,\ j_2=1}^{p_i} [\int B_{ij_1}''(x_i) B_{ij_2}''(x_i) \, \mathrm{d}x_i] \, \beta_{ij_1} \beta_{ij_2}$$

and, therefore

$$\int [f_i''(x_i)]^2 \mathrm{d}x_i = \beta_i^T K_i \beta_i,$$

where entries of the matrix $K_i = \{k^i_{j_1 j_2}\}$ are evaluated as

$$k^i_{j_1 j_2} = \int B_{ij_1}''(x_i) B_{ij_2}''(x_i) \, \mathrm{d}x_i,\ j_1 = 1, \ldots, p_i,\ j_2 = 1, \ldots, p_i. \tag{5.9}$$

By differentiating the functional in (5.7) with respect to β and equating the derivatives to zero, one can obtain the formula for the PLS estimates of the regression coefficients

$$\beta_{\mathrm{pls}} = (A^T A + \Lambda K)^{-1} A^T \mathbf{y}, \tag{5.10}$$

where

$$K = \begin{pmatrix} K_1 & & \\ & \ddots & \\ & & K_d \end{pmatrix}, \tag{5.11}$$

and the matrix Λ is defined in (5.4). Note that the block-ridge procedure (5.2) considered earlier is a special case of the Penalised Least Squares with $K_i = I,\ i = 1, \ldots, d$.

5.2 Modified PLASM

In this section we will consider a modified formulation of PLASM. Specifically, we propose to estimate regression coefficients β according to the following procedure

$$\begin{aligned} \beta(t) &= \text{argmin } (\mathbf{y} - A\beta)^T(\mathbf{y} - A\beta) \\ &\quad \text{subject to } \sum_{i=1}^{d}[\beta_i^T K_i \beta_i]^{\frac{1}{2}} \leq t, \end{aligned} \tag{5.12}$$

where K_i, $i = 1, \ldots, d$ are symmetric positive definite matrices. The formulation (5.12) differs from (3.24) in the respect that l_2 norms of the groups of regression coefficients $[\beta_i^T \beta_i]^{\frac{1}{2}}$, $i = 1, \ldots, d$ in the constraint of PLASM are replaced with expressions corresponding to more general form of a scalar product. In order to ensure that the constraint in (5.12) is active, the parameter t has to be from the following range

$$t \in (0, t_r), \quad \text{where } t_r = \sum_{i=1}^{d}[\beta_i^{oT} K_i \beta_i^o]^{\frac{1}{2}}. \tag{5.13}$$

Here $\beta^o = (A^T A)^{-1} A^T \mathbf{y}$ is the vector of unconstrained least squares estimates of the regression coefficients. As was repeatedly pointed out before, one has to determine the optimal value for t on the basis of minimization of some estimate of the Prediction Error (1.4).

5.3 New Formulation of the Modified PLASM

It was shown at the beginning of the chapter 2 that the numerical solution of an optimization problem of the type involved in (5.12) is difficult and, therefore, an alternative equivalent formulation is required. Due to the fact that the matrices K_i, $i = 1, \ldots, d$ are nonsingular, such formulation is possible to obtain using the results of the chapter 3. Indeed, let us transform regression coefficients β as well as the model matrix A as follows

$$\begin{aligned} \beta^* &\leftarrow K^{\frac{1}{2}}\beta, \\ A^* &\leftarrow AK^{-\frac{1}{2}}, \end{aligned} \tag{5.14}$$

where K is defined in (5.11). In terms of the new variables the optimization problem (5.12) becomes

$$\begin{aligned} \beta^*(t) &= \text{argmin } (\mathbf{y} - A^*\beta^*)^T(\mathbf{y} - A^*\beta^*) \\ &\quad \text{subject to } \sum_{i=1}^{d}[\beta_i^{*T}\beta_i^*]^{\frac{1}{2}} \leq t, \end{aligned} \tag{5.15}$$

where

$$\begin{aligned} t &\in (0, t_r^*), \\ t_r &= \sum_{i=1}^{d}[\beta_i^{*oT}\beta_i^{*o}]^{\frac{1}{2}}, \end{aligned} \tag{5.16}$$

Here $\beta^{*o} = (A^{*T}A^*)^{-1}A^{*T}\mathbf{y}$. Now, the form of the optimization problem in (5.15) coincides with that of the optimization problem in (3.24). Consequently, according to the results of the chapter 3, (5.15) together with (5.16) can be recast as

$$\begin{aligned} \tilde{\beta}(t') &= (A^{*T}A^* + U^{-1})^{-1}A^{*T}\mathbf{y}, \\ \mathbf{u} &= \text{argmin } -\mathbf{y}^TA^*(A^{*T}A^* + U^{-1})^{-1}A^{*T}\mathbf{y} \\ &\quad \text{subject to } \sum_{i=1}^{d} u_i \leq t', \\ &\quad u_i \geq 0, \quad i = 1, \ldots, d, \end{aligned} \tag{5.17}$$

where $t' \in (0, \infty)$. This leads to a new formulation of the modified PLASM procedure

(5.12) that can be derived by reverting to the original regression coefficients β and model matrix A

$$\begin{aligned} \beta &\leftarrow K^{-\frac{1}{2}}\beta^*, \\ A &\leftarrow A^*K^{\frac{1}{2}}, \end{aligned}$$

in the formulation (5.17)

$$\begin{aligned} \tilde{\beta}(t') &= (A^TA + KU^{-1})^{-1}A^T\mathbf{y}, \\ \mathbf{u} &= \text{argmin } -\mathbf{y}^TA(A^TA + KU^{-1})^{-1}A^T\mathbf{y} \\ &\quad \text{subject to } \sum_{i=1}^{d} u_i \leq t', \\ &\quad u_i \geq 0, \quad i = 1, \ldots, d, \end{aligned} \tag{5.18}$$

where $t' \in (0, \infty)$. Below we cast this result as a Proposition.

Proposition 5.3.1 *The sets of regression coefficients $\mathcal{B}_1$ and $\mathcal{B}_2$ defined as*

$$\mathcal{B}_1 = \{\beta(t) : \textit{solutions of (5.12) for } t \in (0, t_r)\}$$

and

$$\mathcal{B}_2 = \{\tilde{\beta}(t') : \textit{solutions of (5.18) for } t' \in (0, \infty)\}$$

are identical.

Thus, as was also the case with the original PLASM (3.24), the set $\mathcal{B}_2$ is just a reparametrized set $\mathcal{B}_1$. Note that all results obtained in the chapter 3 hold for the modified PLASM as well. Now we are able to establish the connection between the modified PLASM and the Penalized Least Squares procedure. Indeed, examination of the formulae for regres-

sion coefficients in (5.18) and (5.10) reveals that both of them have the same form if $\lambda_i = 1/u_i,\ i = 1, \ldots, d$. So, one can view the modified PLASM as a procedure for determination of optimal values for the parameters $\lambda_i,\ i = 1, \ldots, d$ in the Penalized Least Squares approach.

5.4 Case of Positive Semidefinite Matrices K_i

In the previous section we considered the modified version of PLASM (5.12) where matrices $K_i,\ i = 1, \ldots, d$ were assumed to be positive definite. However, as we saw in the section 5.1, one of the situations where PLS or modified PLASM formulations appear naturally is regression based on smoothing splines (5.8). In that case the matrices $K_i,\ i = 1, \ldots, d$ defined in (5.9) are typically only semidefinite due to the fact that $\int [f'']^2 dx = 0$ for linear functions. This renders considerations of the previous section invalid. Below we will demonstrate how this difficulty can be overcome at least in some situations. Let us again consider the estimation of additive models

$$f(\mathbf{x}) = \beta_{00} + \sum_{i=1}^{d} f_i(x_i),\ x_i \in (0,1),\ f_i(0) = 0.$$

Note that we introduced an intercept β_{00} and assumed that all $f_i(x_i)$ vanish at $x_i = 0,\ i = 1, \ldots, d$[1]. This was done to eliminate an ambiguity in the definition of $f_i(x_i),\ i = 1, \ldots, d$. Also, we will assume that the univariate components $f_i(x_i),\ i = 1, \ldots, d$ are modelled in the following way

$$f_i(x_i) = \beta_{i0} x_i + \sum_{j=1}^{p_i} \beta_{ij} B_{ij}(x_i),\quad i = 1, \ldots, d, \tag{5.19}$$

where x_i and $B_{ij}(x_i),\ j = 1, \ldots, p_i$ form a linearly independent system of univariate functions. An example of such system can be built according to the following procedure. Let $\{\gamma_j\}_{j=1}^{p_i}$ be a set of knots placed on the variable x_i. Given these knots, one can construct a set of univariate cubic B-splines [11]

[1] For the sake of simplicity, we assume that predictor data vectors are scattered across d-dimensional unit hypercube.

$$B_{ij}(x_i),\ j = 1, \ldots, p_i + 2 \tag{5.20}$$

Now, let us drop first two ($B_{i1}(x_i)$ and $B_{i2}(x_i)$) B-splines from this basis. The obtained set of functions

$$B_{ij}(x_i),\ j = 3, \ldots, p_i + 2 \tag{5.21}$$

represents a system with the desired properties. Indeed, $B_{i1}(x_i)$ is the only function in (5.20) that does not vanish at $x_i = 0$. Therefore, any linear combination of the basis functions in (5.21) vanishes at $x_i = 0$. Also, $B_{i2}(x_i)$ is the only function in (5.20) whose first order derivative does not vanish to zero at $x_i = 0$. Consequently, x_i and functions in (5.21) are linearly independent.

As was pointed out in the section 5.1, the univariate components $f_i(x_i),\ i = 1, \ldots, d$ can be estimated based on the smoothing methodology that amounts to minimization of the functional (5.8). By inserting the expressions (5.19) for $f_i(x_i),\ i = 1, \ldots, d$ into (5.8), one can reduce it to a finite dimensional problem

$$(\beta_0, \beta) \ = \ \text{argmin}\ (\mathbf{y} - A_0\beta_0 - A\beta)^T(\mathbf{y} - A_0\beta_0 - A\beta) + \sum_{i=1}^{d} \lambda_i \beta_i^T K_i \beta_i. \tag{5.22}$$

Here $\beta_0^T = (\beta_{00}, \beta_{10}, \ldots \beta_{d0})$, $\beta^T = (\beta_1^T, \ldots \beta_d^T)$, where $\beta_i^T = (\beta_{i1}, \ldots \beta_{ip_i})$ and $K_i,\ i = 1, \ldots, d$ are defined in (5.9) though, due to our using special basis functions in (5.19), these matrices are nondegenerate. The model matrices A_0 and A correspond to the linear and nonlinear basis functions respectively. Thus, the penalty term depends only on the coefficients associated with the nonlinear basis functions of the model.

The nondegeneracy of the matrices $K_i,\ i = 1, \ldots, d$ opens up an opportunity for using the modified PLASM procedure introduced earlier

$$\begin{aligned}(\beta_0, \beta) &= \text{argmin } (\mathbf{y} - A_0\beta_0 - A\beta)^T(\mathbf{y} - A_0\beta_0 - A\beta) \\ &\quad \text{subject to } \sum_{i=1}^{d}[\beta_i^T K_i \beta_i]^{\frac{1}{2}} \leq t. \end{aligned} \tag{5.23}$$

As before, the optimal value for the parameter t has to be determined via minimization of some estimate of the Prediction Error (see section 4.6). It is worth noting that the optimization problem (5.23) can be reformulated in terms of variables β only. Indeed, given β, the coefficients β_0 can be determined analytically as there are no constraints in (5.23) involving β_0. Therefore,

$$\beta_0 = (A_0^T A_0)^{-1} A_0^T (y - A\beta). \tag{5.24}$$

Thus, (5.23) can be recast as

$$\begin{aligned}\beta &= \text{argmin } (\tilde{\mathbf{y}} - \tilde{A}\beta)^T(\tilde{\mathbf{y}} - \tilde{A}\beta) \\ &\quad \text{subject to } \sum_{i=1}^{d}[\beta_i^T K_i \beta_i]^{\frac{1}{2}} \leq t, \end{aligned} \tag{5.25}$$

where $\tilde{\mathbf{y}} = \mathbf{y} - H\mathbf{y}$, $\tilde{A} = A - HA$ and $H = A_0(A_0^T A_0)^{-1} A_0$. Thus, one obtains β via solution of the optimization problem (5.25) and β_0 by the formula (5.24). As was shown in the section 5.3, instead of solving (5.25), one can deal with more numerically tractable problem (5.17). In this case the vector β can be computed according to the formula

$$\beta = (\tilde{A}^T \tilde{A} + KU^{-1})^{-1} \tilde{A}^T \tilde{\mathbf{y}}, \tag{5.26}$$

where quantities u_i, $i = 1, \ldots, d$ are obtained by solving the optimization problem (5.17). As we pointed out earlier, in this context the modified PLASM can be regarded as a method

for selection of optimal values for the parameters λ_i, $i = 1, \ldots, d$ in (5.22). Specifically, according to PLASM, $\lambda_i = 1/u_i$, $i = 1, \ldots, d$, where $u_i = 0$ implies that the corresponding nonlinear univariate component $\sum_{j=1}^{p_i} \beta_{ij} B_{ij}(x_i)$ in (5.19) is excluded from the model.

Chapter 6
Probing Least Absolute Deviations Modelling

In the course of our investigation of the PLASM procedure we measured the goodness of fit in terms of the residual sum of squares (RSS). In this chapter we will consider an alternative measure of fit based on the *sum of absolute deviations*. We would like to stress straight away that this kind of procedures is unlikely to be useful for Data Mining purposes due to the high computational cost. So, the considerations of this chapter can be regarded as an academic exercise demonstrating that the ideas used to obtain the PLASM estimator (3.24) can be used in other settings as well.

6.1 Introduction of PLADM

We propose to estimate the regression coefficients (β, β_0) of the model (2.1) according to the following procedure

$$\begin{aligned}(\beta, \beta_0) &= \underset{\beta,\beta_0}{\operatorname{argmin}} \sum_{n=1}^{N} |y_n - \beta_0 - \mathbf{T}_n\beta| \\ &\text{subject to } \sum_{i=1}^{d} [\beta_i^T K_i \beta_i]^{\frac{1}{2}} \leq t, \end{aligned} \tag{6.1}$$

where t is a free parameter of the problem, $\mathbf{T}_n$ is the n-th row of a model matrix T, and K_i, $i = 1, \ldots, d$ are some symmetric positive definite matrices. As before, we assume that the regression coefficients β are split into d groups so that $\beta^T = (\beta_1^T, \ldots, \beta_d^T)$. This

formulation is similar to that of the original PLASM (2.7) except for the function used to measure deviations of datapoints from the regression surface. Specifically, we use the sum of absolute deviations as an objective function to be minimized. Thus, (6.1) can be regarded as a modification of a conventional l_1 regression which is known to be more robust to the presence of outliers compared to a least squares procedure [1].

The objective function in (6.1) is convex rather than strictly convex. Thus, the problem (6.1) may have several solutions. In order to simplify the further theoretical investigations, we replace the objective function in (6.1) with a slightly different one

$$f(\beta, \beta_0) = \sum_{n=1}^{N} [(y_n - \beta_0 - \mathbf{T}_n \beta)^2 + \alpha_1]^{\frac{1}{2}}, \tag{6.2}$$

where α_1 is a small parameter. This measure of deviations of datapoints from a regression surface produces the same results for large residuals as the original one in (6.1). Therefore, regression estimations obtained via minimization of (6.2) have the same favourable robustness properties as those based on the genuine l_1 loss function. Now we will show that the function (6.2) is strictly convex.

Lemma 6.1.1 *The function f (6.2) is strictly convex provided that the matrix $[T\ \mathbf{1}]$ is of full rank.*

Proof. For notational convenience we bundle β_0 and β together and augment the matrix T with the column of ones on the right so that

$$\begin{aligned} \gamma^T &= (\beta^T, \beta_0), \\ B &= [T\ \mathbf{1}]. \end{aligned} \tag{6.3}$$

Consider the n-th term

$$f_n(\gamma) = [(y_n - \mathbf{B}_n \gamma)^2 + \alpha_1]^{\frac{1}{2}}$$

of (6.2) and compute its Hessian matrix. The first and second order derivatives of f_n respectively are

$$\frac{\partial f_n}{\partial \gamma_l} = \frac{B_{nl}(\mathbf{B}_n\gamma - y_n)}{[(y_n - \mathbf{B}_n\gamma)^2 + \alpha_1]^{\frac{1}{2}}}, \quad l = 1, \ldots, P+1,$$

and

$$\begin{aligned}\frac{\partial^2 f_n}{\partial \gamma_l \partial \gamma_k} &= \frac{B_{nl}B_{nk}}{[(y_n - \mathbf{B}_n\gamma)^2 + \alpha_1]^{\frac{1}{2}}} - \frac{B_{nl}(\mathbf{B}_n\gamma - y_n)^2 B_{nk}}{[(y_n - \mathbf{B}_n\gamma)^2 + \alpha_1]^{\frac{3}{2}}} \\ &= \frac{B_{nl}B_{nk}\alpha_1}{[(y_n - \mathbf{B}_n\gamma)^2 + \alpha_1]^{\frac{3}{2}}}, \quad l, k = 1, \ldots, P+1.\end{aligned}$$

Let $\mathbf{z}$ be an arbitrary $(P+1)$-dimensional vector. Then, the quadratic form $\mathbf{z}^T F_n \mathbf{z}$ can be cast as follows

$$\mathbf{z}^T F_n \mathbf{z} = \alpha_1 \frac{\mathbf{z}^T \mathbf{B}_n^T \mathbf{B}_n \mathbf{z}}{[(y_n - \mathbf{B}_n\gamma)^2 + \alpha_1]^{\frac{3}{2}}} \geq 0.$$

where F_n is the Hessian matrix of the function $f_n(\gamma)$. Considering that the Hessian F of the function (6.2) is a sum of F_n, $n = 1, \ldots, N$, we have

$$\mathbf{z}^T F \mathbf{z} = \alpha_1 \sum_{n=1}^{N} \frac{\mathbf{z}^T \mathbf{B}_n^T \mathbf{B}_n \mathbf{z}}{[(y_n - \mathbf{B}_n\gamma)^2 + \alpha_1]^{\frac{3}{2}}} > 0$$

because B is assumed to be of the full rank. Thus, the Hessian F is positive definite and, therefore, the function (6.2) is strictly convex. □

Remark. We introduced the vector γ in (6.3) which may be viewed as having been split into $d+1$ groups: the first d groups correspond to those of the vector β and the last, $(d+1)$-st group is comprised of a single coefficient (intercept) β_0. Also, in our further investigation, we will assume that the matrix B in (6.3) is of the full rank.

Thus, using (6.2) as an objective function, we arrive at the following optimization problem for estimation of the regression coefficients (β, β_0)

$$
\begin{aligned}
(\beta, \beta_0) &= \underset{\beta,\beta_0}{\operatorname{argmin}} \sum_{n=1}^{N} [(y_n - \beta_0 - \mathbf{T}_n\beta)^2 + \alpha_1]^{\frac{1}{2}} \\
&\text{subject to } \sum_{i=1}^{d} [\beta_i^T K_i \beta_i]^{\frac{1}{2}} \leq t,
\end{aligned} \tag{6.4}
$$

where

$$
\begin{aligned}
t &\in (0, t_r), \\
t_r &= \sum_{i=1}^{d} [\beta_i^{oT} K_i \beta_i^o]^{\frac{1}{2}},
\end{aligned} \tag{6.5}
$$

and β^o is an unconstrained minimizer of the function (6.2). It follows from the Lemma above that, for each value of the parameter t from the interval defined in (6.5), the problem (6.4) has a unique solution located on the boundary of the domain defined by its constraint. Note that here, in contrast with the original PLASM (2.7), one cannot dispose of the intercept β_0 by simply centering the columns of the model matrix T.

6.2 Regularized PLADM

Numerical solution of the optimization problem (6.4) based on the Kuhn-Tucker conditions may not be possible. Indeed, due to the presence of the square roots in the PLADM's constraint, its partial derivatives with respect to regression coefficients take the form of the ratio where $[\beta_i^{oT} K_i \beta_i^o]^{\frac{1}{2}}$ appears in the denominator. As some groups of the regression coefficients β_i^*, $i = 1, \ldots, d$ of the solution (β^*, β_0^*) may be zero, this implies that the partial derivatives of the constraint may not be defined in the classical sense.[1]

To circumvent this difficulty, we introduce a regularization parameter α_2 which would render our problem more numerically tractable. The regularized PLADM takes the following

[1] Elementary geometrical considerations suggest that, given a strictly convex objective function, it is the shape of the feasible region which causes some groups of components of the solution vector to vanish to zero. Therefore, considering that the feasible regions of PLASM and PLADM are the same, one may expect that PLADM would set some of the groups of regression coefficients to zero as well.

form

$$
\begin{aligned}
(\beta, \beta_0) &= \underset{\beta,\beta_0}{\operatorname{argmin}} \sum_{n=1}^{N} [(y_n - \beta_0 - \mathbf{T}_n\beta)^2 + \alpha_1]^{\frac{1}{2}} \\
&\text{subject to } \sum_{i=1}^{d} [\beta_i^T K_i \beta_i + \alpha_2]^{\frac{1}{2}} \leq t,
\end{aligned} \tag{6.6}
$$

where the value of the parameter t is from the range

$$
\begin{aligned}
t &\in (t_l^{\alpha_{1,2}}, t_r^{\alpha_{1,2}}), \\
t_l^{\alpha_{1,2}} &= d\alpha_2^{\frac{1}{2}}, \\
t_r^{\alpha_{1,2}} &= \sum_{i=1}^{d} [\beta_i^{oT} K_i \beta_i^o + \alpha_2]^{\frac{1}{2}}.
\end{aligned} \tag{6.7}
$$

Here β^o is an unconstrained minimizer of the function (6.2). It is possible to prove that the solution $\beta(\alpha_2)$ of (6.6) approaches the solution of the problem (6.4) as $\alpha_2 \to 0$. The proof can be produced based on the same approach as that used to prove the Proposition 2.3.1. Indeed, the only property of the objective function used in the proof is its strict convexity which, according to the Lemma 6.1.1, is also possessed by the objective function in (6.6). Further considerations of the regularized PLADM presented in this and the next chapter will follow pretty much the same pattern as those of the chapter 2 dedicated to the original PLASM (2.7).

