Regression Spline via Penalizing Derivatives

ZHU YEYING

(B.Sc. East China Normal Univ.)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF STATISTICS AND APPLIED PROBABILITY

NATIONAL UNIVERSITY OF SINGAPORE

2008

Acknowledgements

First of all, I would like to express my gratitude to my supervisor, Assoc. Professor Zhang Jin-Ting for his guidance and encouragement during my two-year graduate research and study. His ideas and expertise are crucial to the completion of this thesis. I would like to thank him for spending his valuable time revising this thesis. I would also like to thank him for teaching me how to undertake researches and write academic articles step by step.

Then, I would like to give my gratitude to my friends, especially to Li Ziwu for his careful revision of the language of this thesis and to Wang Daqing and Gaoyan for sharing their thoughts with me.

Finally, I want to take this opportunity to thank my parents, who always love me and support me.

Contents

Summary

Regression spline based on a truncated power basis $\Psi(t)$ has been proved to be a very useful nonparametric method for fitting a data set generated from the non-parametric regression model $y_i = m(t_i) + \epsilon_i, i = 1, 2, \ldots, n$, where the underlying function $m(t)$ is unknown. One way to implement this method is to approximate $m(t)$ as $\mathbf{\Psi}(t)^T\beta$ and estimate the coefficient vector β appropriately as done in the literature. In situations when β is large dimensional and sparse, the smoothly clipped absolute deviation (SCAD) method of Fan and Li (2001) can be adopted to select and estimate the non-zero components of β simultaneously. In some other cases, the coefficients in β may not be sparse, but the pth times derivatives of the regression function are sparse. If so, directly applying the SCAD method is less effective. In this thesis, we attempt to re-parameterize the coefficient vector β as a linear function of certain derivative vector γ, whose last $K + 1$ components are the pth times derivatives of the regression spline function. That is, we have $\beta = \mathbf{A}^{-1}\gamma$ where A is a known link matrix which we can derive from the basis functions. Then, we can express $m(t)$ as $\mathbf{\Psi}(t)^T A^{-1}\gamma$ which is an approximation

function of γ. We then apply the SCAD method of Fan and Li (2001) to estimate the coefficient vector γ. Simultaneously, β can be estimated through $\beta = A^{-1}\gamma$. Numerical results show that the newly proposed method is much more accurate than the usual regression spline methods in the literature, especially when the true curve is piecewise with different orders of polynomials at different segments.

Keywords: Regression Splines, SCAD, Derivatives, Sparse

List of Tables

List of Figures

Chapter 1

Introduction

1.1 The Problem

Suppose we have a noisy data set (t_1, y_1), (t_2, y_2),...,(t_n, y_n), generated from the following standard nonparametric regression model:

$$y_i = m(t_i) + \epsilon_i, i = 1, 2, \ldots, n, \tag{1.1}$$

where $m(\cdot)$ denotes the unknown underlying function and ϵ_i denotes the ith measurement error. Usually, we assume $E(\epsilon_i) = 0$ and $V(\epsilon_i) = \sigma^2$, for $i = 1, 2, \ldots, n$.

Methods and theories for estimating the underlying function $m(\cdot)$ in the model (1.1) have been well established, mainly including the kernel methods (Nadaraya 1964, Watson 1964, Gasser and Mueller 1984), local polynomial kernel methods (Fan and Gijbels 1996), smoothing splines (Wahba 1990, Green and Silverman 1994), regression splines (Eubank 1988) and penalized splines (Ruppert and Carroll

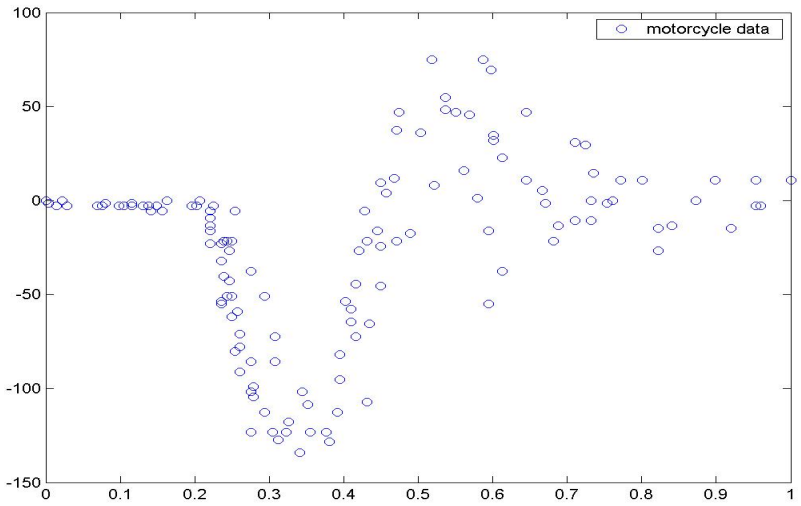

Figure 1.1: *The motorcycle data.*

1997). All these methods work well when we assume that the underlying function $m(\cdot)$ is smooth, i.e., having derivatives of up to some order p where p is a positive integer.

In many cases, the underlying function $m(\cdot)$ may be a piecewise polynomial function but with different polynomial orders at different intervals of the support of the design time points. Figure 1.1 presents a real data example with such a feature. This is a classical data set with $n = 133$ observations, first analyzed by Silverman (1985). The dependent variable is the time after a stimulated impact with motorcycles; for simplicity, throughout this thesis, the design times for the motorcycle data have been scaled to $[0, 1]$. The response variable is the head acceleration of a PTMO (post mortem human test object), which captures the crash effects.

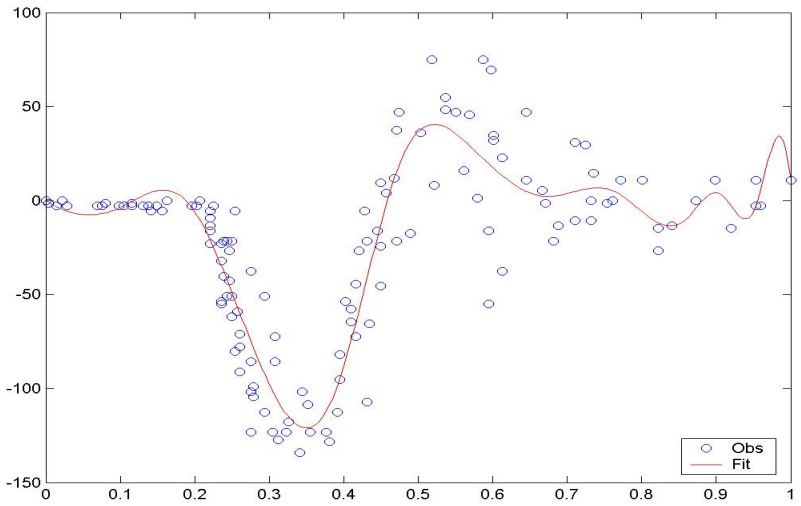

Figure 1.2: *Usual regression spline fit to the motorcycle data.*

It is seen that the underlying function may be a piecewise polynomial but with different orders of the polynomials at different intervals of the support. In the interval $[0, 0.2]$, the underlying function must be a constant function; in the interval $(0.2, 0.5]$, the underlying function may be a quadratic function; while in the interval $(0.5, 1]$, the underlying function seems to be another quadratic function. To fit such a data set, the major smoothing methods mentioned previously do not take the advantage of the above-mentioned information directly. As a result, the resulting fit may not fit the data well.

As an example, we present a usual regression fit to the motorcycle data in Figure 1.2. The fit was obtained using the cubic truncated power basis with 20 initial knots scattered uniformly in the support of the design time points, but with the regression spline coefficients estimated by applying the SCAD method (Fan and

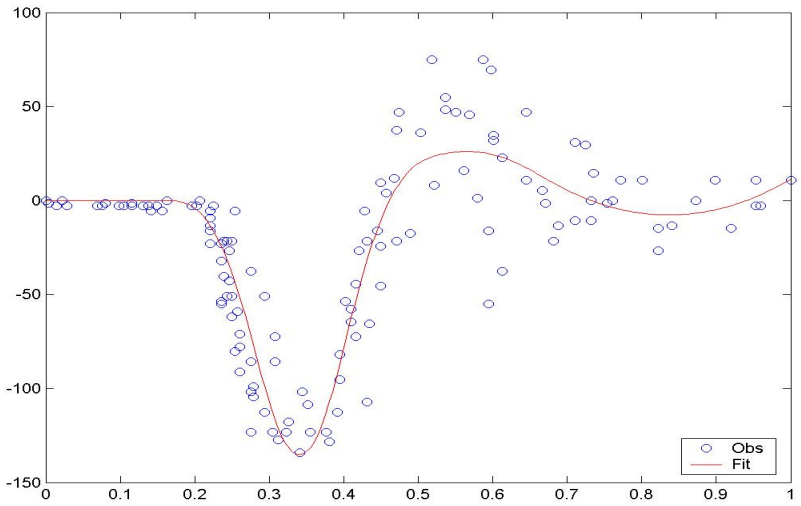

Figure 1.3: *New regression spline fit to the motorcycle data.*

Li 2001) directly to the regression spline coefficients. The details of the method on how to obtain such a fit will be reviewed in Chapter 2. It is seen that overall, the fit looks well to the motorcycle data except in the interval $[0, 0.2]$ where the underlying function looks flat but the usual regression spline fit with the SCAD estimator did not use the information that the third times derivatives of the fitted function would be zero for most design points, and hence can not fit the data well in this interval. Other popular smoothing methods mentioned previously also have such a problem.

This problem can be overcome by the new regression spline fit proposed in this thesis. Figure 1.3 presents the new regression spline fit to the motorcycle data. Similarly, we used the cubic truncated power basis with 20 initial knots scattered uniformly in the support of the design time points, but with the regression coef-

ficients estimated by applying the SCAD method (Fan and Li 2001) to the third times derivative of the regression spline function. We call the new method as regression spline smoothing via penalizing derivatives. The details of the method will be presented in Chapter 3. It is seen that the new regression spline fit outperforms the usual regression spline fit presented in Figure 1.2, especially over the interval $[0, 0.2]$.

1.2 Main Results of the Thesis

In this thesis, we focus on the regression spline model with a pth order truncated power basis $\mathbf{\Psi}(t)$. In many cases, the underlying function may be a piecewise polynomial but with different orders of the polynomials at different intervals of the support. Given p is large enough to capture all the different patterns, the pth times derivatives of the regression spline function could be sparse. This is because the pth times derivatives of the function are zero for the intervals with polynomial orders less than p. Therefore, we attempt to transform the original regression spline coefficient vector β into a new vector γ, whose last $K + 1$ components are the pth times derivatives of the regression spline function within different intervals. We then apply the SCAD method to estimate γ, and β can be obtained by $\mathbf{A}^{-1}\gamma$, where $\mathbf{A}$ is a link matrix. We call the newly proposed smoothing method as "regression spline smoothing via penalizing derivatives". The study of the asymptotic properties of the newly proposed estimator shows that the estimator converges to

the true value with probability tending to 1. Simulations and real data applications demonstrate the good performance of the newly proposed method and also shows that the proposed method outperforms the traditional regression spline methods.

1.3 Organization of the Thesis

The layout of this thesis is as follows. In Chapter 2, we first review the usual regression spline smoothing method. Section 2.2.1 gives the definition of regression splines, followed by a brief description of two widely used spline bases in Section 2.2.2. Section 2.2.3 focuses on the regression spline modeling with truncated power basis. Sections 2.2.4 & 2.2.5 discuss issues on locating knots properly and choosing the number of knots smartly. Section 2.2.6 & 2.2.7 present methods for estimating the regression spline coefficients, including the ordinary least square (OLS) method, the best subset method and the penalized least squares method. Finally, in Section 2.3, we review the SCAD method of Fan and Li (2001) for variable selection for linear models.

The major work of this thesis will be presented in Chapter 3. Section 3.1 introduces the key idea of the new regression spline smoothing method, i.e., regression spline smoothing via penalizing derivatives. We propose a method to re-parameterize the original regression spline model in terms of the p-th order derivatives of the regression spline function. Section 3.2 provides the details of our method and discusses issues for selecting the tuning parameters: the knot locating

method, the number of knots and the choice of the truncated power basis order, for the new regression spline method. Asymptotic properties of the newly proposed estimator are also studied in this section. Simulation studies and real data examples will be presented in Section 3.3 & 3.4, respectively. Finally, some discussions are given in Section 3.5.

Chapter 2

Usual Regression Spline Smoothing

2.1 Introduction

In recent years, nonparametric smoothing methods have received extensive attention for its robustness in estimation and prediction based on the standard nonparametric regression model (1.1), i.e.,

$$y_i = m(t_i) + \epsilon_i, \quad i = 1, 2, \ldots, n, \tag{2.1}$$

when a parametric regression model is not available or not easy to be found for a given noisy data set $(t_1, y_1), (t_2, y_2), \ldots, (t_n, y_n)$. A simple example is the motorcycle data set presented in Figure 1.1. For such a data set, it is not easy to find a proper parametric model. That is why we want to fit it using some nonparametric techniques. As mentioned in Chapter 1, popular nonparametric techniques include the kernel methods (Nadaraya 1964, Watson 1964, Gasser and Muller 1990), local

polynomial kernel smoothing method (Fan and Gijbels 1996), smoothing splines (Wahba 1990, Green and Silverman 1994), regression splines (Eubank 1988) and penalized splines (Ruppert and Carroll 1997) among others.

It is well known that a nonparametric estimator of $m(t)$ obtained using any of the above popular smoothing techniques is a linear smoother of the response vector $\mathbf{y}$. That is, the smoother only employs the information obtained from the noisy sample $(t_1, y_1), (t_2, y_2), \ldots, (t_n, y_n)$, by averaging responses into a smooth function. Compared to a traditional parametric regression function, the nonparametric regression function is more flexible, which might be piecewise or wiggly. Among various nonparametric techniques, regression splines are widely used to estimate $m(t)$ due to its simplicity and natural generalizations from polynomials. In this chapter, we shall make a brief review on usual regression spline smoothing. This is because our new method proposed for estimating $m(\cdot)$ of (1.1) is essentially a natural generalization of the usual regression spline technique. The revision will be done in Section 2.2 below. We will also review the SCAD method of Fan and Li (2001) for variable selection for linear models since we also need this technique for the proposed new method.

2.2 Usual Regression Splines

2.2.1 Definition of a Regression Spline

A regression spline is a piecewise polynomial function connected at some pre-specified locations. These locations are called knots. A group of K knots can be denoted as

$$\tau_1, \tau_2, \ldots, \tau_K, \tag{2.2}$$

which are distinct points and in an increasing order. These knots divide the support of the regression function $m(t)$ (without loss of generality, we assume the support of t is a compact interval $[0, 1]$) into $K + 1$ subintervals. These knots are called inner knots, while in some literatures, $\tau_0 = 0$ and $\tau_{K+1} = 1$ are called boundary knots. It is well known that regression splines are based on the construction of regression spline bases. For a given regression spline basis vector $\mathbf{\Psi}(t) : q \times 1$, we can always express a regression spline function as $\mathbf{\Psi}(t)^T\beta$ where $\beta : q \times 1$ is called the regression spline coefficient vector. Two regression spline bases will be introduced in next subsection.