6.3 Kuhn-Tucker Conditions for the Regularized PLADM

In this section we will derive a system of nonlinear equations equivalent to the Kuhn-Tucker conditions for the regularized PLADM. In order to simplify our calculations, we recast the problem as

$$
(\beta, \beta_0) = \underset{\beta,\beta_0,\xi,\tau}{\operatorname{argmin}} \sum_{n=1}^{N} \xi_n^2
$$

$$\begin{aligned} \text{subject to } & (y_n - \beta_0 - \mathbf{T}_n\beta)^2 + \alpha_1 = \xi_n^4, \quad n = 1, \ldots, N \\ & \beta_i^T K_i \beta_i + \alpha_2 = \tau_i^4, \qquad\qquad i = 1, \ldots, d \\ & \sum_{i=1}^{d} \tau_i^2 \leq t. \end{aligned} \tag{6.8}$$

The Lagrangian for this problem is as follows

$$\begin{aligned} L = \sum_{n=1}^{N} \xi_n^2 \quad & + \quad \sum_{n=1}^{N} \frac{1}{w_n}[(y_n - \beta_0 - \mathbf{T}_n\beta)^2 + \alpha_1 - \xi_n^4] \\ & + \quad \sum_{i=1}^{d} \mu_i[\beta_i^T K_i \beta_i + \alpha_2 - \tau_i^4] + \lambda(\sum_{i=1}^{d} \tau_i^2 - t), \end{aligned} \tag{6.9}$$

where $1/w_n$, $n = 1, \ldots, N$, μ_i, $i = 1, \ldots, d$ and λ are Lagrange multipliers. We wrote $1/w_n$ instead of w_n as this will result in simplification of our further calculations. The Kuhn-Tucker conditions can be obtained by differentiating the above Lagrangian with respect to τ_i, $i = 1, \ldots, d$ and ξ_n, $n = 1, \ldots, N$ and equating the derivatives to zero. This operation yields

$$\begin{aligned} \gamma &= (B^T W^{-1} B + M K_0)^{-1} B^T W^{-1} \mathbf{y}, \\ 1 &= 2\frac{1}{w_n}\xi_n^2, \quad n = 1, \ldots, N, \\ \lambda &= \mu_i 2\tau_i^2, \quad i = 1, \ldots, d, \end{aligned} \tag{6.10}$$

where B and γ are defined in (6.3), and the matrices M, W and K_0 are

$$W = \begin{pmatrix} w_1 & & \\ & \ddots & \\ & & w_N \end{pmatrix}, \quad K_0 = \begin{pmatrix} K_1 & & \\ & \ddots & \\ & & K_d \end{pmatrix}, \tag{6.11}$$

$$0$$

$$M = \begin{pmatrix} \mu_1 & & & & & & \\ & \ddots & & & & & \\ & & \mu_1 & & 0 & & \\ & & & \ddots & & & \\ & 0 & & & \mu_p & & \\ & & & & & \ddots & \\ & & & & & & \mu_p \\ & & & & & & & 0 \end{pmatrix}. \tag{6.12}$$

From the second line in (6.10) it follows that $1/w_n$ are bound to be distinct from zero and, therefore, the replacement of w_n with $1/w_n$ in (6.9) was legitimate.

Insert the expression for γ in (6.10) into the equality constraints of the optimization problem (6.8). This produces the following system of equations $i = 1, \ldots, d$, $n = 1, \ldots, N$:

$$\begin{aligned} \mathbf{y}W^{-1}B(B^TW^{-1}B + V^{-2}K_0)^{-1}K_0I_i(B^TW^{-1}B + V^{-2}K_0)^{-1}B^TW^{-1}\mathbf{y} &= \frac{\lambda^2}{4}v_i^4 - \alpha_2, \\ \left[y_n - \mathbf{B}_n(B^TW^{-1}B + V^{-2}K_0)^{-1}B^TW^{-1}\mathbf{y}\right]^2 + \alpha_1 &= \frac{1}{4}w_n^2, \\ \sum_{i=1}^{d} v_i^2 &= \frac{2t}{\lambda}, \end{aligned} \tag{6.13}$$

where we introduce new variables

$$v_i^2 = \frac{2\tau_i^2}{\lambda}, \quad i = 1, \ldots, d,$$

and make use of the fact that $w_n/2 = \xi_n^2$, $n = 1, \ldots, N$ which follows from (6.10). Now we will show that the system (6.13) has a unique solution in terms of variables w_n, $n = 1, \ldots, N$, v_i^2, $i = 1, \ldots, d$ and λ.

Lemma 6.3.1 *The system (6.13) has a unique solution for* w_n, $n = 1, \ldots, N$, v_i^2, $i = 1, \ldots, d$ *and* λ.

Proof. The Lagrangian for the regularized PLADM optimization problem (6.6) can be written as follows

$$L = \sum_{n=1}^{N}[(y_n - \mathbf{B}_n\gamma)^2 + \alpha_1]^{\frac{1}{2}} + \hat{\lambda}(\sum_{i=1}^{d}[\beta_i^T K_i\beta_i + \alpha_2]^{\frac{1}{2}} - t),$$

where, according to (6.3), $\gamma^T = (\beta_1, \ldots \beta_p, \beta_0)$. The Kuhn-Tucker conditions for this problem are

$$\begin{aligned}\frac{\partial L}{\partial \gamma_l} &= \sum_{n=1}^{N}\frac{(\mathbf{B}_n\gamma - y_n)B_{n,l}}{[(y_n - \mathbf{B}_n\gamma)^2 + \alpha_1]^{\frac{1}{2}}} + \hat{\lambda}\frac{K_{i(l)}\beta_{i(l)}}{[\beta_{i(l)}^T K_{i(l)}\beta_{i(l)} + \alpha_2]^{\frac{1}{2}}} = 0, \quad l = 1, \ldots, P+1, \\ &\hat{\lambda}(\sum_{i=1}^{d}[\beta_i^T K_i\beta_i + \alpha_2]^{\frac{1}{2}} - t) = 0, \end{aligned} \tag{6.14}$$

where $i(l)$ is the number of the group of regression coefficients containing variable γ_l. Note that, according to the remark after the Lemma 6.1.1, the $(d+1)$-st group is comprised of a single coefficient (intercept) β_0 and $K_{i(P+1)} = 0$. We denote

$$\begin{aligned}\hat{w}_n &= 2[(y_n - \mathbf{B}_n\gamma)^2 + \alpha_1]^{\frac{1}{2}}, \quad n = 1, \ldots, N, \\ \hat{v}_i^2 &= \frac{2}{\hat{\lambda}}[\beta_i^T K_i\beta_i + \alpha_2]^{\frac{1}{2}}, \qquad i = 1, \ldots, d. \end{aligned} \tag{6.15}$$

Using this notation one can express the solution γ of the problem (6.6) in the following form

$$\gamma = (B^T\hat{W}^{-1}B + \hat{V}^{-2}K_0)^{-1}B^T\hat{W}^{-1}\mathbf{y}. \tag{6.16}$$

Taking into consideration that $\beta_i^T K_i\beta_i = \gamma^T K_0 I_i\gamma$ and inserting the above expression for γ into the relations (6.15), one obtains a system of nonlinear equations for $\hat{v}_i^2$, $i = 1, \ldots, d$, $\hat{w}_n$, $n = 1, \ldots, N$ and $\hat{\lambda}$

$$\begin{aligned}
\mathbf{y}\hat{W}^{-1}B(B^T\hat{W}^{-1}B+\hat{V}^{-2}K_0)^{-1}K_0I_i(B^T\hat{W}^{-1}B+\hat{V}^{-2}K_0)^{-1}B^T\hat{W}^{-1}\mathbf{y} &= \frac{\hat{\lambda}^2}{4}\hat{v}_i^4-\alpha_2,\\
\left[y_n-\mathbf{B}_n(B^T\hat{W}^{-1}B+\hat{V}^{-2}K_0)^{-1}B^T\hat{W}^{-1}\mathbf{y}\right]^2+\alpha_1 &= \frac{1}{4}\hat{w}_n^2,\\
\sum_{i=1}^{d}\hat{v}_i^2 &= \frac{2t}{\hat{\lambda}}. \qquad (6.17)
\end{aligned}$$

The last equation in the above system is due to the constraint of the optimization problem (6.6). We found out earlier that the problem (6.6) has a unique solution and, therefore, the Kuhn-Tucker conditions (6.14) have a unique solution as well. Due to the one-to-one correspondence between solutions of the systems (6.14) and (6.17), the latter also has a unique solution. Considering that the system (6.17) has the same form as (6.13), we conclude that (6.13) has a unique solution in terms of $w_n,\ n=1,\ldots,N,\ v_i^2,\ i=1,\ldots,d$ and λ too. □

In summary, in this chapter we derived the system of nonlinear equations (6.13) to be solved for $w_n,\ n=1,\ldots,N,\ v_i^2,\ i=1,\ldots,d$ and λ. Having obtained the solution of this system, one can find the solution of the regularized PLADM optimization problem (6.6) according to the formula

$$\gamma=(B^TW^{-1}B+V^{-2}K_0)^{-1}B^TW^{-1}\mathbf{y}. \qquad (6.18)$$

At this point an interesting observation can be made. We found out in the chapter 5 that the Penalised Least Squares estimators of regression coefficients have the form

$$\gamma=(B^TB+\Lambda K_0)^{-1}B^T\mathbf{y},$$

where values for the parameters $\lambda_i,\ i=1,\ldots,d$ can be determined via, for instance, the PLASM approach (3.25). As can be seen, the formula (6.18) has the same structure as the *Penalised Weighted Least Squares* estimator with weights $w_n,\ n=1,\ldots,N$. These weights as well as values for $\lambda_i=v_i^{-2},\ i=1,\ldots,d$ can be obtained via solution of the system (6.13). In the next chapter we will formulate an optimization problem that can

be solved instead of the system (6.13). In fact, we will show that w_n, $n = 1, \ldots, N$ and v_i^{-2}, $i = 1, \ldots, d$ can be obtained via the *Iteratively Reweighted Regularized PLASM* procedure.

Chapter 7
New Form of Regularized PLADM

This chapter will focus on derivation of an optimization problem that can be solved instead of the system (6.13) and, consequently, instead of the original regularized PLADM optimization problem (6.6).

7.1 New PLADM Optimization Problem

In this section we will introduce a new optimization problem and derive Kuhn-Tucker conditions for it. Let U be a matrix having the following structure

$$U = \begin{pmatrix} u_1 & & & & & & \\ & \ddots & & & & & \\ & & u_1 & & 0 & & \\ & & & \ddots & & & \\ & 0 & & & u_d & & \\ & & & & & \ddots & \\ & & & & & & u_d \\ & & & & & & & 1 \end{pmatrix}. \tag{7.1}$$

So, its diagonal is comprised of $d+1$ groups of identical values, the i-th group having as many entries u_i as there are regression coefficients in the i-th group of the vector γ defined in (6.3). Note that the last $(d+1)$-st group is made up of the intercept β_0 and the corresponding quantity on the diagonal of the matrix U is equal to one. Consider the following optimization problem

$$\begin{aligned}
\underset{\mathbf{u},\mathbf{w}}{\text{minimize}} \quad & -\mathbf{y}^T W^{-1} B (B^T W^{-1} B + U^{-1} K_0)^{-1} B^T W^{-1} \mathbf{y} \\
& + \sum_{i=1}^{d} \frac{\alpha_2}{u_i} + \mathbf{y}^T W^{-1} \mathbf{y} + \sum_{n=1}^{N} \frac{\alpha_1}{w_n} + \frac{1}{4} \sum_{n=1}^{N} w_n, \\
\text{subject to} \quad & \sum_{i=1}^{d} u_i \leq t', \\
& u_i \geq 0, \quad i = 1, \ldots, d, \\
& w_n \geq 0, \quad n = 1, \ldots, N.
\end{aligned} \tag{7.2}$$

where $t' \in (0, \infty)$ is a free parameter. The Lagrangian for this problem is as follows

$$\begin{aligned}
L = \; - \; & \mathbf{y}^T W^{-1} B (B^T W^{-1} B + U^{-1} K_0)^{-1} B^T W^{-1} \mathbf{y} + \sum_{i=1}^{d} \frac{\alpha_2}{u_i} + \mathbf{y}^T W^{-1} \mathbf{y} + \\
+ \; & \sum_{n=1}^{N} \frac{\alpha_1}{w_n} + \frac{1}{4} \sum_{n=1}^{N} w_n + \nu \Big(\sum_{i=1}^{d} u_i - t' \Big) - \sum_{i=1}^{d} \sigma_i u_i - \sum_{n=1}^{N} \delta_n w_n.
\end{aligned}$$

Here ν, σ_i, $i = 1, \ldots, d$ and δ_n, $n = 1, \ldots, N$ are Lagrange multipliers. Due to the presence of the terms $1/u_i$ and $1/w_n$ in its objective function, the problem (7.2) cannot have any solutions on the boundaries $u_i = 0$ and $w_n = 0$. Consequently, due to complementarity conditions, multipliers σ_i, $i = 1, \ldots, d$ and δ_n, $n = 1, \ldots, N$ are equal to zero and, therefore, can be ignored in further considerations. The Kuhn-Tucker conditions

$$\partial L / \partial u_i = 0, \; i = 1, \ldots, d$$

for the problem (7.2) are as follows

$$-\mathbf{y}^T W^{-1} B (B^T W^{-1} B + U^{-1} K_0)^{-1} I_i u_i^{-2} K_0 (B^T W^{-1} B + U^{-1} K_0)^{-1} B^T W^{-1} \mathbf{y} - \frac{\alpha_2}{u_i^2} + \nu = 0,$$

or, in a different form,

$$\mathbf{y}^T W^{-1} B (B^T W^{-1} B + U^{-1} K_0)^{-1} I_i K_0 (B^T W^{-1} B + U^{-1} K_0)^{-1} B^T W^{-1} \mathbf{y} + \alpha_2 = \nu u_i^2, \tag{7.3}$$

for $i = 1, \ldots, d$. Similarly, the rest of the of the Kuhn-Tucker conditions $\partial L / \partial w_n = 0,\ n = 1, \ldots, N$ are

$$\begin{aligned} - \quad & \mathbf{y}^T W^{-1} B (B^T W^{-1} B + U^{-1} K_0)^{-1} B^T I_n B w_n^{-2} (B^T W^{-1} B + U^{-1} K_0)^{-1} B^T W^{-1} \mathbf{y} \\ + \quad & 2 \mathbf{y}^T I_n w_n^{-2} B (B^T W^{-1} B + U^{-1} K_0)^{-1} B^T W^{-1} \mathbf{y} - y_n^2 w_n^{-2} - \frac{\alpha_1}{w_n^2} + \frac{1}{4} = 0,\ n = 1, \ldots, N \end{aligned}$$

where I_n is a matrix whose entries are zeros except the n-th entry on the diagonal which is equal to one. The last group of equations can be recast as

$$-\frac{1}{w_n^2} [y_n - \mathbf{B}_n (B^T W^{-1} B + U^{-1} K_0)^{-1} B^T W^{-1} \mathbf{y}]^2 - \frac{\alpha_1}{w_n^2} + \frac{1}{4} = 0,\ n = 1, \ldots, N,$$

or

$$[y_n - \mathbf{B}_n (B^T W^{-1} B + U^{-1} K_0)^{-1} B^T W^{-1} \mathbf{y}]^2 + \alpha_1 = \frac{w_n^2}{4},\ n = 1, \ldots, N, \tag{7.4}$$

where $\mathbf{B}_n$ is the n-th row of the model matrix B. By combining equations (7.3), (7.4) as well as a complementarity condition corresponding to the inequality constraint $\sum_{i=1}^d u_i \leq t'$, one obtains the following system

$$\begin{aligned} \mathbf{y}^T W^{-1} B C I_i K_0 C B^T W^{-1} \mathbf{y} + \alpha_2 &= \nu u_i^2, \\ (y_n - \mathbf{B}_n C B^T W^{-1} \mathbf{y})^2 + \alpha_1 &= \frac{w_n^2}{4}, \\ \nu (\sum_{i=1}^d u_i - t') &= 0, \end{aligned}$$

$$\begin{aligned} n = 1, \ldots, N, \quad w_n &\geq 0, \\ i = 1, \ldots, d, \quad u_i &\geq 0, \end{aligned} \tag{7.5}$$

where $C = (B^T W^{-1} B + U^{-1} K_0)^{-1}$.

7.2 Properties of the New PLADM Optimization Problem

Our approach to the investigation of properties of the new optimization problem (7.2) is very similar to that utilized in the chapter 3. Specifically, we will show that the objective function of (7.2) is strictly convex for $w_n > 0$, $n = 1, \ldots, N$ and $u_i > 0$, $i = 1, \ldots, d$. This, combined with the fact that the feasible set of the problem is also convex, implies that the problem (7.2) (and, consequently, the system (7.5)) has a unique solution. However, before we embark on this route, we will establish a technical result contained in the following Lemma.

Lemma 7.2.1 *For $w_n > 0$, $n = 1, \ldots, N$ and $u_i > 0$, $i = 1, \ldots, d$ the following relation holds*

$$-\mathbf{y}^T W^{-1} B (B^T W^{-1} B + U^{-1} K_\sigma)^{-1} B^T W^{-1} \mathbf{y} + \mathbf{y}^T W^{-1} \mathbf{y} = \mathbf{y}^T (W + B U K_\sigma^{-1} B^T)^{-1} \mathbf{y}, \tag{7.6}$$

where K_σ, $\sigma \in (0, \infty)$ is a matrix of the form

$$K_\sigma = \begin{pmatrix} K_1 & & & \\ & \ddots & & \\ & & K_d & \\ & & & \sigma \end{pmatrix}. \tag{7.7}$$

Proof. For the sake of convenience, we dispose of K_σ and U in the left-hand side of the relation (7.6) and introduce a new matrix $\hat{B}^T = U^{\frac{1}{2}} K_\sigma^{-\frac{1}{2}} B^T$ so that

$$\mathbf{y}^T W^{-1} B (B^T W^{-1} B + U^{-1} K_\sigma)^{-1} B^T W^{-1} \mathbf{y} = \mathbf{y}^T W^{-1} \hat{B} (\hat{B}^T W^{-1} \hat{B} + I)^{-1} \hat{B}^T W^{-1} \mathbf{y}.$$

Note that the above operation is legitimate as the matrix K_σ is nonsingular. Consider $\mathbf{y}^T v W^{-1} \hat{B} (\hat{B}^T v W^{-1} \hat{B} + I)^{-1} \hat{B}^T v W^{-1} \mathbf{y}$ as a function of a positive scalar parameter v. If $v \in (0, v_0)$, where v_0 is small enough so that $||\hat{B}^T v W^{-1} \hat{B}|| < 1$ and $||\hat{B}\hat{B}^T v W^{-1}|| < 1$, then

$$\begin{aligned} \mathbf{y}^T v W^{-1} \hat{B} (\hat{B}^T v W^{-1} \hat{B} &+ I)^{-1} \hat{B}^T v W^{-1} \mathbf{y} \\ &= \mathbf{y}^T v W^{-1} \hat{B} (I - \hat{B}^T v W^{-1} \hat{B} + \ldots) \hat{B}^T v W^{-1} \mathbf{y} \\ &= \mathbf{y}^T v W^{-1} (\hat{B}\hat{B}^T v W^{-1} - \hat{B}\hat{B}^T v W^{-1} \hat{B}\hat{B}^T v W^{-1} + \ldots) \mathbf{y} \\ &= \mathbf{y}^T v W^{-1} (I - (I + \hat{B}\hat{B}^T v W^{-1})^{-1}) \mathbf{y}. \end{aligned}$$

Next we will show that the identity above holds for all positive v. In order to prove this we consider $\mathbf{y}^T z W^{-1} \hat{B} (\hat{B}^T z W^{-1} \hat{B} + I)^{-1} \hat{B}^T z W^{-1} \mathbf{y}$ and $\mathbf{y}^T z W^{-1} (I - (I + \hat{B}\hat{B}^T z W^{-1})^{-1}) \mathbf{y}$ as functions of a complex parameter z such that $\mathrm{Re}(z) > 0$. We have that

$$\phi_1(z) = \mathbf{y}^T z W^{-1} \hat{B} (\hat{B}^T z W^{-1} \hat{B} + I)^{-1} \hat{B}^T z W^{-1} \mathbf{y}$$

and

$$\phi_2(z) = \mathbf{y}^T z W^{-1} (I - (I + \hat{B}\hat{B}^T z W^{-1})^{-1}) \mathbf{y}$$

are analytical functions of z which coincide for all z such that $\mathrm{Im}(z) = 0$, $\mathrm{Re}(z) \in (0, v_0)$, where v_0 is defined above. Therefore, due to the well-known result of the theory of functions of complex variables, one can conclude that $\phi_1(z)$ and $\phi_2(z)$ coincide for all z such that $\mathrm{Re}(z) > 0$. Setting z equal to one and replacing $\hat{B}^T$ with $U^{\frac{1}{2}} K_\sigma^{-\frac{1}{2}} B^T$, we arrive at

the relation

$$\begin{aligned} -\mathbf{y}^T W^{-1} B(B^T W^{-1} B &+ U^{-1} K_\sigma)^{-1} B^T W^{-1} \mathbf{y} + \mathbf{y}^T W^{-1} \mathbf{y} \\ &= -\mathbf{y}^T W^{-1} (I - (I + BUK_\sigma^{-1} B^T W^{-1})^{-1}) \mathbf{y} + \mathbf{y}^T W^{-1} \mathbf{y} \\ &= \mathbf{y}^T (W + BUK_\sigma^{-1} B^T)^{-1} \mathbf{y}. \ \square \end{aligned}$$

Note that

$$g_\sigma(u, w) \to g_0(u, w),$$

as $\sigma \to 0$, where

$$\begin{aligned} g_\sigma(u, w) &= \mathbf{y}^T W^{-1} B(B^T W^{-1} B + U^{-1} K_\sigma)^{-1} B^T W^{-1} \mathbf{y}, \\ g_0(u, w) &= \mathbf{y}^T W^{-1} B(B^T W^{-1} B + U^{-1} K_0)^{-1} B^T W^{-1} \mathbf{y}, \end{aligned}$$

where K_0 is defined in (6.11). Therefore, the convexity of the function $g_\sigma(u, w)$ implies that $g_0(u, w)$ is also convex. In turn, this means that, due to the presence of the strictly convex term $\sum_{n=1}^{N} \alpha_1 / w_n$, the objective function of (7.2) is strictly convex. Thus, in order to prove the strict convexity of the objective function in (7.2), it suffices to establish the convexity of $g_\sigma(u, w)$ which is the focus of the following Proposition.