2.2.2 Regression Spline Bases

There are various kinds of regression spline bases in the literature. Here we introduce two of them, which are most commonly used, including the truncated power basis and the B-spline basis.

Truncated Power Basis (TPB): For a given knot sequence (2.2), a pth order truncated power basis is defined as

$$1, t, t^2, \ldots, t^p, (t-\tau_1)_+^p, \ldots, (t-\tau_K)_+^p, \tag{2.3}$$

where $u_+^p = (u_+)^p$ is a truncated power function and $u_+ = \max(0, u)$, denoting a function truncated at $u = 0$.

Based on a given truncated power basis (2.3), a regression spline can be expressed as

$$f(t) = \beta_0 + \beta_1 t + \cdots + \beta_p t^p + \beta_{p+1}(t-\tau_1)_+^p + \cdots + \beta_{p+K}(t-\tau_K)_+^p. \tag{2.4}$$

When $p = 1, 2, 3$, the associated regression splines are known as linear, quadratic and cubic regression splines.

It can be seen from (2.4) that, the first $p+1$ terms of the regression spline function are the polynomial functions up to pth order. Therefore, a pth order regression spline is a natural generalization of a pth order polynomial. This can be further illustrated by observing $f(t)$ within any two neighboring knots. For example, with the interval (τ_k, τ_{k+1}), the regression spline $f(t)$ as defined in (2.4) can be re-expressed as

$$f(t) = \beta_0 + \beta_1 t + \cdots + \beta_p t^p + \beta_{p+1}(t-\tau_1)^p + \cdots + \beta_{p+k}(t-\tau_k)^p,$$

which is obviously a pth order polynomial, and has any times derivatives within the interval. However, at each knot, there exists only up to $(p-1)$-times continuous

derivatives. Notice that the pth times derivative of the regression spline $f(t)$ is

$$f^{(p)}(t) = p!(\beta_p + \beta_{p+1} + \cdots + \beta_{p+k}), \quad t \in (\tau_k, \tau_{k+1}). \tag{2.5}$$

It follows that

$$\beta_{p+k} = \{f^{(p)}(\tau_k+) - f^{(p)}(\tau_k-)\}/p!. \tag{2.6}$$

That is, the regression spline coefficient β_{p+k} is proportional to the jump of $f^{(p)}(t)$ at the kth knot τ_k. When the jump is 0, the associated regression spline coefficient β_{p+k} is 0 and should be removed from the expression of $f(t)$. It is equivalent to removing the k-th knot from the knot sequence (2.2).

B-spline Basis: When K is too large, the TPB may be close to ill-conditioned, i.e., the associated design matrix may be near degenerated. A B-spline basis is introduced to overcome this problem. For a given knot sequence (2.2), a pth order B-spline basis may be defined as

$$N_{0,p}(t), N_{1,p}(t), \cdots, N_{K-p,p}(t),$$

where $N_{i,j}(t)$ is calculated in a recursive way as follows:

$$N_{i,j}(t) = \frac{t - \tau_i}{\tau_{i+j} - \tau_i} N_{i,j-1}(t) + \frac{\tau_{i+j+1} - t}{\tau_{i+j+1} - \tau_{i+1}} N_{i+1,j-1}(t), \qquad j = 1, \ldots, p,$$

where the boundary knots $\tau_0 = 0$ and $\tau_{K+1} = 1$ are used. When $p = 0$, the associated B-spline are

$$N_{i,0}(t) = I_{[\tau_i, \tau_{i+1})}, \ i = 0, 1, \cdots, K$$

where I is the indicator function. When $p = 1, 2, 3$, the associated B-splines are called linear, quadratic and cubic B-splines respectively, which are widely used.

In the literature, lower order TPB and B-spline bases (especially the quadratic and cubic spline bases) are widely used to fit the standard nonparametric regression model (2.1). TPB is easy to construct and computationally fast. However, as mentioned previously, the design matrix based on a TPB may be ill-conditioned when the number of knots K is too large. For a B-spline basis, the associated design matrix is sparse as $N_{i,j}(t) = 0$ when t is not within the interval $[\tau_i, \tau_{i+j+1})$. So the design matrix is sparse and well conditioned. The major drawback of the B-spline basis is its computational burden. It is far more complicated, comparing to TPB. Wand (2000) pointed out that the regression spline fit of the unknown function $m(t)$ of (2.1) should not be very sensitive to the choice of spline bases. Thus, in this thesis, we only use the TPB due to its simplicity.

2.2.3 Regression Spline Modeling

As mentioned in the last subsection, this thesis focuses on the TPB model only. For a given TPB vector:

$$\mathbf{\Psi}(t) = (1, t, t^2, \cdots, t^p, (t-\tau_1)_+^p, \cdots, (t-\tau_K)_+^p)^T, \tag{2.7}$$

the standard nonparametric regression model (2.1) can be approximated by the following regression spline model:

$$y_i = \mathbf{\Psi}(t_i)^T \beta + \epsilon_i, \quad i = 1, 2, \cdots, n, \tag{2.8}$$

where $\beta = (\beta_0, \beta_1, \ldots, \beta_{p+K})^T$ is the associated regression spline coefficient vector. Denote the response vector by $\mathbf{y} = (y_1, y_2, \ldots, y_n)^T$ and the design matrix by

$\mathbf{X} = (\mathbf{\Psi}(t_1), \cdots, \mathbf{\Psi}(t_n))^T$, the above model can be written in the following matrix form

$$\mathbf{y} = \mathbf{X}\beta + \epsilon, \tag{2.9}$$

where $\epsilon = (\epsilon_1, \cdots, \epsilon_n)^T$ is the noise vector. The design matrix $\mathbf{X}$ can be written in a more clear way as

$$\begin{pmatrix} 1 & t_1 & t_1^2 & \dots & t_1^p & (t_1-\tau_1)_+^p & \dots & (t_1-\tau_K)_+^p \\ 1 & t_2 & t_2^2 & \dots & t_2^p & (t_2-\tau_1)_+^p & \dots & (t_2-\tau_K)_+^p \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & t_n & t_n^2 & \dots & t_n^p & (t_n-\tau_1)_+^p & \dots & (t_n-\tau_K)_+^p \end{pmatrix}.$$

When the number of knots K is well chosen, and the knot sequence (2.2) is well specified, the regression spline model (2.8) can be fitted via the ordinary least squares (OLS) estimator. That is, the regression spline coefficient vector β can be estimated as

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}. \tag{2.10}$$

Therefore, the least squares fit of $m(t)$ is

$$\hat{m}(t) = \mathbf{\Psi}(t)^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}, \tag{2.11}$$

and the least squares fit of the response vector $\mathbf{y}$ is

$$\hat{\mathbf{y}} = \mathbf{A}\mathbf{y} \tag{2.12}$$

where

$$\mathbf{A} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T, \tag{2.13}$$

is the so-called smoother matrix. The advantage of the regression spline smoother (2.11) is that it provides estimates of $m(t)$ at any given time point t. It can be seen that the smoother differs when the location of the knots and the knot number change. Hence, the quality of $\hat{m}(t)$ depends on the location of the knots (2.2) and the choice of the knot number K. In the next two subsections, we shall discuss how to locate the knots and how to select the number of knots.

2.2.4 Locating the Knots

To achieve a good regression spline estimator $\hat{m}(t)$, we first need to locate the knots. There are two simple methods for locating the knots for a regression spline basis. One way for doing that is to scatter the knots uniformly in the support, say, $[0, 1]$. That is, for a given number of knots, K, the knots are specified as

$$\tau_j = j/(K+1), \quad j = 1, 2, \ldots, K. \tag{2.14}$$

This way is usually referred to as the uniform knot locating rule. The advantage of this method is that the knot locating does not depend on the distribution of the design time points.

The other way is to specify the knots as the equally spaced sample quantiles of the design time points $t_1, t_2, \ldots, t_n$. That is, the knots are specified as

$$\tau_j = t_{([100j/(K+1)])}, \quad j = 1, 2, \ldots, K, \tag{2.15}$$

where $[x]$ denotes the integer part of x and $t_{(i)}$ is the ith quantile of $t_1, t_2, \ldots, t_n$. This method is usually referred to the quantile as knots rule. The advantage of

this method is that it specifies more knots where the design time points are more dense. It performs similarly to the first method when the design time points are uniformly scattered in $[0, 1]$.

Besides the two simple methods mentioned above, other methods are also useful. For example, one may locate the knots based on some empirical perspective, e.g., to place the knots at the points, where we believe dramatic changes in the relationship between the response y and the design time point t are likely to happen. For example, the motorcycle data presented in Figure 1.1 shows that a dramatic change may happen somewhere between $[0.2, 0.3]$ and between $[0.5, 0.6]$. Thus, two knots must be placed within these two intervals. Of course, this method is subjective and rough. In practice, we may combine the subjective and objective approaches to locate the knots.

More sophisticated ways of placing knots are also available in the literature. For example, Friedman and Silverman (1989) proposed a series of stepwise knot selection methods to find the best set of knots based on the OLS estimator, where the selection of knots is restricted to a subset of design time points $(t_1, t_2, \ldots, t_n)$. The selection process is based on the minimization of the GCV score. Denote a knot sequence as a vector $\tau = (\tau_1, \tau_2, \ldots, \tau_K)^T$. The GCV score is calculated as

$$\text{GCV}(\tau) = \frac{||\mathbf{y} - \hat{\mathbf{y}}||^2}{n(1 - d(K)/n)^2}, \tag{2.16}$$

where $\mathbf{y}$ is the response vector and $\hat{\mathbf{y}}$ is its estimator as defined in (2.12), $d(K)$ is an increasing function of the number of knots, K. Friedman and Silverman (1989)

suggested to use $d(K) = 3K + 1$.

The process of the forward addition method, which is one of the stepwise knot selection methods is as follows. It starts with a model without any knots. After fitting the model, we calculate the GCV score, after which a new knot is added at each time step. Each new knot is chosen if it produces the largest reduction of GCV at each step. This process is repeated until the total number of knots reaches a size (usually taken to be $n/3$) or the GCV score stops decreasing. In this way, the optimal group of knots is obtained.

Although the more sophisticated ways of locating knots may produce better fitting results, they are usually computationally expensive. In practice, the uniform knot locating rule (2.14) and the quantile as knots rule (2.15) are the most common ways to locate knots for their simplicity. In this thesis, we mainly employ the uniform knot locating rule to place knots in our simulations and real data applications.

2.2.5 Methods for Choosing the Number of Knots

When the knot locating rule is specified, we need to choose the number of knots to give a good estimate to the underlying function $m(t)$ in the standard nonparametric regression model (2.1).

It is known that the choice of the number of knots is an even more crucial issue than locating the knots. This can be illustrated by two extreme situations. One

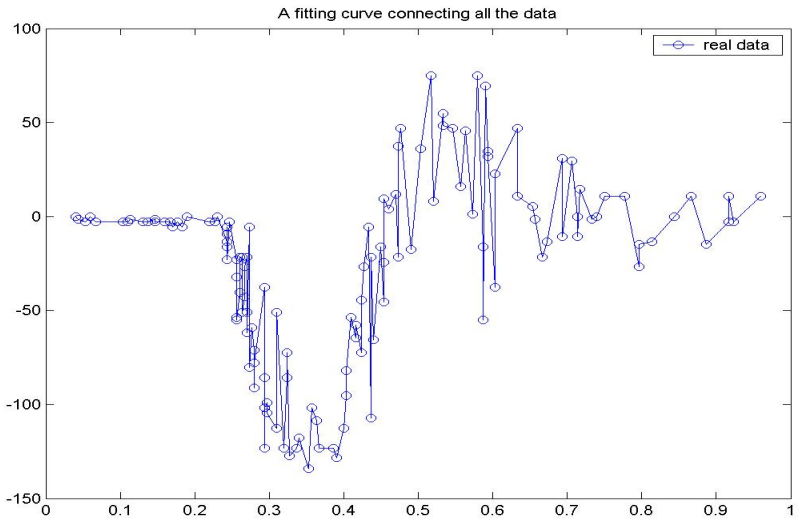

Figure 2.1: *An example of a usual regression spline fit to the motorcycle data set in which a too large number of knots is used.*

extreme situation is to choose as many knots as the number of distinct design points. This may produce a fit to $m(t)$, connecting all the data points. However, as we can see in Figure 2.1, the fit will result in a rapid fluctuation, producing an extremely complicated model. The increased model complexity will make the estimation and prediction more difficult as more computation and information would be needed. On the other hand, if we let the number of knots be too small, it will not fully reflect the pattern in the data (See Figure 2.2), yielding a high prediction error. In both cases, the estimation of the underlying function will not be good. Hence, we should compromise between the model complexity and the goodness of fit, when determining the number of knots. For this purpose, we introduce three methods below which are based on a tradeoff between the model complexity and the goodness of fit of the regression spline model. Other methods will be introduced in the next

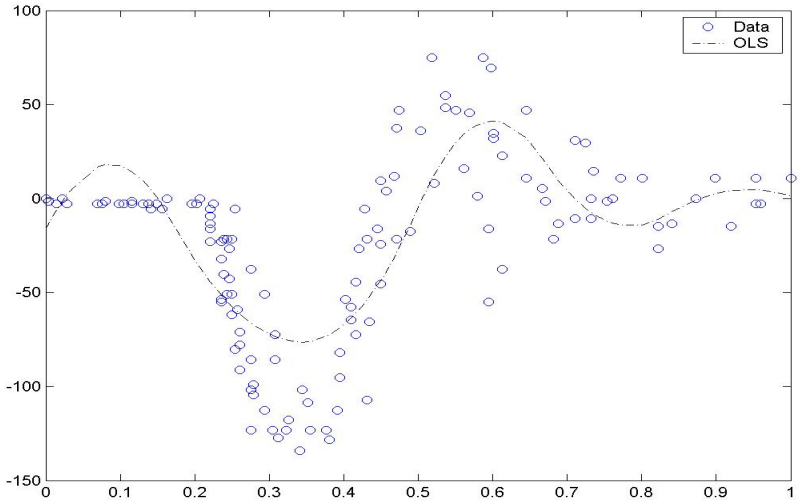

Figure 2.2: *An example of a usual regression spline fit to the motorcycle data set in which a too small number of knots is used.*

two subsections. **Generalized Cross Validation (GCV):** The GCV rule was described in the last subsection for knot locating. When the knot locating rule is specified, we can use GCV to choose a good number of knots too. When the TPB vector (2.7) is given and the order of the TPB, p, is fixed, choosing the knot number K is equivalent to choosing the parameter $\rho = p + K + 1$ which denotes the number of basis functions in the TPB vector. To define a model selection rule such as GCV, we need first define the quantities which measure the model complexity and the goodness of fit.