Proposition 7.2.1 *For any $\sigma > 0$ the function*

$$-\mathbf{y}^T W^{-1} B(B^T W^{-1} B + U^{-1} K_\sigma)^{-1} B^T W^{-1} \mathbf{y} + \mathbf{y}^T W^{-1} \mathbf{y}$$

is convex in the domain $w_n > 0,\ n = 1, \ldots, N$ and $u_i > 0,\ i = 1, \ldots, d$.

Proof. According to the Lemma 7.2.1, it suffices to show that the function

$$f(w,u) = \mathbf{y}^T(W + BUK_\sigma^{-1}B^T)^{-1}\mathbf{y} \tag{7.8}$$

is convex for $w_n > 0,\ n = 1,\ldots,N$ and $u_i > 0,\ i = 1,\ldots,d$. In order to prove this, we will evaluate the Hessian of (7.8) and demonstrate that it is a positive semidefinite matrix. The first order derivatives of the function $f(w,u)$ are

$$\begin{aligned} f'_{w_n} &= -\mathbf{y}^T(W + BUK_\sigma^{-1}B^T)^{-1}I_n(W + BUK_\sigma^{-1}B^T)^{-1}\mathbf{y}, \\ f'_{u_i} &= -\mathbf{y}^T(W + BUK_\sigma^{-1}B^T)^{-1}BI_iK_\sigma^{-1}B^T(W + BUK_\sigma^{-1}B^T)^{-1}\mathbf{y}. \end{aligned}$$

The second order derivatives are

$$\begin{aligned} f''_{w_nw_m} &= 2\mathbf{y}^TCI_nCI_mC\mathbf{y}, \\ f''_{u_iu_i} &= 2\mathbf{y}^TCBI_iK_\sigma^{-1}B^TCBI_iK_\sigma^{-1}B^TC\mathbf{y}, \\ f''_{u_iw_n} &= 2\mathbf{y}^TCBI_iK_\sigma^{-1}B^TCI_nC\mathbf{y}. \end{aligned}$$

where $C = (W + BUK_\sigma^{-1}B^T)^{-1}$. Note that C is a positive definite matrix.

Let $\tilde{\mathbf{z}}_1$ and $\tilde{\mathbf{z}}_2$ be arbitrary d- and n-dimensional vectors respectively and $\tilde{\mathbf{z}}$ be their concatenation $[\tilde{\mathbf{z}}_1^T\tilde{\mathbf{z}}_2^T]^T$. Denoting the Hessian of (7.8) with respect to $(\mathbf{u},\mathbf{v})$ by F, the quadratic form $\tilde{\mathbf{z}}^TF\tilde{\mathbf{z}}$ can be cast as

$$\begin{aligned} \tilde{\mathbf{z}}^TF\tilde{\mathbf{z}} &= 2\mathbf{y}^TCBZ_1K_\sigma^{-1}B^TCBZ_1K_\sigma^{-1}B^TC\mathbf{y} \\ &\quad +2\mathbf{y}^TCZ_2CZ_2C\mathbf{y} + 4\mathbf{y}^TCBZ_1K_\sigma^{-1}B^TCZ_2C\mathbf{y}, \end{aligned}$$

where matrices Z_1 and Z_2 have the same structure as M and W defined in (6.12) and (6.11) respectively.

$$\tilde{z}^T F \tilde{z} = 2\mathbf{y}^T C(BZ_1 K_\sigma^{-1} B^T + Z_2) C(BZ_1 K_\sigma^{-1} B^T + Z_2) C\mathbf{y} \geq 0$$

since $BZ_1 K_\sigma^{-1} B^T + Z_2$ is a symmetric matrix. Thus, the Hessian F of the function (7.8) is at least a positive semidefinite matrix. □

As its objective function is strictly convex, the optimization problem (7.2) has a unique solution and, therefore, so does the system (7.5). The constraint $\sum_{i=1}^{d} u_i \leq t'$ in (7.2) is always active because the gradient of the objective function with respect to the variables u_i, $i = 1, \ldots, d$ does not vanish to zero (in fact, as can be seen from the formula (7.9), all of its components are negative) anywhere in the region $u_i > 0$, $i = 1, \ldots, d$, $w_n > 0$, $n = 1, \ldots, N$. Also, from the sensitivity theorem [35], it follows that the Lagrange multiplier ν corresponding to this constraint is always positive. The next result relates solutions of the Kuhn-Tucker conditions (7.5) for the optimization problem (7.2) with solutions of the Kuhn-Tucker conditions (6.13) of the regularized PLADM (6.6).

Proposition 7.2.2 *There is a one-to-one correspondence between solutions of the system (6.13), $t \in (t_l^{\alpha_{1,2}}, t_r^{\alpha_{1,2}})$ ($t_l^{\alpha_{1,2}}$ and $t_r^{\alpha_{1,2}}$ are defined in (6.7)) and solutions of the system (7.5), $t' \in (0, \infty)$.*

Proof. Given a value for the parameter $t \in (t_l^{\alpha_{1,2}}, t_r^{\alpha_{1,2}})$ and the solution $v_i^2(t)$, $i = 1, \ldots, d$, $w_n(t)$, $n = 1, \ldots, N$ and $\lambda(t)$ of (6.13), one can construct a solution $u_i(t')$, $i = 1, \ldots, d$, $w_n(t')$, $n = 1, \ldots, N$ and $\xi(t')$ of the system (7.5) according to the following formula

$$\begin{aligned} u_i(t') &= v_i^2(t), \quad i = 1, \ldots, d, \\ w_n(t') &= w_n(t), \quad n = 1, \ldots, N, \\ \nu(t') &= \frac{\lambda(t)^2}{4}, \\ t' &= 2t/\lambda(t). \end{aligned} \tag{7.9}$$

If $t_1 \neq t_2$ and $t_1' = t'(t_1)$, $t_2' = t'(t_2)$, then $t_1' \neq t_2'$ since, otherwise, that would mean that the system (7.5) and, consequently, the optimization problem (7.2) have two different solutions for some $t' = t_1' = t_2'$ which contradicts to the uniqueness of the solution for (7.2)

for any $t' \in (0, \infty)$.

Given a value of the parameter $t' \in (0, \infty)$ and the corresponding solution $u_i(t')$, $i = 1, \ldots, d$, $w_n(t')$, $n = 1, \ldots, N$ and $\nu(t')$ of (7.5), one can construct a solution $v_i^2(t)$, $i = 1, \ldots, d$, $w_n(t)$, $n = 1, \ldots, N$ and $\lambda(t)$ of the system (6.13)

$$\begin{aligned} v_i^2(t) &= u_i(t'), \quad i = 1, \ldots, d, \\ w_n(t) &= w_n(t'), \quad n = 1, \ldots, N, \\ \lambda(t) &= 2\sqrt{\nu(t')}, \\ t &= t'\sqrt{\nu(t')} \end{aligned} \tag{7.10}$$

of the system (6.13). Note that $\lambda(t)$ and

$$\gamma(t) = (B^T W^{-1}(t) B + v^{-2}(t) K_0)^{-1} B^T W^{-1}(t) \mathbf{y} \tag{7.11}$$

satisfy the Kuhn-Tucker conditions (6.14) for the regularized PLADM optimization problem (6.6) and, therefore, $\gamma(t)$ is the only solution of (6.6). Due to $\lambda(t) > 0$, the constraint of the problem (6.6) is active and $t \in (t_l^{\alpha_{1,2}}, t_r^{\alpha_{1,2}})$.

If $t_1' \neq t_2'$ and $t_1 = t(t_1')$, $t_2 = t(t_2')$, then $t_1 \neq t_2$ since, otherwise, that would mean that the system (6.13) has two different solutions for some $t = t_1 = t_2$ which is impossible.

Thus, there is a one-to-one correspondence between solutions of the systems (6.13) and (7.5) defined by the formula (7.9) (or by (7.10)). □

In summary, we established that, instead of solving the regularized PLADM optimization problem (6.6) for $t \in (t_l^{\alpha_{1,2}}, t_r^{\alpha_{1,2}})$, one can now deal with the new optimization problem (7.2), $t' \in (0, \infty)$. Having computed $\mathbf{u}(t')$, $\mathbf{w}(t')$ and the Lagrange multiplier $\nu(t')$ for a given t', one can obtain the corresponding solution $\gamma^T(t) = (\beta^T(t), \beta_0(t))$ of PLADM (6.6) for $t = t'\sqrt{\nu(t')}$ according to the formula

$$\gamma(t) = (B^T W^{-1}(t') B + U^{-1}(t') K_0)^{-1} B^T W^{-1}(t') \mathbf{y}. \tag{7.12}$$

Note that, in order to determine the optimal value for the parameter t, one usually has to obtain solutions of PLADM (6.6) for a number of values for t spread over the whole range $(t_l^{\alpha_{1,2}}, t_r^{\alpha_{1,2}})$. This, according to the previous result, can be accomplished via solution of the new optimization problem (7.2) for a number of values for t' from the range $(0, \infty)$. Thus, one does not have to worry about relating parameters t and t' by $t = t'\sqrt{\nu(t')}$. The most optimal value for the parameter t' can be found, for instance, via minimization of the Generalized Cross-Validation criterion as described in the section 4.6.

As was mentioned earlier, the structure of the formula (7.12) suggests that the regularized PLADM based on the new optimization problem (7.2) can be regarded as a method for estimating appropriate values for weights w_n, $n = 1, \ldots, N$ and smoothing parameters λ_i, $i = 1, \ldots, d$ in the Penalized Weighted Least Squares procedure (PWLS). According to this procedure, the regression coefficients are estimated via minimization of the following function

$$\gamma_{\text{pwls}} = \underset{\gamma}{\text{argmin}} \; (\mathbf{y} - B\gamma)^T W (\mathbf{y} - B\gamma) + \sum_{i=1}^{d} \lambda_i \beta_i^T K_i \beta_i. \tag{7.13}$$

Straightforward calculations show that the solution of the above problem can be found as

$$\gamma_{\text{pwls}} = (B^T W B + \Lambda K_0)^{-1} B^T W \mathbf{y}, \tag{7.14}$$

where Λ has the same structure as M defined in (6.12). As can be seen, both (7.12) and (7.14) produce the same estimates if Λ in (7.14) is set to U^{-1}. Note that presence of the regularization parameter α_2 in (6.6) has the effect that none of the groups of γ in (7.12) are set to zero though less important groups are likely to be distinguished by smaller values of the quantities u_i, $i = 1, \ldots, d$.

It appears to be possible to dispose of the regularization parameter α_2 and relate solutions of the original PLADM (6.4) to the solutions of the optimization problem (7.2) with $\alpha_2 = 0$ using the same approach as applied to investigation of PLASM. However, due to illustrative character of this chapter and apparent unsuitability of PLADM for Data

Mining purposes, we would like to stop our investigation of PLADM at this point. To conclude this chapter, we will consider an algorithm which, in principle, can be used to obtain regularized PLADM solutions.

7.3 PLADM and the Iteratively Reweighted PLASM

In this section we will discuss a procedure that can be used to solve the optimization problem (7.2). This problem can be recast in the following way:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & F[\mathbf{u}(\mathbf{w})] + \mathbf{y}^T W^{-1} \mathbf{y} + \sum_{n=1}^{N} \frac{\alpha_1}{w_n} + \frac{1}{4} \sum_{n=1}^{N} w_n \\ \text{subject to} \quad & w_n \geq 0, \quad n = 1, \ldots, N, \end{aligned} \tag{7.15}$$

where

$$\begin{aligned} F[\mathbf{u}(\mathbf{w})] \quad = \quad & \min_{\mathbf{u}} \left[-\mathbf{y}^T W^{-1} B (B^T W^{-1} B + U^{-1} K_0)^{-1} B^T W^{-1} \mathbf{y} + \sum_{i=1}^{d} \frac{\alpha_2}{u_i} \right] \\ & \text{subject to} \quad \sum_{i=1}^{d} u_i \leq t', \\ & \quad u_i \geq 0, \quad i = 1, \ldots, d. \end{aligned} \tag{7.16}$$

Based on this formulation we propose the following algorithm to solve (7.2):

1. Start with the initial weights $\mathbf{w}_0$. Set $k = 0$.
2. Solve the optimization problem (7.16) for $\mathbf{u}$ with $\mathbf{w} = \mathbf{w}_k$ kept fixed. As can be seen, this amounts to solution of the new regularized PLASM optimization problem (3.1) and can be carried out using the procedure outlined in the section 4.5. Set $\mathbf{u}_{k+1} = \mathbf{u}(\mathbf{w}_k)$, where $\mathbf{u}(\mathbf{w}_k)$ is the solution of (7.16).
3. Compute the gradient $\mathbf{g}_k$ of the objective function in (7.15) at $\mathbf{w} = \mathbf{w}_k$.
4. Use the gradient $\mathbf{g}_k$ to obtain $\mathbf{w}_{k+1}$

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \alpha_k \mathbf{g}_k,$$

where

$$\alpha_k = \min_{\alpha>0} Q(\mathbf{w}_k - \alpha \mathbf{g}_k),$$

and

$$Q(\mathbf{w}) = F[\mathbf{u}(\mathbf{w})] + \mathbf{y}^T W^{-1} \mathbf{y} + \sum_{n=1}^{N} \frac{\alpha_1}{w_n} + \frac{1}{4} \sum_{n=1}^{N} w_n.$$

5. Set $k = k + 1$. Go to the step (2) unless a convergence criterion is satisfied.

This is essentially a first order method based on a steepest decent. Better convergence properties could be obtained via the direct application of the Sequential Quadratic Programming Algorithm to the solution of the problem (7.2). However, the advantage of using the abovecited algorithm is that, given the software which solves (3.1), it is relatively easy to implement. The only difficulty is the evaluation of the gradient of the objective function in (7.15). The following Proposition deals with this issue.

Proposition 7.3.1 *The gradient of the objective function in (7.15) can be evaluated according to the formula:*

$$\frac{\partial Q(\mathbf{w})}{\partial w_n} = -\frac{1}{w_n^2} r_n^2 - \frac{\alpha_1}{w_n^2} + \frac{1}{4}, \; n = 1, \ldots, N,$$

where r_n is the n-th component of the vector of residuals $\mathbf{r}$ *defined as*

$$\mathbf{r} = \mathbf{y} - B(B^T W^{-1} B + U^{-1} K_0)^{-1} B^T W^{-1} \mathbf{y}.$$

Proof. According to the formulae (7.3) and (7.4), we have the following expression for $\partial Q(\mathbf{w})/\partial w_n$:

$$- \sum_{i=1}^{d} [\mathbf{y}^T W^{-1} B (B^T W^{-1} B + U^{-1} K_0)^{-1} I_i u_i^{-2} K_0 (B^T W^{-1} B + U^{-1} K_0)^{-1} B^T W^{-1} \mathbf{y}$$

$$- \frac{\alpha_2}{u_i^2}]u'_{iw_n} - \frac{1}{w_n^2}[y_n - \mathbf{B}_n(B^TW^{-1}B + U^{-1}K_0)^{-1}B^TW^{-1}\mathbf{y}]^2 - \frac{\alpha_1}{w_n^2} + \frac{1}{4}, \quad (7.17)$$

where u'_{iw_n} is a partial derivative of the i-th component of the solution $\mathbf{u}(\mathbf{w})$ of (7.16) with respect to the weight w_n. Since $\mathbf{u}(\mathbf{w})$ is a solution of (7.16) the Kuhn-Tucker conditions must hold for $i = 1, \ldots, d$:

$$-\mathbf{c}^T I_i u_i^{-2} K_0 \mathbf{c} - \frac{\alpha_2}{u_i^2} + \nu = 0,$$

where $\mathbf{c} = (B^TW^{-1}B + U^{-1}K_0)^{-1}B^TW^{-1}\mathbf{y}$ and ν is a Lagrange multiplier. This implies that

$$-\sum_{i=1}^{d}[\mathbf{c}^T I_i u_i^{-2} K_0 \mathbf{c} - \frac{\alpha_2}{u_i^2}]u'_{iw_n} + \nu \sum_{i=1}^{d} u'_{iw_n} = 0.$$

As was mentioned earlier, the inequality constraint in (7.16) is always active. Therefore,

$$\frac{\partial}{\partial w_n} \sum_{i=1}^{d} u_i = \sum_{i=1}^{d} u'_{iw_n} = 0,$$

and

$$\sum_{i=1}^{d}[\mathbf{c}^T I_i u_i^{-2} K_0 \mathbf{c} - \frac{\alpha_2}{u_i^2}]u'_{iw_n} = 0.$$

Thus,

$$\frac{\partial Q(\mathbf{w})}{\partial w_n} = -\frac{1}{w_n^2}[y_n - \mathbf{B}_n(B^TW^{-1}B + U^{-1}K_0)^{-1}B^TW^{-1}\mathbf{y}]^2 - \frac{\alpha_1}{w_n^2} + \frac{1}{4}. \quad \square$$

Note that, unlike the new PLASM optimization problem (3.25), the problem (7.2) contains even more parameters than there are datapoints in a dataset. This means that PLADM (at least in the form (7.2)) does not appear to be a suitable tool for analyzing large datasets ($>$ 10000 datapoints).

Chapter 8
Multivariate Adaptive Regression Splines

As we pointed out in chapter 1, the Multivariate Adaptive Regression Splines algorithm developed by J. Friedman can be regarded as one of the most successful large scale exploratory regression tools suitable for solution of Data Mining problems. We start our exploration of MARS with an overview of the original procedure as presented in [19] and [20]. The algorithm selects a relatively small subset from a complete set of tensor product spline functions constructed using truncated power basis functions

$$(\pm(x-t))_+^q \tag{8.1}$$

where t is a *knot* chosen from the set of data values of a predictor x. The typical j-th basis function generated by MARS can be written as

$$T_j(\mathrm{x}) = \prod_{k=1}^{p_j} [s_{kj}(x_{v(k,j)} - t_{kj})]_+^q. \tag{8.2}$$

Here p_j is the number of factors in a tensor product basis function; s_{kj} is a binary indicator taking on two values $+1, -1$; $x_{v(k,j)}$ is a predictor variable associated with the k-th factor, and t_{kj} is a knot location on that variable. Note that the predictors $x_{v(k,j)}$, $k = 1, \ldots, p_j$ involved in the j-th basis function are all distinct. The exponent q determines the order of the spline approximation. In fact, Friedman's implementation of MARS is based on

the approximation of the first order ($q = 1$). Basically, the set of basis functions of the type (8.2) is produced using two procedures: The first one (forward stepwise procedure) is concerned with generation of a model containing a large number of basis functions (8.2) and, probably, overfitting the data. The second one (backward elimination procedure) removes suboptimal basis functions from the model produced by the first procedure so that the resulting model is, in a sense, optimal. The next section is dedicated to a detailed discussion of these two algorithms.

8.1 Friedman's MARS

The forward stepwise procedure utilizes a recursive strategy. The initial model is comprised of one basis function

$$T_0(\mathbf{x}) = 1. \tag{8.3}$$

Assume that J iterations have been carried out. As we will see shortly, this results in $2J + 1$ tensor product basis functions (8.2) being added to the model

$$\{T_j(\mathbf{x})\}_0^{2J}. \tag{8.4}$$

Each iteration of the forward stepwise procedure of MARS is similar in nature to a step of the standard forward subset selection (FSS) algorithm [39]: the model is augmented with basis functions whose addition ensures the greatest improvement of the fit. However, FSS considers all available tensor product basis functions as potential candidates for inclusion and, generally, there are many of them. For example, given six numeric variables with ten univariate basis functions per variable, the corresponding full set of tensor product basis functions would contain 10^6 members. In contrast to the FSS strategy, MARS deals only with those basis functions that can be generated from the functions already in the model. Specifically, during the $(J + 1)$-st iteration MARS adds two new basis functions to the model [20]

$$\begin{aligned} T_{2J+1}(\mathbf{x}; l, v, t) &= T_l(\mathbf{x})[+(x_v - t)]_+^q, \\ T_{2J+2}(\mathbf{x}; l, v, t) &= T_l(\mathbf{x})[-(x_v - t)]_+^q. \end{aligned} \tag{8.5}$$

Thus, functions in (8.5) are products of one of the basis functions $T_l(\mathbf{x})$, $l = 0, \ldots, 2J$ included in the model earlier (called *parent*) and some univariate truncated power basis function defined by its argument x_v (a predictor variable not presented in $T_l(\mathbf{x})$) as well as a knot location t on that variable. In order to ensure that functions (8.5) are the optimal ones, the parameters l, v, t are taken to be the minimizers of the residual sum of squares for the model

$$\begin{aligned} (l^*, v^*, t^*, \{a_j\}_0^{2J+2}) = \arg \min_{l,v,t,\{a_j\}_0^{2J+2}} \sum_{n=1}^{N} \{y_n - \sum_{j=0}^{2J} a_j T_j(\mathbf{x}_n) - \\ a_{2J+1} T_{2J+1}(\mathbf{x}; l, v, t) - a_{2J+2} T_{2J+2}(\mathbf{x}; l, v, t)\}^2. \end{aligned} \tag{8.6}$$

Note that the nature of MARS allows for a straightforward incorporation of restrictions on the level of interaction between predictor variables. For example, if one allows only the constant function (8.3) to serve as a parent for new basis functions, then the resulting model will be additive as it will be comprised of the tensor product basis functions (8.2) with $p_j = 1$.

The strategy outlined above results in the computational work required to carry out one step of MARS being considerably less in comparison with that of the FSS procedure and this reduction is especially dramatic in high-dimensional settings where the number of elements in the full set of tensor product basis functions is astronomical. Of course, the price to be paid for a speedup is, possibly, an inferior quality of models produced by the forward stepwise procedure of MARS.