Notice that for regression spline modeling, the estimated response vector $\hat{\mathbf{y}}$ can be expressed as (2.12). Then the model complexity of regression spline modeling can be measured by the degree of freedom (DF) of regression spline modeling,

which is defined as, the trace of the smoother matrix $\mathbf{A}$:

$$DF_\rho = tr(\mathbf{A}) = p + K + 1 = \rho, \tag{2.17}$$

which increases as K increasing, indicating that the regression spline model is more complicated if a large number of knots K is used. At the same time, the goodness of fit of regression spline modeling can be measured by the sum of squared errors (SSE) defined as

$$SSE_\rho = ||\mathbf{y} - \hat{\mathbf{y}}||^2 = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{2.18}$$

which is small when the regression spline model fit the data closely. In the extreme case, when the fit passes through all the data points, $SSE = 0$, indicating a zero-error fit to the data, but it is obviously not a good fit to the data, as shown in Figure 2.1.

The Generalized Cross Validation (GCV) rule was first proposed by Craven and Wahba (1979). Since then, it has been widely used in various settings to select the optimal tuning parameters in a model. Here, the tuning parameter is ρ. The GCV score as a function of ρ is defined as

$$GCV_\rho = \frac{SSE_\rho}{n(1 - DF_\rho/n)^2}, \tag{2.19}$$

which trades-off the goodness of fit (measured by SSE_ρ) and the model complexity (measured by DF_ρ). The optimal choice of ρ is the one which minimizes (2.19).

Notice that the GCV defined above is slightly different from the GCV defined in (2.16) where the model complexity $d(K) = 3K + 1$ is subjectively chosen.

Akaike Information Criterion(AIC): The AIC rule (Akaike 1973) can be used to select a good number of knots, which is also constructed to trade off the goodness of fit and the model complexity. The AIC is defined as follows:

$$AIC_\rho = log(SSE_\rho) + 2DF_\rho/n, \tag{2.20}$$

since when ρ is large, $log(SSE_\rho)$ becomes small while DF_ρ becomes large. The optimal ρ is chosen to result in the smallest AIC_ρ. It has been shown in the literature that AIC will lead to a large model which may under-smooth the data. To overcome this problem, Schwarz (1978) proposed the Bayesian Information criterion (BIC), which places more penalty on the model complexity. The BIC is defined as,

$$BIC_\rho = log(SSE_\rho) + log(n)DF_\rho/n, \tag{2.21}$$

which is obtained via replacing the 2 of (2.20) with $log(n)$, which is much larger than 2 when n is large. As a result, the optimal ρ which minimizes the BIC_ρ is usually smaller than the optimal ρ which minimizes the AIC_ρ. Compared to the resulting model by AIC, the model chosen by BIC is simpler.

2.2.6 Knot Choosing via Best Subset Selection

When the knots are well located and the knot number is properly chosen as stated in the previous subsection, we can fit the regression spline model (2.8) using the ordinary least squares method as stated in Section 2.2.3. This method has been discussed by Friedman and Silverman (1989), Kooperberg, Bose and Stone (1997), and Lee (2000) among others.

The major limitation of the OLS method is that, to achieve a stable estimator, only a small number of knots is allowed in the model (usually ≤ 10). However, under this restriction, the quality of the associated estimator will be very sensitive to the location of the knots, implying that the method will not perform well if the location of the knots is inappropriately selected. For example, if there is a large fluctuation in the underlying curve, introducing a small size of knots will lead to an over-smoothing estimator, which fails to identify the change in the curve pattern at important locations. Since, in practice, the underlying curve is unknown, we intend to choose a large size of knots.

Usually, a large number of initial knots are introduced into the model at the very beginning. In this case, we can use the best subset method to remove some knots that are less important for estimating and predicting the underlying curve. This is actually a variable selection method for the regression spline model (2.8), via applying the best subset method to the regression spline coefficients. In particular, if we restrict the variable selection only on the coefficients of the truncated power basis functions, i.e., $\beta_{p+k}, k = 1, 2, \cdots, K$, the method works like removing the knot τ_k, if the coefficient β_{p+k} is insignificant and removed from the model.

The best subset method works as follow. For a standard linear regression model, e.g., (2.9) (the regression spline model (2.8) is a standard linear model when the basis vector $\mathbf{\Psi}(t)$ is specified and fixed), it introduces only one covariate into the model at each step, and tests significance of all the coefficients included in the

model. If any coefficient is insignificant, it will be removed from the model. The process stops when no covariate can be added in or removed from the model. However, it is known that the best subset method is computationally expensive. Breiman (1995) also shows that, if a single data case (t_i, y_i) is removed from the data set, the selected covariates will be different from the original ones when the same stepwise procedure is applied. This means that, a small perturbation in the data would lead to a drastic change in the estimated regression function.

2.2.7 Knot Choosing via SCAD Method

When a large number of knots are introduced into the regression spline model (2.8) at the very beginning (sometimes, the size of the knots K is even larger than n), the regression spline coefficient vector β may be sparse in the sense that many of the regression spline coefficients are zero or nearly so. In this case, we can choose the number of the significant knots, which is smaller than K, and estimate the regression coefficients β simultaneously via using a penalized least squares method, such as the SCAD method of Fan and Li (2001). That is, we minimize the following penalized least squares criterion,

$$\frac{1}{2}\sum_{i=1}^{n}\{y_i - \mathbf{\Psi}(t_i)^T\beta\}^2 + \sum_{j=0}^{\rho-1} p_\lambda(|\beta_j|), \tag{2.22}$$

where as before $\rho = K + p + 1$ is the number of basis functions, n is the number of observations and $p_\lambda(|\theta|)$ is some penalized function which is able to truncate those insignificant coefficients into 0 so that the resulting estimates of the regression

spline coefficients are sparse. Notice that when we shrink the regression coefficient, β_{p+k}, of the kth truncated power basis function, $(t-\tau_k)_+^p$, to 0, it is equivalent to removing the kth knot τ_k from the knot sequence (2.2). Sometimes, in the above criterion, we may make the penalty be applied only to those coefficients of the truncated power functions, i.e., $\beta_{p+k}, k = 1, 2, \cdots, K$.

Notice that different penalty functions will result in different estimates of β. For example, when the so-called L_2 penalty is used, i.e., $p_\lambda(|\theta|) = \lambda\theta^2$, the associated method is known as the penalized regression spline method (Ruppert and Carroll 1997). When the so-called L_1 penalty is applied, $p_\lambda(|\theta|) = \lambda|\theta|$, the associated method is known as Lasso (the Least Absolute Shrinkage and Selection Operator); see Tibshirani (1996). In general, a L_q penalty may be used, i.e., $p_\lambda(|\theta|) = \lambda|\theta|^q$ (Frank and Friedman 1993 and Fu 1998). Other penalty functions including the hard thresholding function $p_\lambda(|\theta|) = \lambda^2 - (|\theta| - \lambda)^2 I(|\theta| < \lambda)$ (Antoniadis and Fan 1997), the entropy penalty function $p_\lambda(|\theta|) = (\lambda^2/2) I(|\theta| \neq 0)$ and the SCAD (Smoothly Clipped Absolute Deviation) penalty function (Fan and Li 2001) can also be applied.

A good penalty function should satisfy several properties. First, the penalty function should be continuous to provide regularity. Second, it should be non-decreasing with regard to $|\theta|$, which means coefficients with larger absolute values receive heavier penalties. Last but not the least, in order to simplify the model, the penalty function should be singular at the origin to produce sparse estimators, i.e.,

to truncate those insignificant coefficients to zero. Notice here, singularity means a discontinuous derivative at zero. Lasso and the hard thresholding penalty, as well as the SCAD penalty satisfy these properties.

But the SCAD method is shown to perform better than others, and hence we shall use the SCAD method to select and estimate the regression spline coefficients simultaneously. For the SCAD method, details are given in the next section.