The forward stepwise procedure is allowed to produce models containing a relatively large $J_{\max}$ number of basis functions and, probably, overfitting the data. The necessity to generate a large model stems from that fact that each iteration of MARS results in the derivation of new basis functions from the basis functions contained in the current model. This means that the basis functions produced during early steps of the procedure may

turn out to be much less optimal compared to those obtained later. The only contribution of the earlier functions may be to serve as parents for later (more optimal) basis functions. Thus, we specify $J_{\max}$ to be large enough so as to enable more complex (and, probably, more important) basis functions to be added to the model. Subsequently, during the backward elimination procedure, the less important basis function are discarded. This can be accomplished via a standard backward subset selection procedure [39], [45], [46], where basis functions are regarded as a set of variables from which an optimal subset has to be selected. The subset of basis functions is chosen such that the least squares fit using these functions leads to a low value of an estimate of the Prediction Error (1.4). The Prediction Error can be estimated via the Generalized Cross-Validation score 1.7. The GCV score corresponding to a subset of J basis functions is evaluated as

$$\text{GCV} = N\frac{\sum_{n=1}^{N}[y_n - \hat{f}(\mathbf{x}_n)]^2}{[N - \text{df}]^2}, \quad \text{df} = hJ, \tag{8.7}$$

where $\hat{f}$ is obtained via the least squares fit in the space spanned by these basis functions. Note that the number of degrees of freedom in (8.7) is taken to be hJ which is different from the number of degrees of freedom of an ordinary linear model ($\text{df} = J$). The reason for this is that MARS selects basis functions rather than using prespecified ones and, therefore, the number of degrees of freedom associated with each basis function produced by MARS is greater than 1. The parameter h can be interpreted as a smoothing parameter of the algorithm. Larger values result in fewer basis functions being allowed to stay in the model thereby producing smoother estimates. In general, h has to be chosen via, for example, minimization of the Cross-Validation criterion (1.5) but the choice $h = 3$ was reported in [20] to perform well in a wide variety of situations. Having built the final set of basis functions, MARS determines the regression coefficients $\{a_0, a_j,\ j = 1, \ldots, J\}$ of the model

$$\hat{f}(\mathbf{x}) = a_0 + \sum_{j=1}^{J} a_j \prod_{k=1}^{p_j}[s_{kj}(x_{v(k,j)} - t_{kj})]_+^q, \tag{8.8}$$

via an ordinary linear least squares fit of (8.8) to data.

One of the tasks of Data Mining is to gain an understanding of the structure of underlying functions. However, the model in the form (8.8) is unlikely to provide this kind of insight. In order to circumvent this difficulty, it is proposed in [19] to regroup the terms in (8.8) so that the structure of the MARS model would become more transparent. This can be done in the following way

$$\hat{f}(x) = a_0 + \sum_{p_j=1} f_i(x_i) + \sum_{p_j=2} f_{ij}(x_i, x_j) + \cdots . \tag{8.9}$$

The first sum in (8.9) is comprised of the terms that involve only one variable ($p_j = 1$) and each univariate function $f_i(x_i)$ in this sum is a weighted (weights being the corresponding regression coefficients) sum of univariate basis functions having x_i as their argument. Similarly, the second sum is made up of the terms involving only two variables ($p_j = 2$), where each $f_{ij}(x_i, x_j)$ is a weighted sum of bivariate tensor product basis functions having both x_i and x_j as their arguments. This regrouping can be continued until all basis functions have been assigned to some group. As can be seen, the model in the form (8.9) is much more convenient to interpret. Indeed, it highlights which variables are involved in the model as well as the character of their participation.

In the discussion above we assumed that all of the variables are numerical ones. However, in practice, so-called categorical variables taking on discrete unordered values (like colour or gender) can also be encountered. It turns out [19] that the strategy of the MARS algorithm can be easily adjusted to deal with such variables: one has to use univariate indicator functions instead of truncated power basis functions (8.1)

$$I(x \in A), \tag{8.10}$$

where A is a subset of the set of all possible values for the categorical variable x. The most optimal subset A (which plays the same role as the knot for a truncated power basis function) can be produced via either enumerating all possible subsets (this could be very computationally expensive) or by building it in the manner very similar to that used by the ordinary forward stepwise regression procedure [39]: the algorithm starts with the

subset A containing no categories and it progresses by adding categories to A one at a time in such a way so as to achieve the largest reduction in the value of the residual sum of squares (RSS) at each step. The algorithm stops if further addition of categories does not result in a significant decrease in the value of RSS. Although the approach does not necessarily produce an optimal subset it usually produces a reasonably good one. The univariate indicator functions (8.10) can be used to form new tensor product basis functions (compare with (8.5))

$$\begin{aligned} T_{2J+1}(\mathrm{x}, l, v, A) &= T_l(\mathrm{x}) I(x_v \in A), \\ T_{2J+2}(\mathrm{x}, l, v, A) &= T_l(\mathrm{x}) I(x_v \notin A), \end{aligned} \tag{8.11}$$

where values for l and v as well as a subset A are chosen to achieve the largest reduction in the residual sum of squares. This constitutes the only required change to the MARS algorithm to enable it to deal with categorical variables.

8.2 MARS algorithm based on B-splines

The MARS algorithm is based on truncated power basis functions. It is known that such a basis may lead to ill-conditioned systems linear of equations [12] whereas some other bases for representing spline approximations, such as, for instance, B-splines, have superior numerical properties [49]. In this section we describe the MARS-like algorithm that utilizes B-splines.

In principle, one could implement BMARS using B-splines of any order. However, we utilized B-splines of the second order [12]:

$$B_t(x_i) = (\lambda_t - \lambda_{t-2})[\lambda_{t-2}, \lambda_{t-1}, \lambda_t](\cdot - x_i)_+,$$

where $\lambda_{t-2}, \lambda_{t-1}, \lambda_t$ are knots and $[\lambda_{t-2}, \lambda_{t-1}, \lambda_t](\cdot - x_i)_+$ denotes the divided difference at $\lambda_{t-2}, \lambda_{t-1}, \lambda_t$ of the bivariate function $(s - x_i)_+$ with respect to its first variable [12], [14]. The reason for using this type of functions is two-fold: firstly, utilization of the simplest continuous B-splines significantly simplifies implementation of the BMARS algo-

rithm and, secondly, approximation with piecewise linear functions is more resistant to the so-called end-effects [20]. Thus, although they are constructed using different bases, models produced by MARS and BMARS belong to the same space of piecewise linear d-variate functions.

Assume that for each continuous predictor variable univariate B-splines having various sizes of support intervals (or scales as we will call them) are introduced. Such B-splines can be constructed based on a special sequence of sets of knots $\{S_l^i\}_{l=0}^L$, $i = 1, \ldots, d$, where L determines the number of distinct scales used to build regression models and it is specified by a user. Each set S_l^i contains $(2^l + 1)$ distinct knots placed either uniformly or at the $(2^{-l} \times 100)$-th percentiles of the marginal distribution of the i-th predictor variable (0- and 100-percentiles are included). For each set of knots S_l^i one can construct a set of univariate B-spline basis functions and, as can be seen, the scale of B-splines based on S_l^i decreases as the index l increases. Note that B-splines of the largest scale are, in fact straight lines.

We mentioned earlier that the forward stepwise procedure of MARS can be regarded as an approximation to the ordinary statistical forward subset selection (FSS) algorithm and this interpretation can be extended to BMARS as well. However, the way in which BMARS approximates the FSS algorithm is somewhat different from that of MARS. As was set out before, MARS selects its new basis functions from among the candidates derived according to the rule (8.5). Notice that univariate truncated powers corresponding to *all* available knot locations t on the particular predictor x_v are allowed to be the factors. The BMARS strategy goes even further in narrowing the set of candidates. Although it uses a rule similar to (8.5) to define the set of candidate basis functions, at each step it restricts the univariate factors to being B-splines of a certain scale. Of course, using B-splines of only one scale would result in a poor accuracy of the generated models. Therefore, the algorithm was enabled to use different scales at different steps.

Like the original MARS, our algorithm consists of two phases: a forward stepwise procedure intended to construct a model made up of a large number of basis functions (and probably overfitting the data), and a backward stepwise procedure which removes sub-optimal basis functions from the model produced at the previous stage. The forward procedure starts by following the MARS strategy with the only difference being that only B-splines of the largest available scale are allowed to form tensor product basis functions

$$T_j(\mathbf{x}) = \prod_{k=1}^{p_j} B_{t(k,j)}(x_{v(k,j)}), \tag{8.12}$$

where the predictors $x_{v(k,j)}$, $k = 1, \ldots, p_j$ involved in the j-th basis function are all distinct. To clarify this point consider what the $(J+1)$-st iteration does in this case: after the J-th iteration there are $J+1$ functions

$$\{T_j(\mathbf{x})\}_0^J \tag{8.13}$$

in the model, each of the form (8.12). The $(J+1)$-st iteration adds one new basis function

$$T_{J+1}(\mathbf{x}; l, v, s) = T_l(\mathbf{x}) B_s(x_v). \tag{8.14}$$

Here $T_l(\mathbf{x})$ is one of the $J+1$ already chosen basis functions (8.13), $0 \leq l \leq J$; x_v is one of the predictor variables not present in $T_l(\mathbf{x})$; and s labels univariate B-spline basis functions of the variable x_v and the current active scale. The three parameters l, v, s defining T_{J+1} are chosen such that they provide the largest reduction in the sum of squared residuals. Proceeding along these lines the algorithm is likely to reach a point where the approximating ability of B-splines of the current scale is exhausted. Indeed, such splines are able to approximate accurately only relatively narrow class of functions whose values do not change dramatically over regions of much smaller scale compared to the current one. In order to determine the most appropriate moment for changing over to the B-splines of smaller scale BMARS estimates the prediction accuracy of the current model using the Generalized Cross-Validation (GCV) criterion (8.7). When the GCV score ceases to decrease the algorithm changes over to the next smaller scale which implies that B-spline functions of that scale only will be allowed to participate in construction of new basis functions (8.14) (see Figure (8.1)). The algorithm proceeds in this manner until the size of the model exceeds a prespecified level. It is worth noting that, because each new basis function has been derived from an earlier basis function, B-splines of all scales may,

in principle, appear as factors in any basis function (8.12).

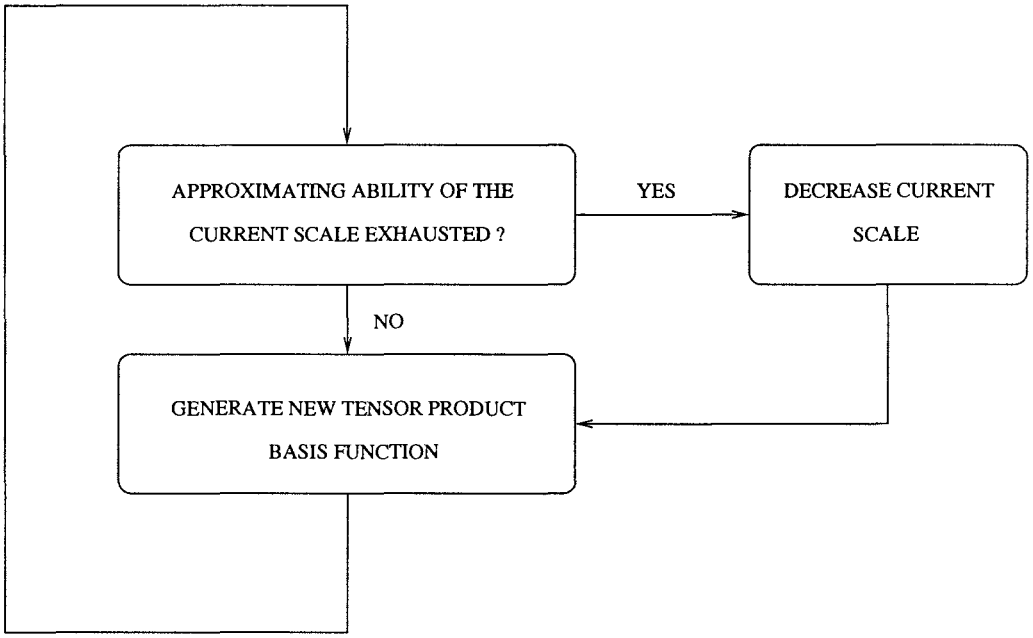


Figure 8.1: Modified forward stepwise procedure of BMARS.

The backward elimination procedure and the approach to handling categorical variables are similar to those utilized in the original MARS. As one of the anonymous referees of this thesis noticed, BMARS is not a translation invariant procedure. Indeed, if the underlying function happens to have curvature between two basis function of the current scale, the algorithm will try to deal with it when it switches over to a smaller scale. On the other hand, if the curvature happens to be near one of the basis functions, the approximation with B-splines of the current scale may prove to be adequate and the switch may not occur at all. This means that the way in which the model is built might change if one or more predictors were translated. To eliminate the practical implications of that, the algorithm scales as well as translates predictor vectors so that they are located in a unit hypercube prior to the analysis.

8.3 Computational Complexity of BMARS

In order to make the solution of Data Mining problems a practical exercise, one has to make sure that the tools used have a computational complexity proportional to the number of datapoints. In this section we demonstrate that this is the case with BMARS. For the

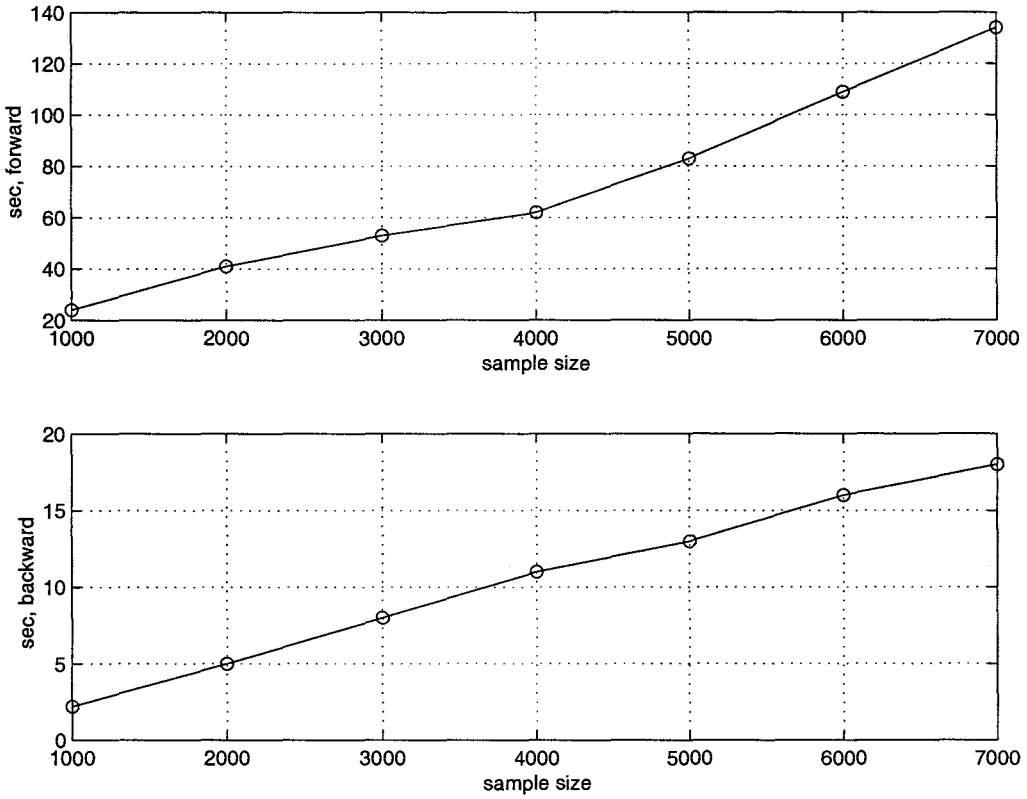


Figure 8.2: Complexities of the forward and backward parts of BMARS as functions of the size of a sample (dataset).

sake of simplicity, we assume that there is a fixed number $\tilde{K}$ of knots per numeric variable and a fixed number of categories $\tilde{C}$ per categorical variable.

As was set out before, in order to select a new tensor product basis function, the algorithm evaluates the goodness of each candidate basis function based on the reduction in the residual sum of squares, where the candidate basis functions are those produced according to the rule (see discussion in the previous section for more details)

$$\begin{aligned} T_{\text{cand}} &= T_{\text{prnt}} B(x), \qquad \text{if } x \text{ is a numeric variable,} \\ T_{\text{cand}} &= T_{\text{prnt}} I(x \in A), \text{ if } x \text{ is a categorical variable,} \end{aligned} \tag{8.15}$$

where T_{cand} is a candidate basis function, T_{prnt} is one of the basis functions contained in the current model (parent function), and $B(x)$ and $I(x \in A)$ are univariate factors used to modify the parent function. As will be shown in section 9.2, evaluation of the reduction in

the value of the residual sum of squares resulting from the inclusion of a candidate basis function (8.15) in a current model amounts to the computation of $(J+3)$ scalar products

$$\sum_{n=1}^{N} q_{jn} T_{\text{cand}}(\mathbf{x}_n), \quad j = 1, \ldots, J$$
$$\sum_{n=1}^{N} T_{\text{cand}}(\mathbf{x}_n)^2, \quad \sum_{n=1}^{N} y_n T_{\text{cand}}(\mathbf{x}_n),$$
$$\sum_{n=1}^{N} T_{\text{cand}}(\mathbf{x}_n), \tag{8.16}$$

where $\mathbf{q}_j$, $j = 1, \ldots, J$ are orthonormalized columns of the model matrix corresponding to J basis functions contained in a current model (see section 9.2). Although the dimensionality of the vector $[T_{\text{cand}}(\mathbf{x}_1), \ldots, T_{\text{cand}}(\mathbf{x}_N)]$ is equal to the size N of the dataset, the fact that B-splines $B(x)$ and indicator functions $I(x \in A)$ have the compact support property implies that a significant number of components of this vector will be equal to zero. Therefore, the cost of the computation of a single scalar product in (8.16) is roughly proportional[1] to $N/\tilde{K}$ or $N/\tilde{C}$ depending on the nature of the argument x in (8.15). The latter estimate can be explained as follows. Due to orthogonality of indicator functions $I(x = c_i)$ and $I(x = c_j)$, where c_i and c_j are distinct categories of a variable x, it suffices to compute scalar products (8.16) for a candidate basis function (8.15) formed using univariate factors $I(x = c_i)$, $i = 1, \ldots, \tilde{C}$ and the obtained results can be used to compute scalar products (8.16) for a candidate basis function (8.15) formed using arbitrary indicator function $I(x \in A)$. Thus, in order to estimate the computational cost of BMARS, one can assume that only the following indicator functions are used in (8.15): $I(x = c_i)$, $i = 1, \ldots, \tilde{C}$. Thus, the cost ($P_{\text{one}}^{\text{num}}$ or $P_{\text{one}}^{\text{cat}}$) of the computation of the reduction in the value of the residual sum of squares corresponding to the inclusion of a candidate basis function can be estimated as

$$P_{\text{one}}^{\text{num}} \sim (J+3)\frac{N}{\tilde{K}},$$
$$P_{\text{one}}^{\text{cat}} \sim (J+3)\frac{N}{\tilde{C}}.$$

[1] In fact, $\tilde{K}$ changes as the algorithm switches from scale to scale.

Therefore, computation of the reductions for all candidate basis functions is

$$P_{\text{all}} \sim J\left[d_{\text{num}}\tilde{K}P_{\text{one}}^{\text{num}} + d_{\text{cat}}\tilde{C}P_{\text{one}}^{\text{cat}}\right] = J(J+3)Nd,$$

where d_{num} and d_{cat} are the number of numeric and categorical variables respectively and $d = d_{\text{num}} + d_{\text{cat}}$. Thus, P_{all} is the cost of producing one new tensor product basis function. Hence, the cost of building a model comprised of $J_{\max}$ basis functions is as follows

$$P_{\text{total}} \sim J_{\max}^2(\alpha J_{\max} + \beta)Nd.$$

where α and β are some parameters independent of the parameters of the problem in hand (d, N etc). As can be seen, the complexity is linear in the number of datapoints as well as the number of predictor variables.

Finally, the cost of the backward elimination procedure of BMARS is, obviously, less than the cost of the forward part and in fact, it is a function of the number of basis functions $J_{\max}$ only. Thus, BMARS is suitable for efficient solution of large scale problems. The results of numerical simulations shown in the Figure (8.2) confirm the validity of the estimates above.

8.4 Parallel BMARS

Processing of large amounts of data is likely to be a very time and resources consuming exercise and, therefore it is imperative to pay close attention to development of parallel algorithms which are able to make a full use of modern computational systems. In this section we will consider a parallel version of the BMARS algorithm intended to run on a multiprocessor system with distributed memory. The forward stepwise part accounts for the bulk of the computational work carried out by BMARS and, therefore it will become the focus of our considerations.