2.3 SCAD Method for Variable Selection in Linear Models

2.3.1 SCAD Penalized Function

In this section, we shall review the SCAD method of Fan and Li (2001) for variable selection for linear models. When the basis vector $\mathbf{\Psi}(t)$ is fixed, the regression spline model (2.8) is a standard linear regression model and hence the SCAD method can be applied directly to estimate the regression spline coefficient vector.

Consider the following linear regression model

$$\mathbf{y} = \mathbf{X}\beta + \epsilon, \quad \epsilon \sim N(0, \sigma^2 \mathbf{I}_n), \tag{2.23}$$

where $\mathbf{y}$ is an $n \times 1$ response vector, $\mathbf{X}$ is an $n \times d$ design matrix, β is a $d \times 1$ coefficient vector and ϵ is the $n \times 1$ random error. To select covariates and estimate coefficients simultaneously, Fan and Li (2001) proposed the smoothly clipped absolute deviation (SCAD) method, estimating β by minimizing the following penalized

least squares function

$$\frac{1}{2}||\mathbf{y} - \mathbf{X}\beta||^2 + \sum_{j=1}^{d} p_\lambda(|\beta_j|), \tag{2.24}$$

where $||\cdot||$ is the L_2-norm of a vector, and $p_\lambda(\cdot)$ is the SCAD penalty in the following form:

$$p_\lambda(|\theta|) = \begin{cases} \lambda|\theta| & \text{when } |\theta| \leq \lambda \\ -(|\theta|^2 - 2a\lambda|\theta| + \lambda^2)/[2(a-1)] & \text{when } \lambda < |\theta| \leq a\lambda \\ (a+1)\lambda^2/2 & \text{when } |\theta| > a\lambda \end{cases}$$

for some $a > 2$ and $\lambda > 0$.

The first derivative of $p_\lambda(|\theta|)$ can be written in the following equation:

$$p_\lambda^{'}(|\theta|) = \lambda\{I(|\theta| \leq \lambda) + \frac{(a\lambda - |\theta|)_+}{(a-1)\lambda} I(|\theta| > \lambda)\}, \quad \theta \neq 0. \tag{2.25}$$

Notice $p_\lambda^{'}(|\theta|)$ does not exist at $\theta = 0$ and vanishes when $|\theta| > a\lambda$. This demonstrates the advantages of using the SCAD method: SCAD produces sparse estimates and meanwhile, may leave large coefficients unbiased. To implement this method, the tuning parameters λ and a, denoted as $\eta = (\lambda, \mathbf{a})$, can be selected by a data-driven method, which will be discussed later.

2.3.2 Explicit Solution when the Design Matrix is Orthonormal

When the design matrix $\mathbf{X}$ is orthonormal, we can give an explicit solution to the penalized least squares criterion (2.24). In fact, under this condition, the

componentwise solution to (2.24) can be expressed as

$$\hat{\beta}_j = \begin{cases} sgn(z_j)(|z_j| - \lambda)_+, & \text{when } |z_j| \leq 2\lambda, \\ \{(a-1)z_j - sgn(z_j)a\lambda\}/(a-2), & \text{when } 2\lambda < |z_j| \leq a\lambda, \\ z_j, & \text{when } |z_j| > a\lambda, \end{cases}$$

for $j = 1, 2, \cdots, d$, where $\mathbf{z} = (z_1, \cdots, z_d)^T = \mathbf{X}^T\mathbf{y}$ is the ordinary least square estimator of β.

We can show the above expression as follows. Notice that under the orthonormality assumption, $\mathbf{z} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \mathbf{X}^T\mathbf{y}$ is the OLS estimator of β since $\mathbf{X}$ is orthonormal so that $\mathbf{X}^T\mathbf{X} = \mathbf{I}$, and $\hat{\mathbf{y}} = \mathbf{X}\mathbf{X}^T\mathbf{y}$.

Proof: First, we examine the penalized least square function,

$$\begin{aligned} \frac{1}{2}||\mathbf{y} - \mathbf{X}\beta||^2 + \sum_{j=1}^{d} p_\lambda(|\beta_j|) &= \frac{1}{2}||\mathbf{y} - \hat{\mathbf{y}} + \hat{\mathbf{y}} - \mathbf{X}\beta||^2 + \sum_{j=1}^{d} p_\lambda(|\beta_j|) \\ &= \frac{1}{2}||\mathbf{y} - \hat{\mathbf{y}}||^2 + \frac{1}{2}||\hat{\mathbf{y}} - \mathbf{X}\beta||^2 + \sum_{j=1}^{d} p_\lambda(|\beta_j|) \\ &= \frac{1}{2}||\mathbf{y} - \hat{\mathbf{y}}||^2 + \frac{1}{2}||\mathbf{z} - \beta||^2 + \sum_{j=1}^{d} p_\lambda(|\beta_j|), \end{aligned}$$

where we use the fact $||\hat{\mathbf{y}} - \mathbf{X}\beta||^2 = ||\mathbf{z} - \beta||^2$ due to the orthonomality of $\mathbf{X}$.

Since $\frac{1}{2}||\mathbf{y} - \hat{\mathbf{y}}||^2$ is a constant, minimizing the above function is equivalent to minimizing

$$\sum_{j=1}^{d} \left\{ \frac{1}{2}(z_j - \beta_j)^2 + p_\lambda(|\beta_j|) \right\}. \tag{2.26}$$

Taking the partial derivative of (2.26) with respect to β_j, we get,

$$-(z_j - \beta_j) + p'_\lambda(|\beta_j|)sgn(\beta_j) = 0.$$

Plugging (2.25) into the above expression, we get

$$-z_j + \beta_j + \lambda\{I(|\beta_j| \leq \lambda) + \frac{(a\lambda - |\beta_j|)_+}{(a-1)\lambda} I(|\beta_j| > \lambda)\} sgn(\beta_j) = 0. \tag{2.27}$$

Notice that when $|\beta_j| > a\lambda$, we have $|\beta_j| > \lambda$ since $a > 2$. It follows that the third term of the left hand side of (2.27) is 0 and hence the solution to (2.27) is

$$\hat{\beta}_j = z_j.$$

On the other hand, when $\lambda < |\beta_j| \leq a\lambda$, careful derivation leads to that the solution to (2.27) is,

$$\beta_j = \frac{(a-1)z_j - a\lambda sgn(z_j)}{a-2}.$$

as desired. The proof is completed.

2.3.3 Iterative Solution when the Design Matrix is not Orthonormal

When the design matrix $\mathbf{X}$ is not orthonormal, an explicit solution to (2.24) is hard to obtain. However, we may obtain an iterative solution which will be described in this subsection.

Notice when the sample size n grows larger, the first part of (2.24), i.e, $\frac{1}{2}||\mathbf{y} - \mathbf{X}\beta||^2$ will have more weight in the penalty function, which makes the penalty inconsistent. To overcome this problem, we incorporate n into the second part of (2.24). Therefore, the penalty function is redefined as

$$\frac{1}{2}||\mathbf{y} - \mathbf{X}\beta||^2 + n\sum_{j=1}^{d} p_\lambda(|\beta_j|), \tag{2.28}$$

The iterative algorithm for solving (2.28) is as follows. Given an initial value β_0, which is assumed to be close to the true value of β. When the jth component β_{j0} is very close to 0, we set $\hat{\beta}_{j0} = 0$. Otherwise, we set $\hat{\beta}_{j0} = \beta_{j0}$ and approximate $[p_\lambda(|\beta_j|)]'$ as

$$[p_\lambda(|\beta_j|)]' = p'_\lambda(|\beta_j|)sgn(\beta_j) = \frac{p'_\lambda(|\beta_j|)}{|\beta_j|}\beta_j \approx \frac{p'_\lambda(|\beta_{j0}|)}{|\beta_{j0}|}\beta_j. \tag{2.29}$$

It is equivalent to approximating $p_\lambda(|\beta_j|)$ by its Taylor expansion, because, when $\beta_j \approx \beta_{j0}$,

$$\begin{aligned} p_\lambda(|\beta_j|) &\approx p_\lambda(|\beta_{j0}|) + p'_\lambda(|\beta_{j0}|)(|\beta_j| - |\beta_{j0}|) \\ &= p_\lambda(|\beta_{j0}|) + p'_\lambda(|\beta_{j0}|)\frac{\beta_j^2 - \beta_{j0}^2}{|\beta_j| + |\beta_{j0}|} \\ &\approx p_\lambda(|\beta_{j0}|) + \frac{1}{2}p'_\lambda(|\beta_{j0}|)\frac{\beta_j^2 - \beta_{j0}^2}{|\beta_{j0}|}. \end{aligned}$$

Taking the first derivative of both sides of the equation with respect to β_j, we get,

$$[p_\lambda(|\beta_j|)]' = \{p'_\lambda(|\beta_{j0}|)/|\beta_{j0}|\}\beta_j,$$

which is the same as (2.29). Once $\hat{\beta}_0$ is obtained, we denote $\tilde{\beta}_0$ as the non-zero components of $\hat{\beta}_0$. This is actually a variable selection procedure: the important coefficients, .i.e., the components in $\tilde{\beta}_0$, are selected and estimated simultaneously, and the unimportant coefficients are shrunken to zero. Solving the penalized least squares problem in (2.28) produces an iterative solution:

$$\hat{\beta}_{k+1} = \{\mathbf{X}_k^T\mathbf{X}_k + n\Sigma_\lambda(\tilde{\beta}_k)\}^{-1}\mathbf{X}_k^T\mathbf{y}, \quad for\ k = 0, 1, 2, \ldots, \tag{2.30}$$

where

$$\Sigma_\lambda(\tilde{\beta}_k) = diag\{p'_\lambda(|\tilde{\beta}_{1k}|)/|\tilde{\beta}_{1k}|, \ldots, p'_\lambda(|\tilde{\beta}_{hk}|)/|\tilde{\beta}_{hk}|\}.$$

Notice h is the dimension of $\tilde{\beta}_k$, and $\mathbf{X}_k$ is the corresponding sub design matrix related to $\tilde{\beta}_k$. For $\hat{\beta}_{k+1}$, its dimension is the same as $\tilde{\beta}_k$. Again, we need to select important covariates in $\tilde{\beta}_{k+1}$ and shrink insignificant coefficients to zero. This is done by resetting $\hat{\beta}_{j(k+1)} = 0$, if the jth component $\hat{\beta}_{j(k+1)}$ is close to zero. Then, by eliminating the zero components of $\hat{\beta}_{k+1}$, we get $\tilde{\beta}_{k+1}$. Repeating (2.30) until $||\hat{\beta}_{l+1} - \tilde{\beta}_l|| < \delta$, where δ is a pre-specified number of precision, usually taken to be 10^{-4}.

An important issue associated with the above iterative algorithm is when to set $\hat{\beta}_{j(k+1)} = 0$. This problem is equivalent to identifying the insignificant coefficients in $\hat{\beta}_{k+1}$. Actually, it can be solved by t-test. From (2.30), the estimated covariance matrix for $\hat{\beta}_{k+1}$ can be derived as follows,

$$\Sigma = \widehat{cov}(\hat{\beta}_{k+1}) = \{\mathbf{X}_k^T\mathbf{X}_k + \Sigma_\lambda(\tilde{\beta}_k)\}^{-1}\mathbf{X}_k^T\mathbf{X}_k\{\mathbf{X}_k^T\mathbf{X}_k + n\Sigma_\lambda(\tilde{\beta}_k)\}^{-1}\hat{\sigma}^2,$$

where

$$\hat{\sigma}^2 = \frac{||\mathbf{y} - \mathbf{X}_k(\mathbf{X}_k^T\mathbf{X}_k)^{-1}\mathbf{X}_k^T\mathbf{y}||^2}{n-h},$$

and h is the dimension of $\tilde{\beta}_k$. Suppose the jth diagonal element of the estimated covariance matrix is Σ_{jj}, which is the estimated variance of $\hat{\beta}_{j(k+1)}$, then the t-statistic for the jth coefficient in $\hat{\beta}_{k+1}$ is $T_j = |\hat{\beta}_{j(k+1)}|/\sqrt{\Sigma_{jj}}$. If the jth coefficient appears to be insignificant, we reset $\hat{\beta}_{j(k+1)} = 0$. Otherwise, it remains unchanged.

In practice, the tuning parameter $\eta = (\lambda, a)$ is usually selected by minimizing the GCV score (Breiman, 1995). As mentioned earlier, the GCV rule in different settings may have different meanings. Here, it comes from the idea of a leaving-

out-one operation. Given $\mathbf{y}$ and $\mathbf{X}$, the above algorithm implies that $\hat{\beta}$ can be determined by the choice of η. To test the performance of a η at a given data point (t_i, y_i), we leave out this point, fitting the model and predict y_i, denoted as $\hat{y}_i^{(-i)}$. Applying this procedure on every single data point, we can compute the following cross validation criterion:

$$CV_\eta = n^{-1} \sum_{i=1}^{n} (y_i - \hat{y}_i^{(-i)})^2 \tag{2.31}$$

for each η. The best η is the one which minimizes (2.31). Computing (2.31) would be time-consuming, because for each data point y_i, we need to fit a new model and estimate $\hat{y}_i^{(-i)}$. By formalizing and generalizing (2.31), the GCV score, which is a function of η can be defined as follows,

$$GCV_\eta = \frac{||\mathbf{y} - \mathbf{X}\hat{\beta}_\eta||^2}{n(1 - DF_\eta/n)^2},$$

where

$$DF_\eta = tr\{\mathbf{X}(\mathbf{X}^T\mathbf{X} + n\Sigma_\lambda(\hat{\beta}_\eta))^{-1}\mathbf{X}^T\}.$$

For each step of the iteration in (2.30), we will select the best η, which can minimize GCV_η, Then, we use the best η to estimate the coefficients. Since η is two-dimensional, searching the best η can be computationally expensive. Fan and Li (2001) showed that SCAD is not very sensitive to the values of a. And from a perspective of Bayes risk, $a = 3.7$ is the best choice as it leads to the minimum Bayes risk.

Fixing $a = 3.7$, the GCV score is only a function of λ. If we denote $SSE_\lambda =$

$||\mathbf{y} - \mathbf{X}\hat{\beta}_\lambda||^2$ as done in Subsection 2.2.5,

$$\begin{aligned} log(GCV_\lambda) &= log(SSE_\lambda) - 2log(1 - DF_\lambda/n) - log(n) \\ &\approx log(SSE_\lambda) + 2DF_\lambda/n - log(n). \end{aligned}$$

In fact, $log(GCV_\lambda)$ is similar to the traditional model selection criterion AIC, which is $log(SSE_\lambda)+2DF_\lambda/n$. In the literature, AIC has been proven not consistent in the sense that it does not select the correct model with probability approaching 1 in large samples when the true model is of finite dimension (Wang 2007). Instead, Wang (2007) proposed the BIC criterion to select λ:

$$BIC_\lambda = log(SSE_\lambda) + DF_\lambda log(n)/n - log(n).$$

Theoretical results show that the BIC criterion is a consistent criterion to produce λ which can identify the true model with probability approaching 1 when the sample size grows larger.

According to Fan and Li (2001), SCAD estimates coefficients as well as if the true submodel is known. This is called an oracle property. The drawback of this method is that once a coefficient is shrunken to zero, it will stay at zero in the following iterations. However, this method significantly reduces the computational burden.