We begin by pointing out that, according to the BMARS' strategy, the structure of each new basis function added to the model depends on structures of the previously generated

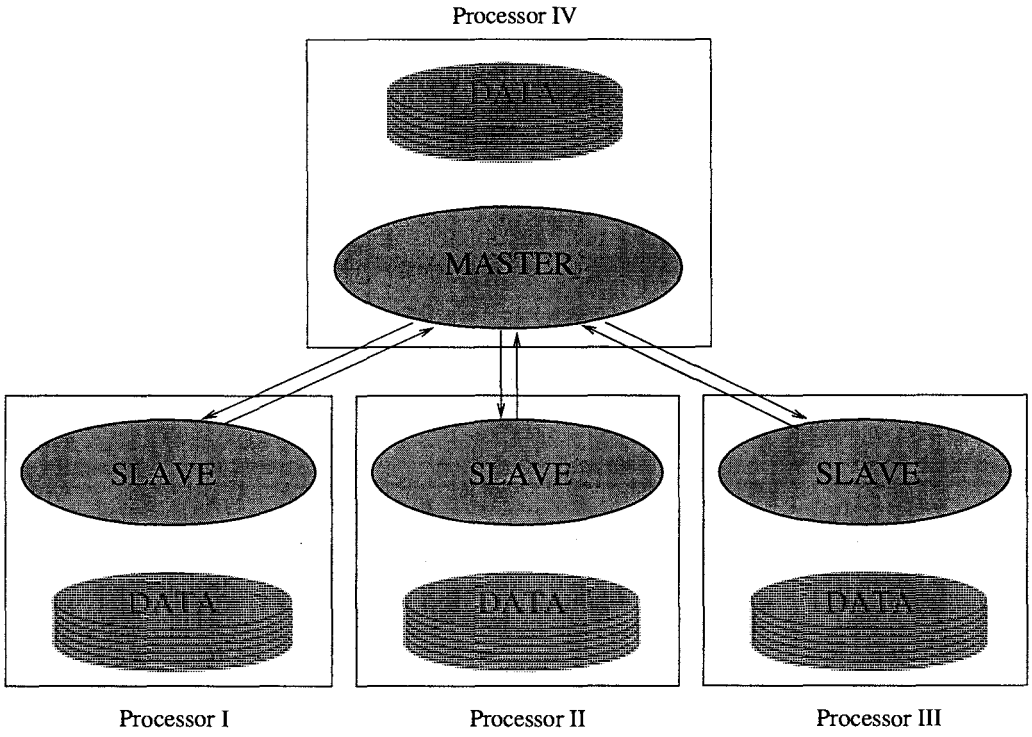


Figure 8.3: The diagram of the parallel BMARS.

ones. Thus, one cannot generate several basis functions in parallel. However, it turns out that it is quite feasible to come up with a scalable algorithm for generating a single basis function. Indeed, in order to generate a new tensor product basis function (8.14), one has to perform least squares fits for all appropriate values for the parameters l, v and s. So, the algorithm based on parallelization of each least squares fit would ensure a uniform distribution of the computational load among processors. Since almost any algorithm intended for performing a least squares fit amounts to computation of a number of scalar products [25], the approach based on data partitioning seems to be appropriate in this situation. Assuming that our system is comprised of p processors, the data partitioning involves the allocation of N/p records of the dataset to each processor of the system where the corresponding partial scalar products are computed[1].

A simple diagram of the parallel forward stepwise procedure running on the system comprised of four processors is shown in the Figure (8.3). As can be seen, it is based on the "master-slave" paradigm. All of the CPU's ("slaves") apart from the one called "master"

[1] It should be noted that, in order to ensure the uniform distribution of the computational load, data records have to be assigned to processors randomly since, otherwise, due to the compact support property of B-splines and indicator functions, it may happen that some of the processors would have to deal with subvectors comprised mostly of zeros while others would have to perform computations with dense subvectors.

run the identical code intended for computation of partial scalar products. Also, they store portions of the dataset having approximately the same size. In addition to the code for computation of scalar products, the "master" processor runs a program carrying out bookkeeping tasks: collecting information produced by other CPU's; adding new basis functions to the model; generating instructions for other processors etc.

We used the Parallel Virtual Machine (PVM) programming environment as a basis for our implementation of the parallel BMARS[1]. PVM enables a collection of heterogeneous computer systems to be viewed as a single parallel virtual machine with distributed memory and, therefore BMARS based on PVM is able to run on a variety of architectures. An example of application of the algorithm can be found in section 10.3. In our experiments we used a multiprocessor system with 10 SPARC processors and 4.75 Gbytes of shared memory.

[1] A short user's guide to the BMARS software can be found in the Appendix A.

Chapter 9

BMARS: Implementation Issues

9.1 Smoothing of BMARS Models

BMARS produces models using piecewise linear B-splines. Models of this type do not have continuous derivatives and graphs of the models' components may appear quite rough. To get around this problem one can utilize the approach suggested by Friedman [20] which, essentially, amounts to smoothing first order truncated power basis functions comprising tensor product basis functions (8.2)

$$b(x, s, t) = [s(x - t)]_+. \tag{9.1}$$

This can be done by replacing this function with a piecewise cubic function of the form

$$C(x|s = +1, t_-, t, t_+) = \begin{cases} 0 & x \leq t_-, \\ p_+(x - t_-)^2 + r_+(x - t_-)^3 & t_- < x < t_+, \\ x - t & x \geq t_+, \end{cases} \tag{9.2}$$

$$C(x|s = -1, t_-, t, t_+) = \begin{cases} -(x - t) & x \leq t_-, \\ p_-(x - t_+)^2 + r_-(x - t_+)^3 & t_- < x < t_+, , \\ 0 & x \geq t_+, \end{cases} \tag{9.3}$$

where

$$\begin{aligned} p_+ &= (2t_+ + t_- - 3t)/(t_+ - t_-)^2, \\ r_+ &= (2t - t_+ - t_-)^3, \\ p_- &= (3t - 2t_- + t_+)/(t_- - t_+)^2, \\ r_- &= (t_- + t_+ - 2t)/(t_- - t_+)^3. \end{aligned} \tag{9.4}$$

$C(x|s, t_-, t, t_+)$, $s = \pm 1$ are continuous and have continuous first order derivatives. They are characterised by three knots t_-, t_+, t whose locations are chosen so as to decrease the number of discontinuities of the second order derivatives. Graphs of a truncated power basis function and its smoothed piecewise cubic counterpart are shown in the Figure (9.1).

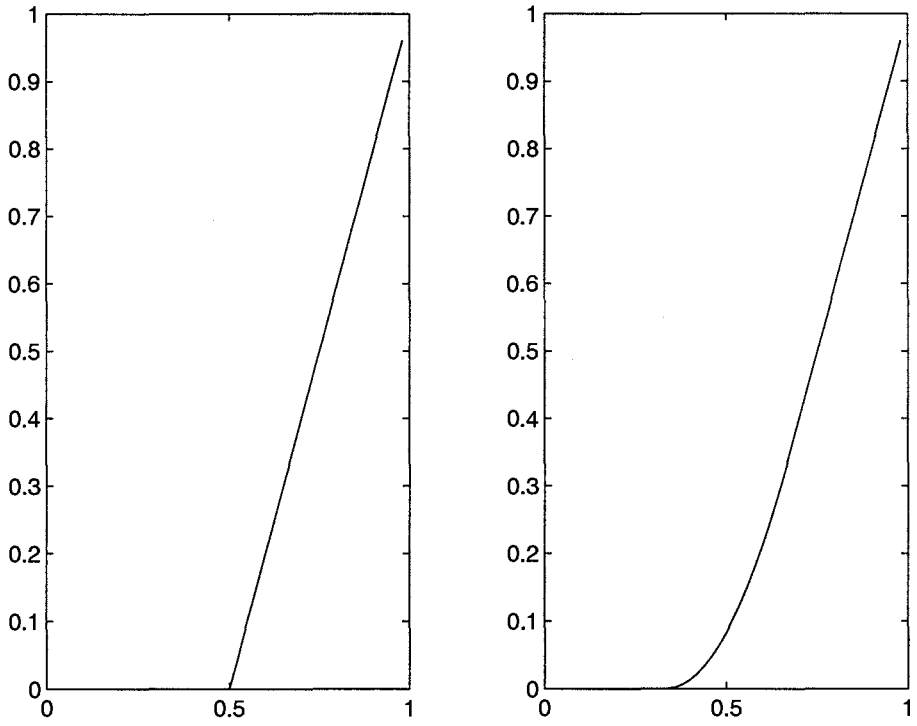


Figure 9.1: Smoothing of a truncated power basis function.

As every basis function in a MARS' model is a product of a certain number of univariate factors (truncated power basis functions), the replacement of the factor (9.1) by its piecewise cubic counterpart (9.2, 9.3) produces a set of smooth basis functions. The final model is obtained via a least squares fit of the piecewise cubic basis functions to data. The resulting model will have continuous first (but not second) order derivatives.

In order to use this approach for smoothing BMARS models, one can use the fact that

a B-spline function of the second order can be represented as a linear combination of the three truncated power basis functions. Thus, we can replace each of those truncated powers with an appropriate piecewise cubic function and thereby obtain a piecewise cubic model having continuous first order derivatives.

9.2 Least Squares Fit Procedure

A linear least squares (LSQ) fit is an important component of BMARS. Therefore, special attention has to be paid to an efficient and reliable implementation of the LSQ procedure. We employ a procedure based on the idea that was suggested in [20] (see the Rejoinder section). According to this approach, the columns of the design matrix corresponding to a current model should be kept orthonormal to each other. Specifically, let $Q_{J-1} = [\mathbf{q}_1, \ \ldots, \mathbf{q}_{J-1}]$ be a matrix obtained via orthonormalization of the columns of the model matrix produced as a result of $(J-1)$ iterations of the forward stepwise procedure and $\mathbf{s}_{J-1}$ be a vector with entries $s_j = \sum_{n=1}^{N} q_{jn} y_n, \ j = 1, \ldots, (J-1)$. Let $\mathbf{t}_J$ be a column corresponding to the (centered) basis function $T_J(\mathbf{x})$ selected as a result of the J-th iteration of the forward stepwise procedure. Then the matrix Q_{J-1} and the vector $\mathbf{s}_{J-1}$ are updated according to the formulae

$$Q_J = [Q_{J-1}, \mathbf{q}_J],$$

where

$$\mathbf{q}_J = \frac{\mathbf{t}_J - Q_{J-1} Q'_{J-1} \mathbf{t}_J}{||\mathbf{t}_J - Q_{J-1} Q'_{J-1} \mathbf{q}_J||},$$

and

$$\mathbf{s}_J^T = [\mathbf{s}_{J-1}^T, s_J]^T, \ s_J = \sum_{n=1}^{N} q_{Jn} y_n.$$

The matrix Q_J and the vector $\mathbf{s}_J$ play an important role when it comes to selection of a new tensor product basis function $T_{J+1}(\mathbf{x})$. Specifically, in order to carry out this task,

one has to compute the reduction in the value of the residual sum of squares for each candidate basis function (8.14). This can be done according to the following formula

$$R = \frac{[(\mathbf{t}_{J+1} - Q_J Q_J^T \mathbf{t}_{J+1})^T \cdot \mathbf{y}]^2}{||\mathbf{t}_{J+1} - Q_J Q_J^T \mathbf{t}_{J+1}||^2} = \frac{[\mathbf{t}_{J+1}^T \mathbf{y} - \mathbf{t}_{J+1}^T Q_J \mathbf{s}_J]^2}{||\mathbf{t}_{J+1} - Q_J Q_J^T \mathbf{t}_{J+1}||^2} = \frac{[\mathbf{t}_{J+1}^T \mathbf{y} - \mathbf{v}_J^T \mathbf{s}_J]^2}{\mathbf{t}_{J+1}^T \mathbf{t}_{J+1} - \mathbf{v}_J^T \mathbf{v}_J}, \tag{9.5}$$

where $\mathbf{t}_{J+1}$ is a column corresponding to the (centered) candidate function $T_{J+1}(\mathbf{x}; l, v, s)$ being tested and $\mathbf{v}_J = Q_J^T \mathbf{t}_{J+1}$. Thus, the evaluation of the reduction in the value of the residual sum of squares amounts to computation of (the most expensive to compute) $(J+3)$ scalar products involving vectors whose length is equal to the size of a dataset N:

$$\sum_{n=1}^{N} q_{jn} T_{J+1}(\mathbf{x}_n), \quad j = 1, \ldots, J$$
$$\sum_{n=1}^{N} T_{J+1}(\mathbf{x}_n)^2, \quad \sum_{n=1}^{N} y_n T_{J+1}(\mathbf{x}_n), \quad \sum_{n=1}^{N} T_{J+1}(\mathbf{x}_n).$$

The last scalar product is due to the necessity to center the vector $\mathbf{t}_{J+1}$ corresponding to the candidate basis function $T_{J+1}(\mathbf{x}; l, v, s)$.

9.3 Logistic Regression with Offset

So far we have been concerned with building models based on minimization of the residual sum of squares. However, as was mentioned in section 1.2, certain types of data require a different approach to regression analysis. In this section, we describe a technique that can be used to perform the so-called logistic regression [12] which is suitable for dealing with the situation where a response variable assumes only two values: for example, 0 and 1. In this case, it is inappropriate to try to fit a regression surface directly to response values as the model (1.1) is certainly not applicable. Instead, one can model the log-odds function $\log[p(\mathbf{x})/(1 - p(\mathbf{x}))]$, where $p(\mathbf{x})$ is the probability of a response variable y taking on the value of 1

$$\log[p(\mathbf{x})/(1-p(\mathbf{x}))] = \hat{f}(\mathbf{x}).$$

Here $\hat{f}(x)$ is the model of the type produced by the BMARS algorithm. The logistic regression problem can be solved using BMARS algorithm with the linear least squares fit replaced with the procedure performing linear logistic regression procedure which estimates regression coefficients associated with a set of tensor product basis functions via maximization of the so-called *log-likelihood* function (see below). However, this way of performing of logistic regression analysis turns out to be quite expensive. A compromise strategy seems to be able to provide a good approximation [20]: the tensor product basis functions are selected using the BMARS squared-error-based loss criterion (least squares fit), and only the regression coefficients associated with the final model are estimated using a linear logistic regression. The linear logistic regression can be performed using the *Fisher Scoring* iterative fitting procedure [37]. In this section we consider the so-called *linear logistic regression with offset.* The necessity for using an offset arises in many applications and an appropriate example will be given in a later section dedicated to the analysis of motor vehicle insurance data.

Given a set of tensor product basis functions $T_j(\mathbf{x})$, $j = 1, \ldots, J$, the respective regression coefficients $\mathbf{a}$ can be estimated via maximization of the following log-likelihood function

$$l(\mathbf{a}) = \sum_{n=1}^{N} y_n \log(p(\mathbf{x}_n; \mathbf{a})) + (1 - y_n) \log(1 - p(\mathbf{x}_n; \mathbf{a})), \tag{9.6}$$

where

$$p(\mathbf{x}_n; \mathbf{a}) = \frac{\exp(\eta_n(\mathbf{x}_n; \mathbf{a}))}{1 + \exp(\eta_n(\mathbf{x}_n; \mathbf{a}))}, \quad \eta_n(\mathbf{x}_n; \mathbf{a}) = \sum_j a_j T_j(\mathbf{x}_n) + e_n. \tag{9.7}$$

The quantity e_n appearing in (9.7) is called an offset. It is convenient to rewrite (9.6) in the canonical form [37]:

$$l(\mathbf{a}) = \sum_{n=1}^{N} [y_n \eta_n(\mathbf{x}_n; \mathbf{a}) - b(\eta_n(\mathbf{x}_n; \mathbf{a}))], \tag{9.8}$$

where $\eta_n(\mathbf{x}_n; \mathbf{a}) = \ln(p(\mathbf{x}_n; \mathbf{a})/(1 - p(\mathbf{x}_n; \mathbf{a})))$ and

$$b(\eta_n(\mathbf{x}_n; \mathbf{a})) = \ln[1 + \exp(\eta_n(\mathbf{x}_n; \mathbf{a}))].$$

In order to determine the coefficients $\mathbf{a}$ maximizing the log-likelihood function one has to solve the following system of nonlinear equations

$$\nabla l(\mathbf{a}) = \mathbf{0}.$$

According to the Fisher Scoring approach [37], one can solve this system using a version of the Newton's algorithm [18]. The vector $\mathbf{a}$ obtained through least squares fitting of the model to the data can serve as a starting point for the procedure. Let $\mathbf{a}_k$ be the vector of regression coefficients obtained after k iterations. Then, the next approximation $\mathbf{a}_{k+1}$ is determined via solution of the following system of linear equations

$$K(\mathbf{a}_k)\mathbf{a}_{k+1} = K(\mathbf{a}_k)\mathbf{a}_k + \mathbf{u}(\mathbf{a}_k), \tag{9.9}$$

where $K(\mathbf{a}_k)$ and $\mathbf{u}(\mathbf{a}_k)$ are the Hessian matrix taken with an opposite sign and the gradient of the log-likelihood function respectively evaluated at $\mathbf{a}_k$. It is quite straightforward to obtain expressions for these quantities. The first order derivative of the log-likelihood function with respect to a_j is as follows

$$u_j(\mathbf{a}) = \frac{\partial l}{\partial a_j} = \sum_{n=1}^{N} (y_n - p(\mathbf{x}_n; \mathbf{a}))T_j(\mathbf{x}_n). \tag{9.10}$$

Similarly, the second order derivative takes the form

$$K_{ij}(\mathbf{a}) = -\frac{\partial^2 l}{\partial a_i \partial a_j} = \sum_{n=1}^{N} p(\mathbf{x}_n;\mathbf{a})(1-p(\mathbf{x}_n;\mathbf{a}))T_i(\mathbf{x}_n)T_j(\mathbf{x}_n). \tag{9.11}$$

Rewriting (9.9) in a more detailed form we obtain for $i = 1, \ldots, J$

$$(K(\mathbf{a}_k)\mathbf{a}_{k+1})_i = \sum_{n=1}^{N} T_i(\mathbf{x}_n)p(\mathbf{x}_n;\mathbf{a}_k)(1-p(\mathbf{x}_n;\mathbf{a}_k))\times$$

$$\times \left(\sum_{j=1}^{J} a_{jk}T_j(\mathbf{x}_n) + \frac{(y_n - p(\mathbf{x}_n;\mathbf{a}_k))}{p(\mathbf{x}_n;\mathbf{a}_k)(1-p(\mathbf{x}_n;\mathbf{a}_k))} \right). \tag{9.12}$$

The system of linear equations (9.12) has to be solved for $\mathbf{a}_{k+1}$ and this step has a simple interpretation: the coefficients $\mathbf{a}_{k+1}$ can be regarded as those obtained through the weighted linear least squares fit of the BMARS model to the new response vector $\mathbf{z}$ whose components are defined as

$$z_n = \sum_{j=1}^{J} a_{jk}T_j(\mathbf{x}_n) + \frac{(y_n - p(\mathbf{x}_n;\mathbf{a}_k))}{p(\mathbf{x}_n;\mathbf{a}_k)(1-p(\mathbf{x}_n;\mathbf{a}_k))}$$

with weights

$$w_n = p(\mathbf{x}_n;\mathbf{a}_s)(1-p(\mathbf{x}_n;\mathbf{a}_k)).$$

Thus, in this case each iteration of the Newton's algorithm can be viewed as a weighted least squares fit. The iterations are repeated until they fail to produce any significant improvement of the quality of the fit measured in terms of the *deviance* $D(\mathbf{a})$ [37]

$$D(\mathbf{a}) = 2\sum_{n=1}^{N} \left[y_n \ln \frac{1-p(\mathbf{x_n};\mathbf{a})}{p(\mathbf{x_n};\mathbf{a})} + \ln \frac{1}{1-p(\mathbf{x_n};\mathbf{a})} \right].$$

Chapter 10
Numerical Experiments with BMARS

This chapter is dedicated to a comparative study of the MARS and BMARS algorithms. It is based on a number of synthetic datasets as well as a large real-life dataset provided by the NRMA Insurance company. In order to measure the accuracy level of models built for synthetic datasets, we used the Scaled Mean Squared Error (SMSE) defined in (1.3).

10.1 Synthetic Datasets

In this section we deal with synthetic datasets generated using functions mentioned in the original paper on the MARS algorithm [20]. Based on these functions, a number of datasets of various sizes (N) and levels of the signal-to-noise ratio (SNR) were generated. For each value of N and SNR, fifty independent datasets were generated and both BMARS and MARS were applied to the data to produce regression models. On the basis of the simulations, the average SMSE's of the models and the corresponding standard deviations were computed. The covariates of each dataset were sampled from an appropriate (multivariate) uniform distribution and the corresponding response values were evaluated according to the formula

$$y_n = f(\mathbf{x}_n) + \epsilon_n, \quad n = 1, \ldots, N,$$

where $f(\mathbf{x})$ is a target function, and ϵ_n, $n = 1, \ldots, N$ are sampled from a normal zero mean distribution with such a variance so as to ensure a desired level of SNR. The parameter h in GCV (8.7) was set to three for MARS and seven for BMARS and the maximal level

of interactions among predictor variables was set to the values suggested in [20]. Here are the functions we used in our experiments (H^d denotes a unit d-dimensional hypercube).

1. The first function is defined over H^{10}

$$f(\mathbf{x}) = 0.1\exp(4x_1) + 4/(1+\exp(-20(x_2-0.5))) + 3x_3 + 2x_4 + x_5. \tag{10.1}$$

2. The following two functions $Z(\cdot)$ and $\phi(\cdot)$ are defined of the domain

$$\begin{aligned} r: &\quad 0 \le r \le 100 \text{ ohms}, \\ \omega: &\quad 40\pi \le \omega \le 560\pi, \\ c: &\quad 1 \le c \le 11 \text{ microfarads}, \\ l: &\quad 0 \le l \le 1 \text{ henries}, \end{aligned}$$

$$Z(r,\omega,l,c) = \left[r^2 + (\omega l - 1/\omega c)^2\right]^{\frac{1}{2}}, \tag{10.2}$$

$$\phi(r,\omega,l,c) = \tan^{-1}\left[\frac{\omega l - 1/\omega c}{r}\right]. \tag{10.3}$$

3. The next function is dependent on numeric variables x_3 and x_4 ($(x_3,x_4) \in H^2$) as well as categorical variables x_1 and x_2 each of which takes on two distinct values "e" and "o"

$$f(x_3,x_4) = \begin{cases} 0 & \text{if } x_1 = \text{e and } x_2 = \text{e}, \\ 2\sin(\pi x_3 x_4) & \text{if } x_1 = \text{o and } x_2 = \text{e}, \\ \cos(\pi x_3) + \exp(x_4) & \text{if } x_1 = \text{e and } x_2 = \text{o}, \\ 2\sin(\pi x_3 x_4) + \cos(\pi x_3) + \exp(x_4) & \text{if } x_1 = \text{o and } x_2 = \text{o}. \end{cases} \tag{10.4}$$

4. The last function we consider is defined as

$$f(\mathbf{x}) = 10\sin(\pi x_1 x_2) + 20(x_3-0.5)^2 + 10x_4 + 5x_5, \quad \mathbf{x} \in H^5. \tag{10.5}$$

The results of the simulations are presented in the Figures 10.1 - 10.5. Each diagram shows the results of the modelling corresponding to three different levels of the size of a dataset and a certain level of the signal-to-noise ratio. The triangles and circles represent the average values of SMSE (computed based on 50 independent datasets) for models produced by BMARS and MARS respectively and whiskers display ave(SMSE) $\pm \sigma_{SMSE}$ intervals.

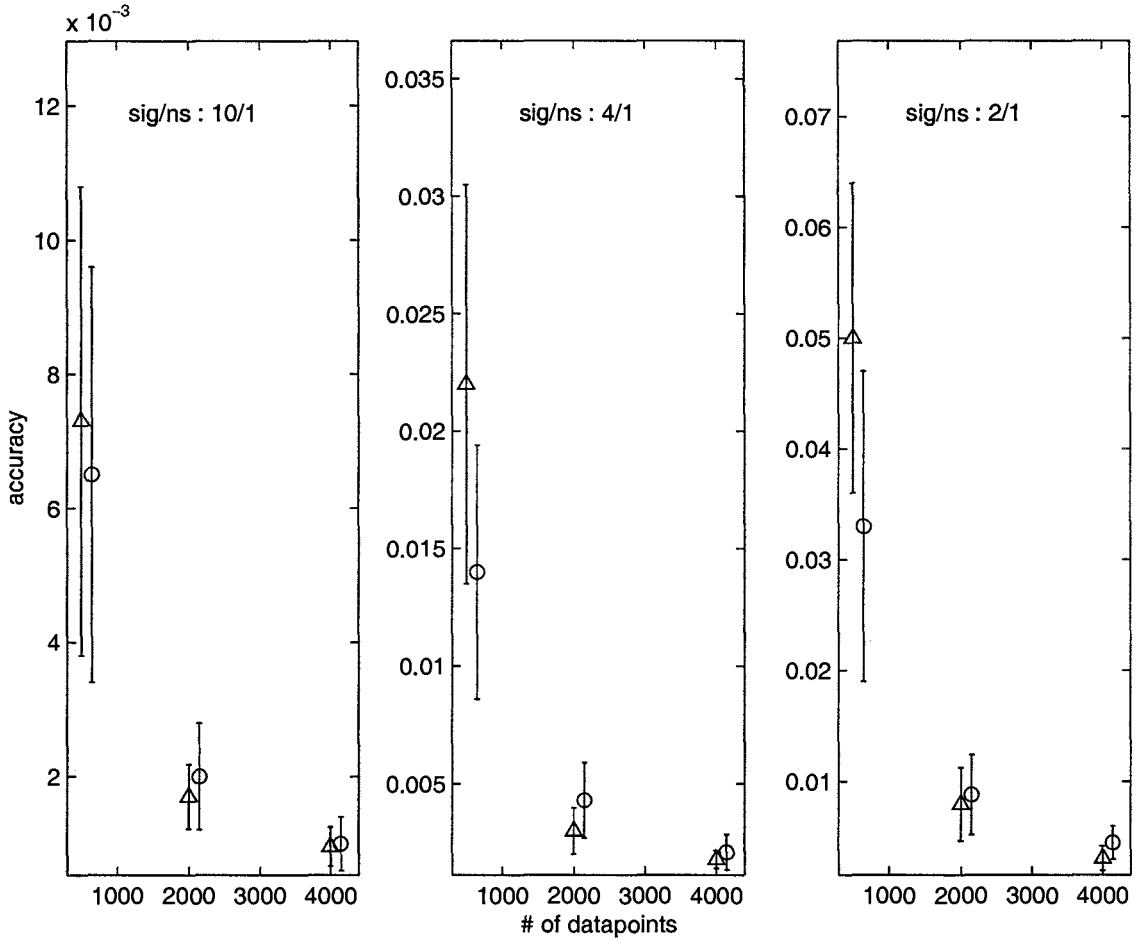


Figure 10.1: Average SMSE levels of models of the function (10.1) by MARS (circles) and BMARS (triangles) for various dataset sizes and signal-to-noise ratios (whiskers span ave(SMSE) $\pm \sigma_{SMSE}$ intervals).

The results of the simulations suggest that there is no considerable difference between accuracy levels of models by BMARS and MARS.

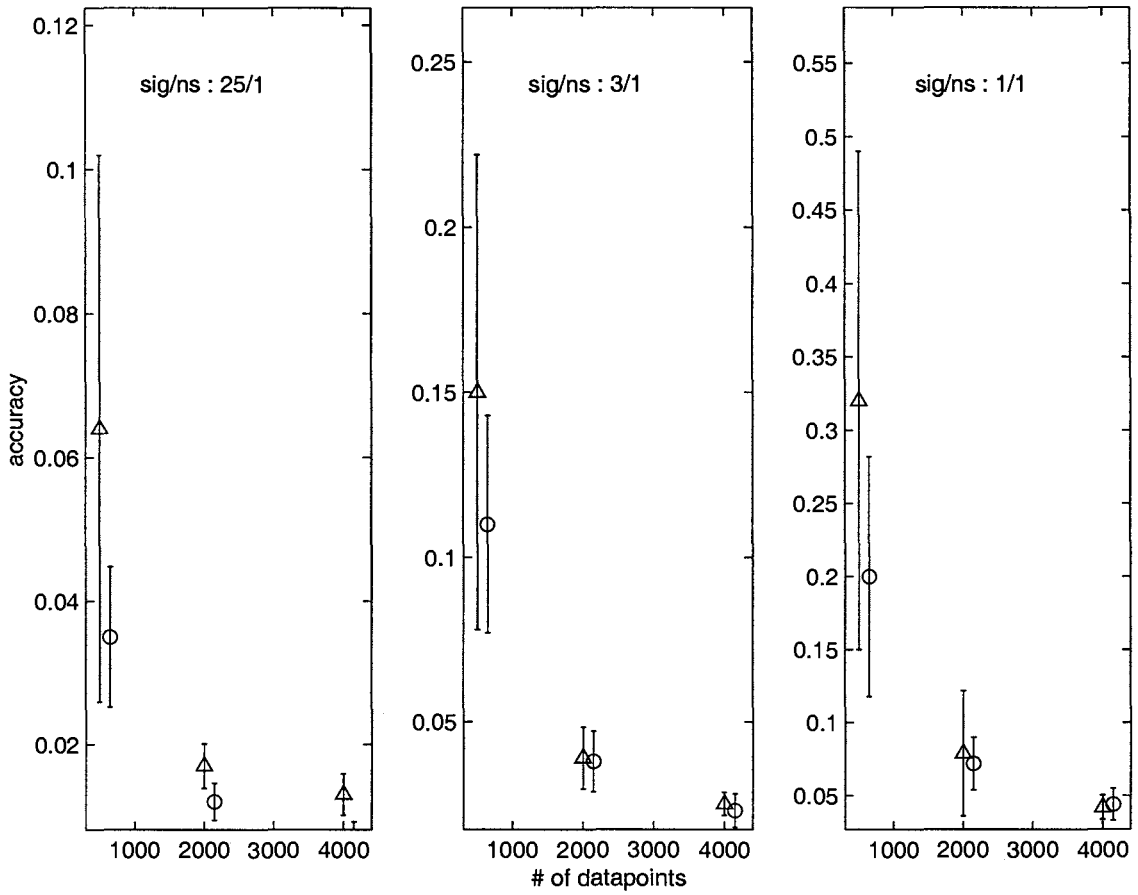


Figure 10.2: Average SMSE levels of models of the function (10.2) by MARS (circles) and BMARS (triangles) for various dataset sizes and signal-to-noise ratios (whiskers span ave(SMSE) $\pm \sigma_{SMSE}$ intervals).

10.2 Modelling "Hard" Dataset

This example is intended to demonstrate the advantage of using B-splines instead of truncated powers[1]. We generated a dataset using the following SAS program:

```
DATA test1;
ARRAY covs 10 x1-x10;
RPT: DO i = 1 TO 10;
covsi = RANGAM(75371,1)/5.0;
END;
resp = 0.1*EXP(4*x1) + 2.0*SIN(4.0*x2) + 3*x3 + 2*x4 + x5;
```

[1] The idea of this experiment was suggested by J. Friedman.

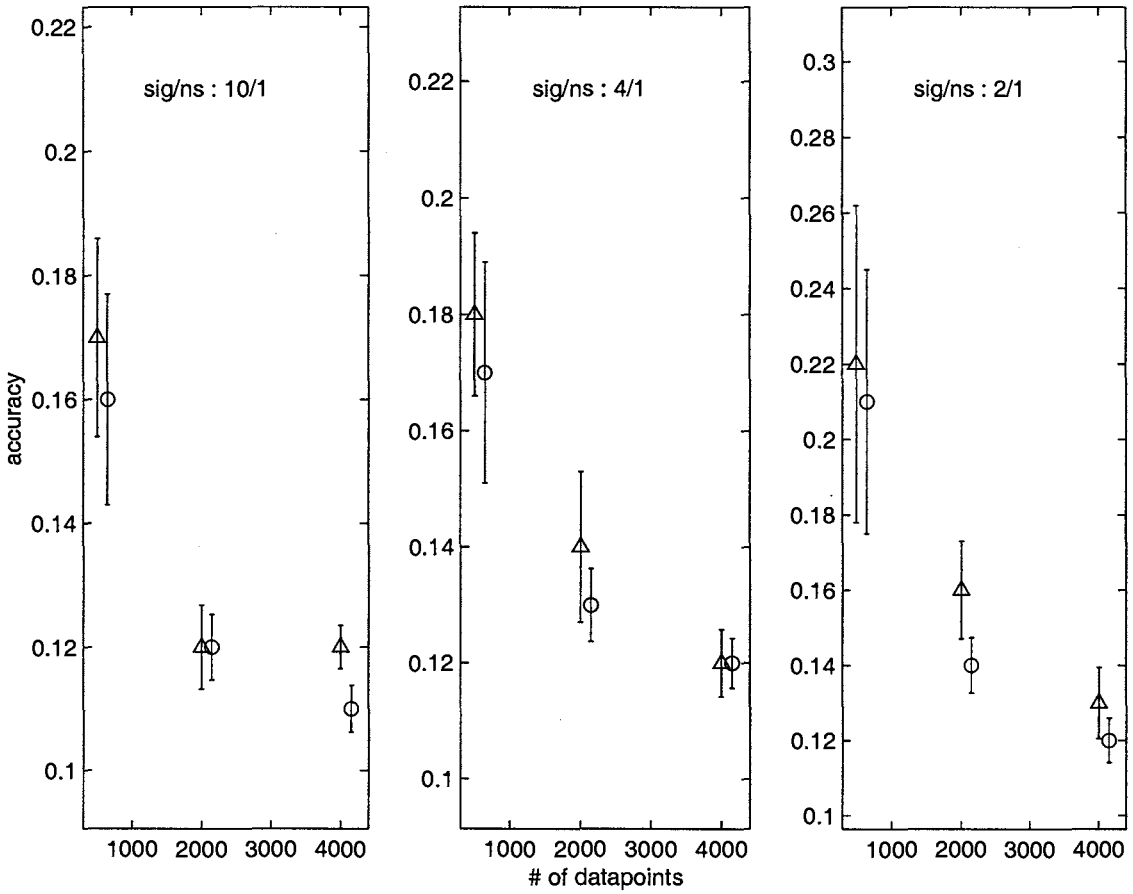


Figure 10.3: Average SMSE levels of models of the function (10.3) by MARS (circles) and BMARS (triangles) for various dataset sizes and signal-to-noise ratios (whiskers span ave(SMSE) $\pm \sigma_{SMSE}$ intervals).

```
OUTPUT;
KEEP x1-x10 resp;
j+1;
IF j < 5000 THEN GO TO RPT;
RUN;
```

All covariate values were sampled from a gamma distribution which means that the knots set at the percentiles of the marginal data distributions were distributed very unevenly. This, in turn means that the matrix of normal equations formed using truncated powers would probably be ill-conditioned. Both BMARS and MARS were used to build additive models and graphs of some of the univariate components estimated by the procedures are shown in the Figure (10.6). The wiggles interrupting the graphs of the curves produced by MARS are, apparently, due to the numerical instability caused by the truncated power

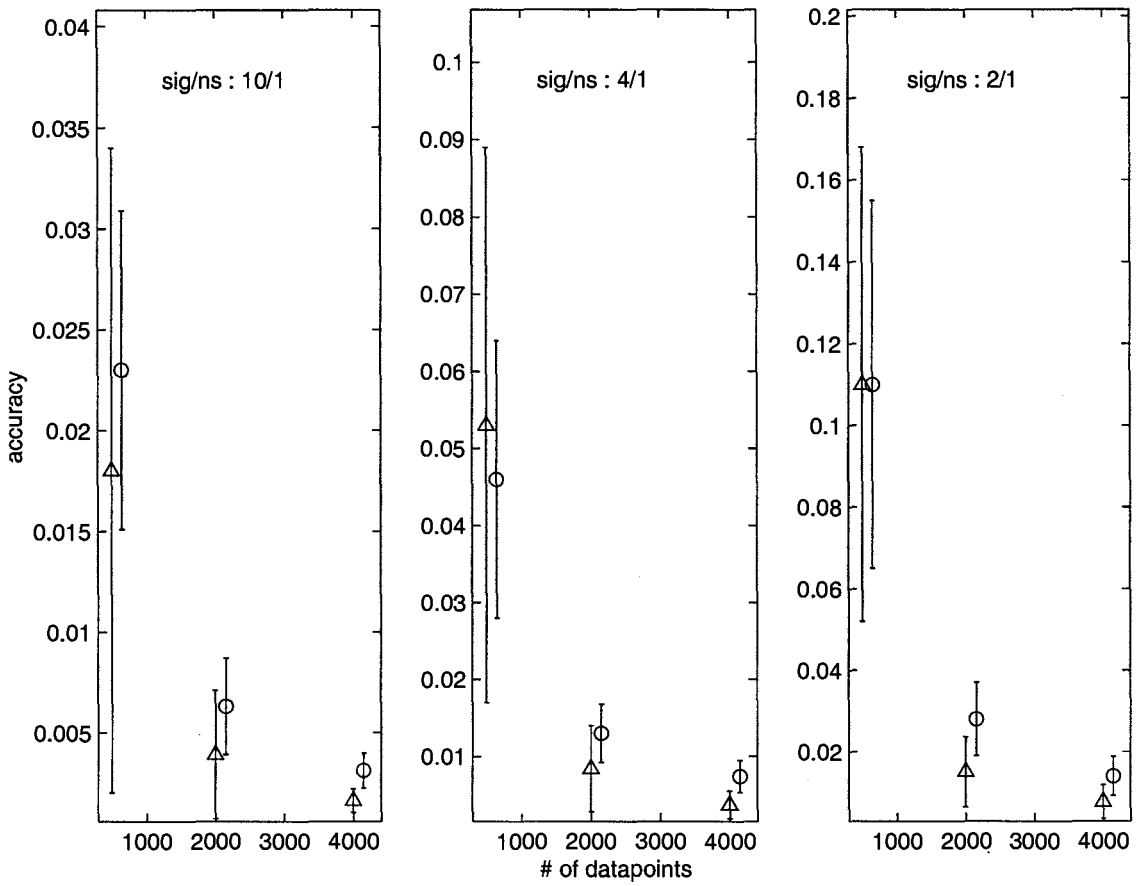


Figure 10.4: Average SMSE levels of models of the function (10.4) by MARS (circles) and BMARS (triangles) for various dataset sizes and signal-to-noise ratios (whiskers span ave(SMSE) $\pm \sigma_{SMSE}$ intervals).

basis functions.

10.3 Case Study: NRMA Claims Data

We tested the parallel BMARS on the dataset provided by the NRMA motor vehicle insurance company, Australia. The purpose of this analysis was to find a predictive model for the financial risk posed by each policy holder so that a premium setting strategy could be developed [2]. The dataset contained 1,601,277 records of which 131,995 were claims. Each record corresponded to a policy and contained values of 17 (most of them categorical) predictor variables as well as a response variable indicating the amount of money claimed by the policy holder. To model the financial risk r_n posed by the n-th policy holder, we used the following approach proposed in [53]. Let i_n be an indicator variable of a claim having been made by a policy holder and let c_n be the cost of a claim. The financial risk

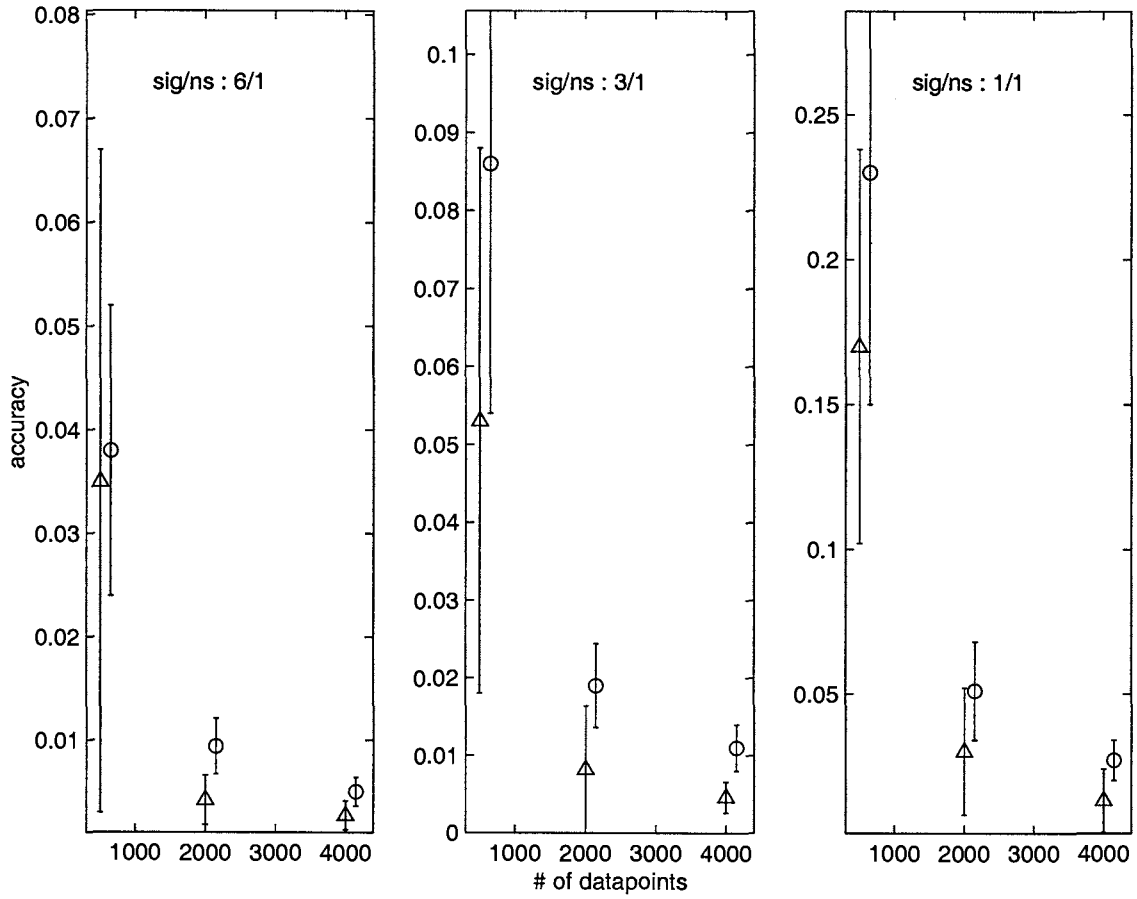


Figure 10.5: Average SMSE levels of models of the function (10.5) by MARS (circles) and BMARS (triangles) for various dataset sizes and signal-to-noise ratios (whiskers span ave(SMSE) $\pm\, \sigma_{SMSE}$ intervals).

can now be expressed as

$$r_n = c_n \cdot i_n, \tag{10.6}$$

where c_n and i_n can be modelled separately. We used the following models

$$c_n = f(\mathbf{x}_n) + \epsilon_n,$$

$$i_n \sim \text{Bernoulli}[p(\mathbf{x}_n)], \tag{10.7}$$

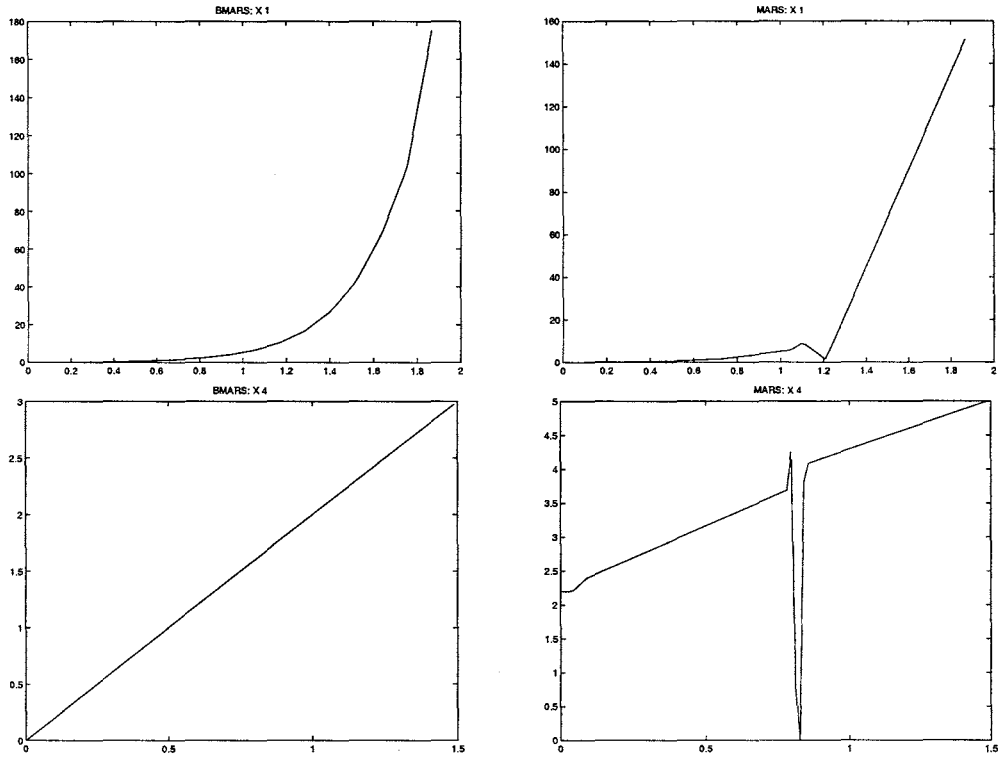


Figure 10.6: Modelling of a hard dataset in section (10.2).

where $\mathbf{x}_n$ denotes the covariate variables describing the n-th policy holder, and the independent identically distributed noise variables ϵ_n, $n = 1, \ldots, N$ are assumed to have zero mean. Taking expectations in the equation (10.6) yields

$$E(r_n) = f(\mathbf{x}_n) \cdot p(\mathbf{x}_n). \tag{10.8}$$

Thus in order to estimate the expected risk for a given policy we were required to estimate the expected cost of a claim given that a claim had been made and the probability of making a claim.

To model the cost of a claim given that a claim had been made, we extracted a smaller dataset from the NRMA data that was comprised only of records where a claim had occurred. Two types of models were generated using BMARS. One was a purely additive model and the other was a model allowing for the second order interactions between variables (two-way model). To compare models produced by BMARS with those produced by MARS, we used R^2 which measures the goodness of fit and which is defined as

$$R^2 = 1 - \frac{\sum_{n=1}^{N}(y_n - \hat{f}(\mathbf{x}_n))^2}{\sum_{n=1}^{N}(y_n - \bar{y})^2},$$

where N is the number of records in the dataset, y_n is the actual claim cost for the n-th policy holder, $\hat{f}(\mathbf{x}_n)$ is the cost evaluated by a model, and $\bar{y}$ is the average claim cost. R^2 is known to be a very poor estimator of the Prediction Error (1.4) of a fit (see, for instance, [39]). However, in order to be able to refer to the experimental results on MARS contained in [53], we had to use this quantity. The Table (10.1) contains R^2 for both MARS and BMARS claim cost models.

	MARS	BMARS
Additive	10.5%	10.5%
Two-way	12.8%	12.2%

Table 10.1: Goodness of fit measures for the MARS and BMARS claim cost models.

	True Claim Rate		True No Claim Rate	
	MARS	BMARS	MARS	BMARS
Additive	67.6%	67.3%	57.8%	58.9%
Two-way	67.6%	67.0%	58.0%	59.1%

Table 10.2: Classification rates for the MARS and BMARS claim probability models.

According to [53], a proper modelling of the claim probability requires utilization of the so-called logistic regression with offset (see section 9.3). The necessity to use an offset arises due to unequal lengths of exposure of policies to risk. As before, both additive and two-way models were generated with BMARS, and the respective claim ($\mathrm{R}_{\mathrm{claim}}$) and no claim ($\mathrm{R}_{\mathrm{noclaim}}$) classification rates

$$\mathrm{R}_{\mathrm{claim}} = \frac{\text{number of true } \texttt{claims} \text{ predicted as } \texttt{claims}}{\text{true number of claims in a dataset}},$$

$$\mathrm{R}_{\mathrm{noclaim}} = \frac{\text{number of true no } \texttt{claims} \text{ pred icted as no } \texttt{claims}}{\text{true number of 'no claims' in a dataset}},$$

for MARS and BMARS models are shown in the Table (10.2). The threshold probability of 0.08 was used to obtain these rates.

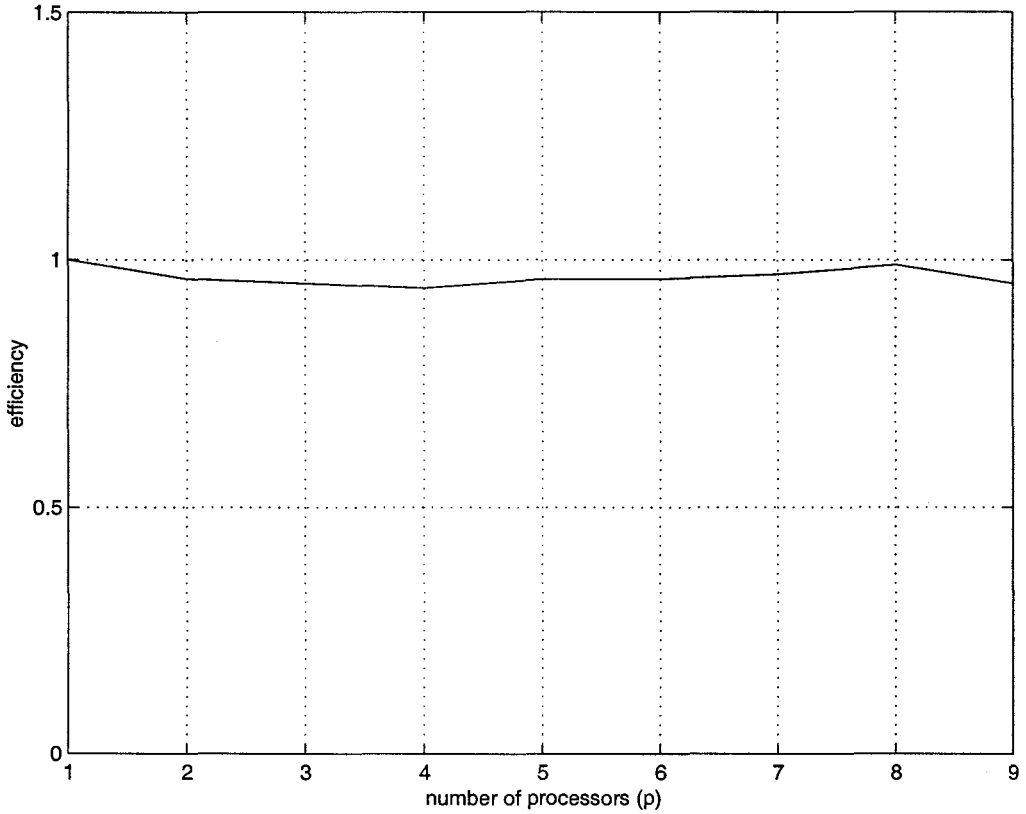


Figure 10.7: Efficiency of the parallel BMARS against the number of processors.

As the above results suggest, BMARS and MARS produced models of comparable levels of R^2 and classifications rates. The structures of the respective models were quite similar as well.

10.4 Scalability of the Parallel BMARS

To test performance of the parallel BMARS we carried out several runs of the algorithm each time generating an additive claim probability model and engaging a different number of processors. The graph in the Figure (10.7) displays the dependence of an efficiency of the algorithm on the number of processors involved. The parallel efficiency is defined as

$$\text{efficiency} = \frac{T_1}{T_p \cdot p},$$

where p is the number of processors involved, and T_1 and T_p are execution times on one and p processors respectively. Considering that the efficiency level for an ideal algorithm would be equal to unity, one can conclude that the parallel BMARS is quite an efficient scalable algorithm.

It should be mentioned that it took BMARS $\sim$ 3.5 hours to produce a model on one processor while MARS performed a similar task in $\sim$ 7 hours [53]. The reduction in computational cost can be attributed to the scale-by-scale strategy used in BMARS. Indeed, at each step our algorithm allows functions from a relatively small subset of all possible univariate B-splines (in particular, B-splines of one particular scale only) to form tensor product basis functions whereas MARS always considers the entire set of all eligible candidates. Of course, the speed-up of BMARS is even greater on a number of processors more than one. For instance, the execution time on 9 processors was $\sim$ 0.4 hours.

Chapter 11
Convergence of a Greedy Algorithm

In spite of the unquestionable success of MARS in a variety of situations, very little is known about the convergence properties of the algorithm. Due to the high level of sophistication of MARS, it is quite difficult to analyze the original procedure. Instead, we will consider an Adaptive Least Squares (ALS) algorithm which is a close relative of MARS but, on the other hand, is more amenable to theoretical study. It is based on the so-called Greedy Approximation strategy [21] whereby, at each iteration, a model is updated in such a way so as to achieve the greatest possible improvement of the fit. In other words, each iteration is locally optimal. Note that MARS, the Backfitting Algorithm [5] and various statistical subset selection procedures [39] also follow this type of model building strategy.

11.1 Adaptive Least Squares Procedure

Given a set of linearly independent basis functions $T_j(\mathbf{x})$, $j = 1, \ldots, J$ and a dataset $\mathcal{D} = \{(\mathbf{x}_n, y_n),\ n = 1, \ldots, N\}$, the Adaptive Least Squares procedure performs the least squares fit of the model $\hat{f}(\mathbf{x}) = \sum_{j=1}^{J} \beta_j T_j(\mathbf{x})$ to the dataset $\mathcal{D}$. Before formulating the ALS algorithm, we introduce several auxiliary notions. First, we define a set of vectors $\mathbf{f}_j$, $j = 1, \ldots, J$

$$\mathbf{f}_j = \beta_j \mathbf{t}_j,\ \beta_j \in R^1,\ \mathbf{t}_j = [T_j(\mathbf{x}_1), \ldots, T_j(\mathbf{x}_N)]^T,\ j = 1, \ldots, J. \tag{11.1}$$

Let $\mathbf{T}_j$, $j = 1, \ldots, J$ be spaces spanned by the vectors $\mathbf{t}_j$, $j = 1, \ldots, J$ and S_j, $j =$

$1, \ldots, J$ be $N \times N$ projection matrices corresponding to the respective spaces $\mathbf{T}_j$, $j = 1, \ldots, J$. The least squares fitting problem can be recast as follows: one has to find solution $\mathbf{f}_j \in \mathbf{T}_j$, $j = 1, \ldots, J$ of the following optimization problem

$$\underset{\mathbf{f}_1,\ldots,\mathbf{f}_J}{\text{minimize}} \ \text{RSS}(\mathbf{f}_1, \ldots, \mathbf{f}_J), \quad \text{where} \quad \text{RSS}(\mathbf{f}_1, \ldots, \mathbf{f}_J) = ||\mathbf{y} - \sum_{j=1}^{J} \mathbf{f}_j||^2 \tag{11.2}$$

It has a unique solution $\mathbf{f}_j^*$, $j = 1, \ldots, J$ and can be reformulated in terms of differences $\delta\mathbf{f}_j = \mathbf{f}_j - \mathbf{f}_j^*$

$$\underset{\delta\mathbf{f}_1,\ldots,\delta\mathbf{f}_J}{\text{minimize}} \ ||\mathbf{y} - \sum_{j=1}^{J} \mathbf{f}_j^* - \sum_{j=1}^{J} \delta\mathbf{f}_j||^2.$$

As the vector of residuals $\mathbf{y} - \sum_{j=1}^{J} \mathbf{f}_j^*$ is orthogonal to the vector $\sum_{j=1}^{J} \delta\mathbf{f}$, the above optimization problem can be cast as follows

$$\underset{\delta\mathbf{f}_1,\ldots,\delta\mathbf{f}_J}{\text{minimize}} \ ||\sum_{j=1}^{J} \delta\mathbf{f}_j||^2 + \text{RSS}(\mathbf{f}_1^*, \ldots, \mathbf{f}_J^*).$$

Thus, without any loss of generality, we can investigate the convergence properties of the ALS procedure assuming that $\mathbf{y} = 0$. Thus, from now on we will be concerned with the solution of the following problem

$$\underset{\mathbf{f}_1,\ldots,\mathbf{f}_J}{\text{minimize}} \ \text{RSS}(\mathbf{f}_1, \ldots, \mathbf{f}_J), \quad \text{where} \quad \text{RSS}(\mathbf{f}_1, \ldots, \mathbf{f}_J) = ||\sum_{j=1}^{J} \mathbf{f}_j||^2. \tag{11.3}$$

Note that it has a unique solution $\mathbf{f}_1 = 0, \ldots, \mathbf{f}_J = 0$ and the minimal value of the objective function in (11.3) is zero. The Adaptive Least Squares algorithm can be formulated as follows

Initialise: $\mathbf{f}_j = \mathbf{f}_j^0 \in \mathbf{T}_j$, $j = 1, \ldots, J$

Repeat: $k = 1, 2, \ldots$

Compute: $r_j,\ j = 1, \ldots, J$, where

$$r_j = \mathrm{RSS}(\mathbf{f}_1, \ldots, \mathbf{f}_j, \ldots, \mathbf{f}_J) - \mathrm{RSS}(\mathbf{f}_1, \ldots, \tilde{\mathbf{f}}_j, \ldots, \mathbf{f}_J), \quad \tilde{\mathbf{f}}_j \leftarrow -S_j(\sum_{l \neq j} \mathbf{f}_l).$$

Determine: $j_k = \underset{\mathrm{j}}{\mathrm{argmax}}\, r_j$.

Update:

$$\mathbf{f}_{j_k} \leftarrow -S_{j_k}(\sum_{l \neq j_k} \mathbf{f}_l),$$

Until: there is no significant change in the value of $\mathrm{RSS}(\cdot)$.

So, one iteration of ALS amounts to updating of one of the vectors $\mathbf{f}_j,\ j = 1, \ldots, J$. An update of the form $\mathbf{f}_j \leftarrow -S_j(\sum_{l \neq i} \mathbf{f}_j)$ of the vector $\mathbf{f}_j$ for some j results in a reduction in the value of the objective function $\mathrm{RSS}(\cdot)$ as the updated vector solve the following partial optimization problem

$$\underset{\mathbf{f_j}}{\text{minimize}}\ ||-(\sum_{l \neq j} \mathbf{f}_l) - \mathbf{f}_j||^2.$$

The algorithm computes the tentative updates $\tilde{\mathbf{f}}_j,\ j = 1, \ldots, J$ and evaluates the respective reductions $r_j,\ j = 1, \ldots, J$ in the value of the objective function. Based on these results, it determines the index j_k corresponding to the largest reduction and updates the vector $\mathbf{f}_{j_k}$. Such iterations are repeated until the value of the objective function $\mathrm{RSS}(\cdot)$ fails to decrease significantly.

As can be seen, the Adaptive Least Squares procedure is similar to both the forward subset selection [39] and the Backfitting Algorithm [5]. The well-known results concerning convergence properties of general optimization routines [35] can be utilized in the present investigation.

11.2 General Properties of ALS

In order to investigate properties of the Adaptive Least Squares procedure, it is convenient to define several new objects. First, we introduce a set Ω of vectors $\mathbf{h}$

$$\mathbf{h} = \begin{pmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_J \end{pmatrix}, \tag{11.4}$$

where $\mathbf{f}_1, \ldots, \mathbf{f}_J$ are defined in (11.1). Using this notation, the optimization problem (11.3) can be recast as follows

$$\underset{\mathbf{h}}{\text{minimize}} \ \text{RSS}(\mathbf{h}), \tag{11.5}$$

where $\text{RSS}(\mathbf{h}) = ||\sum_{j=1}^{J} \mathbf{f}_j||^2$, $\mathbf{f}_1, \ldots, \mathbf{f}_J$ being components of $\mathbf{h}$. Given a vector $\mathbf{h} \in \Omega$, we denote

$$\mathbf{h}^i = \begin{pmatrix} \mathbf{f}_1 \\ \vdots \\ -S_i(\sum_{j \neq i} \mathbf{f}_j) \\ \vdots \\ \mathbf{f}_J \end{pmatrix}. \tag{11.6}$$

Based on the above notation a point-to-set mapping $\mathbf{C} : \Omega \mapsto \Omega$ can be defined as follows

$$C(\mathbf{h}) = \{\mathbf{h}^{i_I}, \ i_I \in I \subset \{1, \ldots, J\}\}, \tag{11.7}$$

where the subset of indexes I is defined as

$$I = \underset{i\in\{1,\ldots,J\}}{\operatorname{argmin}} \operatorname{RSS}(\mathbf{h}^i).$$

Proposition 11.2.1 *The mapping C is closed*[1] *at any point* $\mathbf{h} \in \Omega$.

Proof. Suppose that $\mathbf{h}_k \to \hat{\mathbf{h}}$. Suppose also that $\mathbf{g}_k \in \mathbf{C}(\mathbf{h_k})$ and $\mathbf{g}_k \to \hat{\mathbf{g}}$. We have to show that $\hat{\mathbf{g}} \in C(\hat{\mathbf{h}})$. First we will prove that

$$\hat{\mathbf{g}} = \hat{\mathbf{h}}^{i^*}$$

for some i^*, where the definition of $\hat{\mathbf{h}}^{i^*}$ is given in (11.6). Indeed, for each $\mathbf{g}_k$ the following equality holds

$$\mathbf{g}_k = \mathbf{h}_k^{i_k}.$$

Therefore, we can select a subsequence $\{\mathbf{g}_{k_l} = \mathbf{h}_{k_l}^{i_{k_l}}\}_{l=1}^{\infty}$ such that $i_{k_1} = i_{k_2} = \ldots = i^*$. Such a subsequence always exits as there is only the finite number J of different values that can be taken on by the index i_k. Since $\mathbf{h}_{k_l} \to \hat{\mathbf{h}}$ and $\mathbf{g}_{k_l} \to \hat{\mathbf{g}}$, we conclude that

$$\hat{\mathbf{g}} = \mathbf{h}^{i^*}.$$

It also can be shown that $\hat{\mathbf{g}} \in C(\hat{\mathbf{h}})$. Assume that this is not the case which means that there exists i^{**} such that

$$\operatorname{RSS}(\hat{\mathbf{h}}^{i^{**}}) < \operatorname{RSS}(\hat{\mathbf{h}}^{i^*}).$$

Due to $\mathbf{h}_{k_l} \to \hat{\mathbf{h}}$, this is impossible because, otherwise, it would mean that $\operatorname{RSS}(\hat{\mathbf{h}}_{k_l}^{i^{**}}) < \operatorname{RSS}(\mathbf{h}_{k_l}^{i^*})$ for all l greater than some l_0 which contradicts to the fact that $\mathbf{h}_{k_l}^{i^*} \in C(\mathbf{h}_{k_l})$. Thus, $\hat{\mathbf{g}} \in C(\hat{\mathbf{h}})$ holds. □

It is possible to reformulate the Adaptive Least Squares procedure in terms of the above

[1] A point-to-set mapping $A: \Omega \to \Omega$ is said to be *closed* [35] at $\mathbf{h} \in \Omega$ if the assumptions $\mathbf{h}_k \to \hat{\mathbf{h}}$, $\mathbf{h}_k \in \Omega$ and $\mathbf{g}_k \to \hat{\mathbf{g}}$, $\mathbf{g}_k \in A(\mathbf{h}_k)$ imply that $\hat{\mathbf{g}} \in A(\hat{\mathbf{h}})$.

mapping: given an initial vector $\mathbf{h}_0 \in \Omega$, ALS generates a sequence $\{\mathbf{h}_k\}_{k=0}^{\infty}$ such that

$$\mathbf{h}_{k+1} \in C(\mathbf{h}_k). \tag{11.8}$$

The following Proposition establishes that the sequence $\{\mathbf{h}_k\}_{k=0}^{\infty}$ is bounded.

Proposition 11.2.2 *Let $\{\mathbf{h}_k\}_{k=0}^{\infty}$ be the sequence generated by the Adaptive Least Squares procedure. Then, there exists a constant $D > 0$ such that $||\mathbf{h}_k||^2 \leq D$ for all k.*

Proof. According to the nature of the algorithm, $\{\mathrm{RSS}(\mathbf{h}_k)\}_{k=1}^{\infty}$ decreases monotonically. Due to linear independency of the vectors $\mathbf{t}_j$, $j = 1, \ldots, J$ in (11.1), this implies that the euclidian norms of the vectors $\mathbf{h}_k$, $k = 0, 1, \ldots$ are bounded by some constant D. □

The properties of the ALS procedure stated in the Propositions 11.2.1 and 11.2.2 allow us to make use of the well-known results proved in [35] (the formulations are slightly adapted for the present context).

Proposition 11.2.3 *Let $\mathbf{C} : \Omega \mapsto \Omega$ be a point-to-set mapping, and suppose that, given $\mathbf{h}_0 \in \Omega$, the sequence $\{\mathbf{h}_k\}_{k=0}^{\infty}$ satisfying*

$$\mathbf{h}_{k+1} \in C(\mathbf{h}_k).$$

is generated. Let a set $\Gamma \subset \Omega$ be given, and suppose

1. *all points $\mathbf{h}_k$ are contained in a compact set $R \subset \Omega$*
2. *there is a continuous function Z on Ω such that*
   (a) *if $\mathbf{h} \notin \Gamma$, then $Z(\mathbf{g}) < Z(\mathbf{h})$ for all $\mathbf{g} \in C(\mathbf{h})$*
   (b) *if $\mathbf{h} \in \Gamma$, then $Z(\mathbf{g}) \leq Z(\mathbf{h})$ for all $\mathbf{g} \in \mathbf{C}(\mathbf{h})$*
3. *the mapping C is closed at points outside Γ.*

Then the limit $\hat{\mathbf{h}}$ of any convergent subsequence of $\{\mathbf{h}_k\}_{k=0}^{\infty}$ belongs to Γ.

Proposition 11.2.4 *If, under the conditions of the previous Proposition,* Γ *consists of a single point* $\hat{\mathbf{h}}$*, then the sequence* $\{\mathbf{h}_k\}_{k=0}^{\infty}$ *converges to* $\hat{\mathbf{h}}$.

It follows from the Propositions 11.2.1 and 11.2.2 that the mapping C defined in (11.7) as well as the sequence (11.8) satisfy the conditions of the Proposition 11.2.3 with the continuous function $Z(\mathbf{h})$ taken to be equal to $\mathrm{RSS}(\mathbf{h})$ and the set Γ comprised of a single zero vector. Thus, according to the Proposition 11.2.4, the sequence (11.8) generated by the Adaptive Least Squares procedure converges to the solution $\mathbf{h} = 0$ of the optimization problem (11.5).

11.3 Estimation of the Convergence Ratio

Now we would like to estimate the rate of convergence of the ALS procedure. We start with consideration of an auxiliary algorithm intended to minimnize the quadratic form $E(\mathbf{x}) = 1/2\mathbf{x}^T M\mathbf{x}$, where M is a positive definite $J \times J$ matrix and $\mathbf{x}$ is a J-dimensional vector

$$\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_J \end{pmatrix}. \tag{11.9}$$

Given a vector $\mathbf{x}$ let us consider one step of minimization of the form $E(\mathbf{x})$ according to the following algorithm:

1. compute the gradient of $E(\mathbf{x})$: $\mathbf{g} = M\mathbf{x} = (g_1 \cdots g_J)^T$.
2. pick the g_j having the largest absolute value, say g_{j^*}.
3. find α_0 such that

$$\alpha_0 = \arg[\min_{0 \le \alpha < \infty} E(\mathbf{x} - \tilde{\mathbf{g}}\alpha)],$$

where $\tilde{\mathbf{g}} = (0, \ldots, g_{j^*}, \ldots, 0)^T$.

4. $\tilde{\mathbf{x}} = \mathbf{x} - \tilde{\mathbf{g}}\alpha_0$.

Proposition 11.3.1 *Given* $\mathbf{x}$ *and* $\tilde{\mathbf{x}}$ *as defined in the above algorithm, the following estimate holds*

$$E(\tilde{\mathbf{x}}) \leq \left\{1 - \frac{1}{J}\frac{a}{A}\right\} E(\mathbf{x}),$$

where a *and* A *are smallest and largest eigenvalues respectively of the matrix* M.

Proof. First we compute the value of α_0 in the above procedure. We have

$$E(\mathbf{x} - \alpha\tilde{\mathbf{g}}) = (\mathbf{x} - \alpha\tilde{\mathbf{g}})^T M(\mathbf{x} - \alpha\tilde{\mathbf{g}}).$$

This function is minimized at

$$\alpha_0 = \frac{\tilde{\mathbf{g}}^T M\mathbf{x}}{\tilde{\mathbf{g}}^T M\tilde{\mathbf{g}}}.$$

Thus,

$$\begin{aligned}\frac{E(\mathbf{x}) - E(\tilde{\mathbf{x}})}{E(\mathbf{x})} &= \frac{2\alpha_0\tilde{\mathbf{g}}^T M\mathbf{x} - \alpha_0^2\tilde{\mathbf{g}}^T M\tilde{\mathbf{g}}}{\mathbf{x}^T M\mathbf{x}} \\ &= \frac{(\tilde{\mathbf{g}}^T M\mathbf{x})^2}{(\tilde{\mathbf{g}}^T M\tilde{\mathbf{g}})(\mathbf{x}^T M\mathbf{x})} \\ &= \frac{(\tilde{\mathbf{g}}^T\mathbf{g})^2}{(\tilde{\mathbf{g}}^T M\tilde{\mathbf{g}})(\mathbf{g}^T M^{-1}\mathbf{g})}.\end{aligned}$$

Also, the following inequality holds

$$g_{i^*}^2 \geq \frac{1}{J}(\mathbf{g}^T\mathbf{g})$$

as g_{i^*} has the largest absolute value among g_j, $j = 1, \ldots, J$. Therefore,

$$\frac{(\tilde{\mathbf{g}}^T\mathbf{g})^2}{(\tilde{\mathbf{g}}^T M\tilde{\mathbf{g}})(\mathbf{g}^T M^{-1}\mathbf{g})} \geq \frac{1}{J}\frac{(\tilde{\mathbf{g}}^T\tilde{\mathbf{g}})(\mathbf{g}^T\mathbf{g})}{(\tilde{\mathbf{g}}^T M\tilde{\mathbf{g}})(\mathbf{g}^T M^{-1}\mathbf{g})}.$$

Hence,

$$\frac{1}{J}\frac{(\tilde{\mathbf{g}}^T\tilde{\mathbf{g}})(\mathbf{g}^T\mathbf{g})}{(\tilde{\mathbf{g}}^T M\tilde{\mathbf{g}})(\mathbf{g}^T M^{-1}\mathbf{g})} \geq \frac{1}{J}\frac{a}{A}.$$

The statement of the Proposition follows from the above inequality. □

In order to apply this result to the ALS procedure, we make use of the fact that $\mathbf{f}_j$, $j = 1,\ldots,J$ belong to respective spaces $\mathbf{T}_j$, $j = 1,\ldots,J$ spanned by the system of linearly independent vectors $\mathbf{t}_j$, $j = 1,\ldots,J$ (see (11.1)). Thus, the optimization problem (11.3) can be recast in terms of the coefficients β_j, $j = 1,\ldots,J$ as

$$\underset{\beta_1,\ldots,\beta_J}{\text{minimize}}\ \mathrm{RSS}(\beta_1,\ldots,\beta_J), \quad \text{where} \quad \mathrm{RSS}(\beta_1,\ldots,\beta_J) = \beta^T T\beta,$$

where T is a $J\times J$ matrix with elements $T_{ij} = \mathbf{t}_i^T\mathbf{t}_j$, $i,j = 1,\ldots,J$, and $\beta = (\beta_1,\ldots,\beta_J)^T$. As can be seen, one step of the Adaptive Least Squares procedure is at least as optimal as one step of the algorithm described in this section. Therefore, considering the Proposition 11.3.1, convergence of the ALS procedure is at least linear

$$\mathrm{RSS}(\mathbf{h}_{k+1}) < \gamma\mathrm{RSS}(\mathbf{h}_k), \quad 0 < \gamma < 1,$$

and the upper bound for the convergence ratio γ is

$$\gamma < \left\{1 - \frac{1}{J}\frac{a}{A}\right\},$$

where a and A are smallest and largest eigenvalues respectively of the matrix T.

Chapter 12
Conclusion

In this thesis we are primarily concerned with two nonparametric regression analysis techniques: the Multivariate Regression Splines based on B-splines and Probing Least Absolute Squares Modelling. Below we recap on the main results obtained in this work and outline directions for future research.

12.1 Overview of the Main Results

The first part of the thesis is concerned with the Probing Least Absolute Squares Modelling (PLASM) technique which is a generalization of the LASSO approach proposed by R. Tibshirani [55]. The LASSO procedure (2.4) estimates regression coefficients of a linear model comprised of a set of basis functions via constrained minimization of the residual sum of squares. We introduced PLASM (2.7) which, although based on the ideas similar to those of LASSO, allows for arbitrary grouping of coefficients via utilization of a more general constraint. Below we summarize the most important properties of PLASM and emphasize the advantages of using it for solution of large scale regression estimation problems.

- PLASM retains some of the groups of basis functions in the model and removes the others by setting the respective coefficients to zero. Thus, PLASM performs model selection in terms of groups of basis functions rather than in terms of individual basis functions. One can say that PLASM is a product of LASSO and ridge regression

$$\text{PLASM} = \text{LASSO} \times \text{Ridge Regression},$$

in the sense that, the coefficients within each group are estimated in the way similar to that of the ridge regression whereas groups themselves are treated as if they were a sort of coefficients in the LASSO procedure. This approach to regression estimation is preferable in many situations such as, for example, high-dimensional additive modelling because the interpretability of a model is determined by the number of predictors involved in it rather than by the number of basis functions present in the model (see section 2.2 for more details).

- We developed an alternative formulation of PLASM (3.25) which is amenable to an efficient numerical solution. The important fact is that, unlike some of the non-parametric regression procedures mentioned in chapter 1, the computational cost associated with the numerical solution of PLASM is proportional to the number of data points which means that PLASM can be used efficiently in the Data Mining context.

- The PLASM approach with a modified constraint (5.12) turns out to be closely related to the so-called Smoothing Splines regression estimators (see section 5.4). As was pointed out in the introductory chapter 1, such estimators depend on a number of smoothing parameters. Determination of the optimal values for those parameters (based on, for instance, minimization of the Generalized Cross-Validation score) would involve solution of a (generally) nonconvex optimization problem with multiple local minima. The numerical solution of such problems is a difficult task. In contrast to this, the PLASM optimization problem (5.18) is convex and has a unique minimum point. Thus, PLASM with a modified constraint (5.12) can be viewed as an alternative way for estimation of smoothing parameters in the Smoothing Splines procedure.

- PLASM does not impose any limitations on the properties of the basis functions used to construct regression models and is able to deal with a variety of bases ranging from elementary piecewise constant functions to wavelets. PLASM appears to be suitable for dealing with categorical variables as well. The only modification required is to use appropriate indicator basis functions in the model.

- The ideas of PLASM are very fruitful and can be extended to tackle various other problems as will be discussed in the next section.

The second part of this thesis is dedicated to further modification of the Multivariate Adaptive Regression Splines algorithm. As J. Friedman pointed out in [20], his implementation of MARS was the first attempt at implementing this type of model building strategy. BMARS, although based on the same fundamental ideas, can be regarded as a further attempt to improve the performance of the procedure and the experimental results provided in this work demonstrate that this goal is accomplished at least to some extent. Although it is quite difficult to draw any definitive conclusions concerning the accuracy levels of models by MARS against models by BMARS, the computational cost of modelling with BMARS was considerably reduced through utilization of a new scale-by-scale approximation strategy as well as parallelization of the algorithm. This is of particular importance in the context of Data Mining.

12.2 Future Work

As was pointed out in section 12.1, the ideas of PLASM can be extended to deal with a variety of situations.

- Due to the often highly sophisticated structure of large high-dimensional data sets, additive models may not be adequate and, therefore, more complex models may have to be considered. This can be done as follows: regression coefficients corresponding to each univariate, bivariate etc component in the following model

$$f(\mathbf{x}) = \sum_{i=1}^{d} f_i(x_i) + \sum_{i,j=1,\ i>j}^{d} f_{ij}(x_i, x_j) + \dots \tag{12.1}$$

  are grouped and estimated via PLASM. Note that the dimensionality of the PLASM optimization problem (3.25) to be solved is equal to the number of terms in the model above rather than to the total number of basis functions involved (it can be large if interactions between variables are considered).

- One more application of PLASM is to tree-based modelling which is another important Data Mining tool. Tibshirani and LeBlanc investigated this opportunity for LASSO [32]. However, PLASM seems to be more suitable in such context as it allows one to group parameters describing a branch of a tree. Thus, PLASM appears be

able to perform tree pruning without resorting to artificial monotonicity constraints used in LASSO-based tree pruning.

- As was pointed out by P. Hall and his co-workers in the paper [27], block-thresholding can considerably improve the quality of estimators based on wavelets. This result is stated only for orthogonal wavelet bases. On the other hand, according to the results of section (4.1) dedicated to the orthogonal design case, PLASM turns out to block-threshold regression coefficients as well. Therefore, it can be regarded as an extension of the block-thresholding approach to a more general case of arbitrary (nonorthogonal) basis functions. Thus, a possible direction for future research is to look into properties of PLASM regression estimators based on general wavelet dictionaries.

- As was mentioned in the first item of this list, PLASM can be used to estimate regression models comprised of univariate as well as bivariate terms. In order to model bivariate terms, the approach based on, for instance, finite element thin plate splines [29] can be used. However, it can be shown that, in order to ensure the uniqueness of a solution, one has to consider the PLASM formulation (2.7) coupled with a number of linear equality constraints. The results of a preliminary investigation indicate that a considerable modification of the approach used in this thesis is required to deal with the new optimization problem.

- The formulation of the PLASM optimization problem (3.25) contains a free parameter $t' \in (0, \infty)$ and the optimal value for this parameter has to be determined based on the minimization of some estimate of the Prediction Error (1.4). This means that a complete characterization of PLASM solutions for $t' \in (0, \infty)$ would have to be obtained in order to carry out the minimization efficiently. This could be done via an adaptation of a homotopy approach which was used successfully to characterize solutions of LASSO [44] and in some other situations [42].

- The BMARS algorithm is good at selecting the most important variables and their interactions. Moreover, parallel BMARS is able to carry out this task quite quickly. However, the tensor product basis functions used in the procedure fail to provide an adequate fit in some situations (such as approximation of highly structured interaction terms). To rectify this deficiency, we propose the following two-stage algorithm for solution of large scale regression estimation problems. First, in order to select the most influential variables and their interactions, one runs BMARS and, based on

the results, forms a model of the type (12.1) involving only the important univariate and bivariate terms. Second, PLASM is applied to obtain higher quality fit of these terms using, for example, finite elements thin plate splines [29].

Appendix A

A Short User's Guide to BMARS

BMARS is a software package implementing the parallel Multivariate Adaptive Regression Splines algorithm based on B-splines (see chapter 8). The parallelization of BMARS is achieved via utilization of the Parallel Virtual Machine (PVM) software package that is able to hook together a heterogeneous collection of computers by a network so that they can be used as a single large parallel computer with distributed memory. PVM is available for a variety of platforms including a number of multiprocessor systems. A program based on PVM is able to run on any of the above platforms without any alterations. The PVM software allows one to solve Data Mining problems more efficiently via

- distribution of the computational load among several processors,
- utilization of the Random Access Memory of a number of systems.

This guide does not discuss issues related to the installation and/or configuration of the PVM package as the relevant information can be found on the PVM Web site

`http://www.netlib.org/pvm3`.

The BMARS package is comprised of two modules

- `BM_main.out` is the *master* module of the algorithm (see section 8.4). There is only one process by this name running on the Parallel Virtual Machine at any time.
- `slave1.0.out` is the *slave* module of the algorithm (see section 8.4). The master module spawns a number (specified by a user) of the slave processes. Depending on the type of the hardware comprising the Virtual Machine, the slave processes can run either on the processors of a multiprocessor system or on the remote computers.

The BMARS software does not require recompilation as the parameters of the problem in hand change. There is a configuration file called `bmarsconfig` containing parameters relevant to a problem and which is read by the `BM_main.out` module. Thus, the only thing one has to do in order to switch from problem to problem is to edit the configuration file. It is an ASCII file where each line contains a value (a number or a string) for one parameter. Below is the list of the acceptable parameters in the order as they should appear in the configuration file:

1. number of data records in a data set (integer number)
2. number of numeric predictors (integer number)
3. number of categorical predictors (integer number)
4. maximum order of interaction between predictor variables in a model (integer number)
5. number of tensor product basis function to be produced by the forward stepwise procedure of BMARS (integer number)
6. maximum number of distinct values for any categorical variable (integer number)
7. predictor variable flags (a sequence of integer numbers separated by spaces): 1 – categorical variable, 0 – numeric variable
8. logistic regression flag (integer number): 1 – logistic regression is to be performed, 0 – logistic regression is not to be performed
9. offset flag (integer number): 1 – offset values are to be used when performing logistic regression, 0 – offset values are not to be used
10. weighted regression flag (integer number): 1 – weights are to be used when building a BMARS model, 0 – weights are not to be used
11. number of raster points for depicting curves and surfaces (first variable) (integer number)
12. number of raster points for depicting surfaces (second variable) (integer number)
13. graphical output flag (integer number): 1 – MATLAB script files are to be produced, 0 – MATLAB script files are not to be produced

14. model flag (integer number): 1 – piecewise cubic model, 0 – piecewise linear model
15. smoothing parameter used in the forward stepwise procedure of BMARS (real number, F15.10 format)
16. smoothing parameter used in the backward elimination procedure of BMARS (real number, F15.10 format)
17. name of the directory in which the MATLAB script files are to be placed (string)
18. name of a binary file containing specifications of a produced BMARS model (string)
19. name of a data set (string)
20. number of the slave modules to be spawned (integer number)
21. list of names of hosts comprising the virtual machine (one name per line), duplicate names are allowed (strings)

The data should be in the form of an ASCII file containing one data record per line. The format of a data record is as follows: values for predictor variables should go first followed by a value for a response variable. The next (and the last) position of a data record can be empty or, alternatively it can contain a value for either a weight variable or an offset variable (depending on the nature of the problem in hand).

The BMARS executables compiled on a SUN workstation are available from the author on request.

Bibliography

[1] P. Bloomfield and W.L. Steiger. *Least Absolute Deviations: Theory, Applications, and Algorithms.* Birkhauser: Boston, 1983.

[2] K. Borch. *Economics of insurance.* Elsevier Science, 1990.

[3] R. Brachman, T. Khabaza, W. Kloesgen, G. Piatetsky-Shapiro, and E. Dimoudis. Industrial applications of data mining and knowledge discovery. *Communications of ACM*, 39(11), 1996.

[4] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees.* Wadsworth, Belmont, 1984.

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

[6] G. Casella and E.I. George. Explaining the gibbs sampler. *The American Statistician*, 46:167–174, 1992.

[7] Z. Chen. Beyond additive models: interactions by smoothing spline methods. Technical Report SMS-009-90, The Australian National University, 1990.

[8] H. Chipman. Bayesian variable selection with related predictors. *The Canadian Journal of Statistics*, 24(1):17–36, 1996.

[9] W. S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74:829–836, 1979.

[10] R. Courant and D. Hilbert. *Methods of Mathematical Physics.* N.Y., Interscience, 1953.

[11] D.R. Cox. *Analysis of Binary Data.* Chapman and Hall, 1970.

[12] M.G. Cox. Practical spline approximation. In *Lecture Notes in Mathematics, 965*, pages 79–112. Springer, Berlin-New York, 1982.

[13] P. Craven and G. Wahba. Smoothing noisy data with spline functions: Estimating the correct degree of smoothing by the method of generalized cross-validation. *Numerische Mathematik*, 31:317–403, 1979.

[14] C. De Boor. *A practical guide to splines.* Springer-Verlag, 1978.

[15] A. Dobson. *An introduction to generalized linear models.* Chapman and Hall, 1990.

[16] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap.* Chapman and Hall, 1993.

[17] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery: An overview. In *Advances in Knowledge Discovery and Data Mining*, pages 1–36. MIT Press, Cambridge, MA, 1996.

[18] R. Fletcher. *Practical Methods of Optimization, 2nd Ed.* John Wiley & Sons, 1987.

[19] J. Friedman. Estimating functions of mixed ordinal and categorical variables. Technical Report 108, Stanford University, 1991.

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

[21] J.H. Friedman. Greedy function approximation: A gradient boosting machine. *http://stat.stanford.edu/ jhf/ftp/trebst.ps*, 1999.

[22] J.H. Friedman and W. Stuetzle. Projection pursuit regression. *Journal of American Statistical Association*, 76:817–823, 1981.

[23] E.I. George and R.E. McCulloch. Variable selection via gibbs sampling. *Journal of the American Statistical Association*, 88:881–889, 1993.

[24] F. Girosi. An equivalence between sparse approximation and support vector machines. Technical Report AI Memo No. 1606, CBCL No. 147, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1997.

[25] G.H. Golub and C.F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, Baltimore, 1983.

[26] P.G. Green and B.W. Silverman. *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach.* Chapman and Hall, 1994.

[27] P. Hall, S. Penev, G. Kerkyacharian, and D. Picard. Numerical performance of block thresholded wavelet estimators. *Statistics and Computing*, 7:115–124, 1997.

[28] T. J. Hastie and R. J. Tibshirani. *Generalized additive models.* Chapman and Hall, 1990.

[29] M. Hegland, S. Roberts, and I. Altas. Finite element thin plate splines for data mining applications. In *Mathematical Methods for Curves Surfaces II*, 1997.

[30] A.E. Hoerl and R.W. Kennard. Ridge regression: Applications to nonorthogonal problems. *Technometrics*, 12:69–82, 1970.

[31] A.E. Hoerl and R.W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

[32] M. LeBlanc and R. Tibshirani. Monotone shrinkage of trees. Technical Report 9420, Department of Statistics, University of Toronto, 1994.

[33] H. Linhart and W. Zucchini. *Model selection.* New York: Wiley, 1986.

[34] M. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of seond order cone programming. *Linear Algebra and Applications*, 284:193–228, 1998.

[35] D.G. Luenberger. *Linear and Nonlinear Programming.* Reading, Massachusetts, 1984.

[36] Zhen Luo. Backfitting in smoothing spline anova with application to historical global temperature data. Technical Report 964, Department of Statistics, University of Wisconsin-Madison, 1996.

[37] P. McCullagh and J.A. Nelder. *Generalized Linear Models.* Chapman and Hall, 1983.

[38] I. McIntosh. Development and analysis of a scalable data mining technique based on generalized additive models. Master's thesis, The Department of Computer Science, The Australian National University, 1997.

[39] A.J. Miller. *Subset Selection in Regression.* Chapman and Hall, 1990.

[40] R.G. Miller. The jackknife - a review. *Biometrika*, 61:1–15, 1974.

[41] Yu. Nesterov and A. Nemerovsky. *Interior-point polynomial methods in convex programming.* SIAM, Philadelphia, PA, 1994.

[42] M. R. Osborne. An effective method for computing regression quantiles. *IMA Journal of Numerical Analysis*, 12:151–166, 1992.

[43] M. R. Osborne. Variable selection and control in least squares problems. Technical Report MRR 047-98, Centre for Mathematics and its Applications, The Australian National University, 1998.

[44] M. R. Osborne, B. Presnell, and B. A. Turlach. Berwin's problem. Technical Report MRR 049-98, Centre for Mathematics and its Applications, The Australian National University, 1998.

[45] M.R. Osborne. On the computation of stepwise regressions. *The Australian Computer Journal*, 8(2):61–67, 1976.

[46] M.R. Osborne. Gram-schmidt for least squares regression problems : A sweep based algorithm for orthogonal factorization. Technical Report SMS-015-90, The Australian National University, 1990.

[47] E. Osuna, R. Freund, and F. Girosi. Improved training algorithm for support vector machines. In *1997 IEEE Workshop on Neural Networks for Signal Processing*, 1997.

[48] M. J. D. Powell. A fast algorithm for nonlinearly constrained optimization calculations. In *Lecture Notes in Mathematics, 630*, pages 144–157. Springer, Berlin, 1978.

[49] L. Schumaker. Discussion: Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1):112–113, 1991.

[50] M. Smith and R. Kohn. Nonparametric regression using bayesian variable selection. *Journal of Econometrics*, 75:317–344, 1996.

[51] C.J. Stone. Additive regression and other nonparametric models. *Annals of Statistics*, 13(2):689–705, 1985.

[52] C.J. Stone. The dimensionality reduction principle for generalized additive models. *Annals of Statistics*, 14(2):590–606, 1986.

[53] G. Stone. Analysis of motor vehicle claims data using statistical data mining. Technical Report CMIS-97/73, CMIS, CSIRO, 1997.

[54] M. Stone. Cross-validation: a review. *Math. Operationsforsch, Ser. Statist.*, 9:127–139, 1978.

[55] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society B*, 58:267–288, 1996.

[56] J. W. Tukey. *Exploratory Data Analysis.* Reading, Massachusetts, 1977.

[57] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, 1995.

[58] V. Vapnik, S. Golowich, and A. Smola. Support vector method for function approximation, regression estimation, and signal processing. In M. Mozer, M. Jordan, and T. Petsche, editors, *Neural Information Processing Systems.* MIT Press, Cambridge, MA, 1997.

[59] G. Wahba. *Spline Models for Observational Data.* SIAM, Philadelphia, 1990.