Chapter 3

Regression Spline Smoothing via Penalizing Derivatives

3.1 Introduction

In the previous chapters, we review the usual regression spline smoothing method. This method aims to fit the nonparametric regression model (2.1) which can be rewritten as follows:

$$y_i = m(t_i) + \epsilon_i, \quad i = 1, 2, \ldots, n, \quad t \in [0, 1]. \tag{3.1}$$

Given a noisy data set $(t_1, y_1), (t_2, y_2), \ldots, (t_n, y_n)$ and the truncated power basis

$$\mathbf{\Psi}(t) = (1, t, t^2, \ldots, t^p, (t - \tau_1)_+^p, \ldots, (t - \tau_K)_+^p)^T, \tag{3.2}$$

with a sequence of K given knots $\tau_1, \tau_2, \cdots, \tau_K$, the nonparametric regression model (3.1) can be approximated by the regression spline model (2.8), i.e.,

$$y_i = f(t_i) + \epsilon_i, i = 1, 2, \cdots, n, \tag{3.3}$$

where the regression spline function is

$$f(t_i) = \mathbf{\Psi}(t_i)^T \beta = \sum_{r=0}^{p} \beta_r t_i^r + \sum_{k=1}^{K} \beta_{p+k} (t_i - \tau_k)_+^p, \quad t \in [0, 1]. \tag{3.4}$$

In situations when the regression spline coefficients $\beta_r, r = 0, 1, \cdots, p + K$ are sparse, we can employ the SCAD method of Fan and Li (2001) directly to the above model (3.3) as described previously. However, in many cases, the regression spline coefficients may be not very sparse but the derivatives of the regression spline function are sparse. Take Simulation 2 in Section 3.3.2 for example: the coefficients of the truncated power functions are $(2, -2, 2, 2, -4, 2, -2)^T$ and the first derivatives of the regression spline function within different intervals are $(2, 0, 2, 4, 0, 2, 0)^T$. Obviously, the former vector is not sparse, while the latter one is sparse. Therefore, directly applying the SCAD method to the original coefficient vector is less effective. Another example is the motorcycle data which will be discussed in Section 3.4.1. Figure 3.1 shows that, for fitting the motorcycle data, the SCAD estimator of the regression spline coefficients of the cubic truncated power basis model are not very sparse, but the re-parameterized coefficients via penalizing derivatives are sparse.

In the above-mentioned cases, we can re-parameterize the regression spline model (3.3) in terms of its pth times derivatives. There are two reasons for using the

pth times derivative in our proposed method. First, it is much easier to establish a direct relationship between the regression spline coefficients $\beta_r, r = 0, 1, \cdots, p+K$ and the pth times derivatives of the regression spline function than other lower order derivatives. Second, it is known that the regression spline model results in a piecewise polynomial. Given p is large enough to capture all the different patterns in the underlying function, the pth times derivatives of the function are zero for the intervals with polynomial orders less than p.

After re-parameterizing the coefficient vector into a new vector which is sparser, we arrive at a new regression spline model. We then apply the SCAD method to the new model. This chapter is organized as follows. In section 3.2, we present the regression spline smoothing method via penalizing derivatives. In Subsection 3.2.1, we show how to re-parameterize the regression spline model (3.3). In Subsection 3.2.2, we shall discuss how to determine the tuning parameters. Subsection 3.3.3 presents some asymptotic properties of the newly proposed estimator. Section 3.3 is devoted to present two simulation studies. Two real data examples are given in Section 3.4. We conclude this chapter by some discussions in Section 3.5.

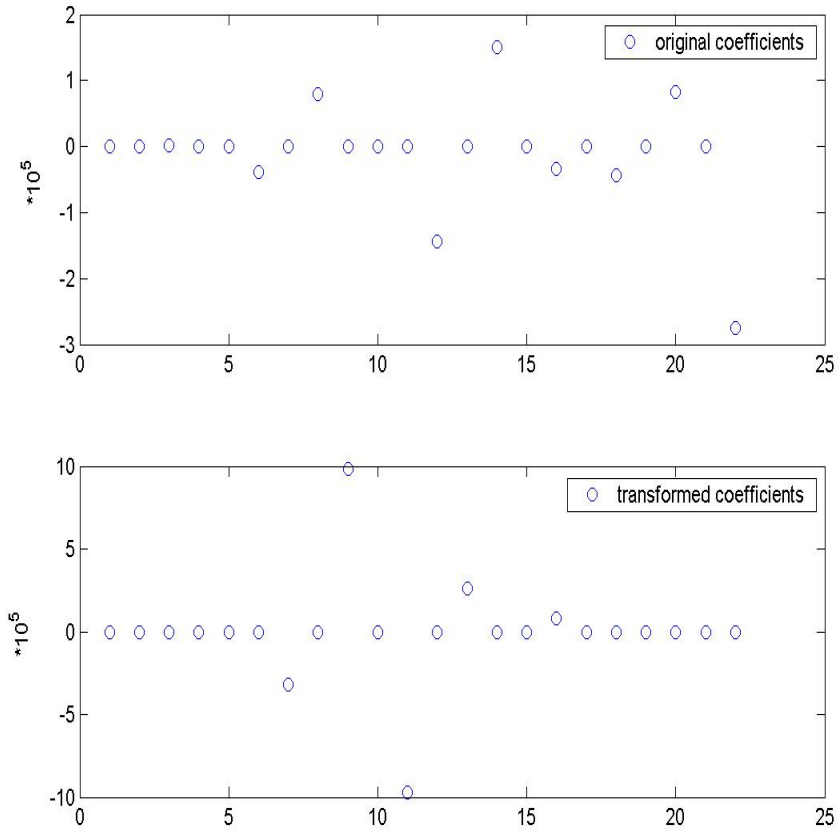

Figure 3.1: *The transformed coefficients are sparser than the original coefficients of the cubic truncated power basis model for the motorcycle data.*

3.2 Re-parameterizing the Regression Spline Model

3.2.1 Model Representation

Notice that the regression spline function (3.4), has only up to $(p-1)$ times continuous derivatives at knots, but its pth times derivatives exist at all the non-knot points. In fact, as presented in (2.5) of Chapter 2, we have

$$f^{(p)}(t) = p! \left\{\beta_p + \beta_{p+1} + \cdots + \beta_{p+k}\right\}, \quad t \in (\tau_k, \tau_{k+1}), \tag{3.5}$$

for $k = 0, 1, \cdots, K$ where $\tau_0 = 0$ and $\tau_{K+1} = 1$ are two boundary knots. That is, within any two neighboring knots, the pth order derivatives are constants. This is an important observation since we only need at most $K+1$ constants to represent all the pth order derivatives of the regression spline $f(t)$. To re-parameterize the regression spline model (3.3), set

$$\begin{aligned}
\gamma_0 &= \beta_0, \\
\gamma_1 &= \beta_1, \\
\cdots \quad &\cdots \quad \cdots, \\
\gamma_{p-1} &= \beta_{p-1}, \\
\gamma_p &= p!\beta_p, \\
\gamma_{p+1} &= p!(\beta_p + \beta_{p+1}), \\
\cdots \quad &\cdots \quad \cdots, \\
\gamma_{p+K} &= p!(\beta_p + \beta_{p+1} + \cdots + \beta_{p+K}).
\end{aligned}$$

Notice that here $\gamma_r = \beta_r, r = 0, 1, \cdots, p-1$ are not directly related to the pth order derivatives but are related to the first p terms of the polynomial function (2.4), and only $\gamma_{p+k}, k = 0, 1, \cdots, K$ are directly related to or represent the pth order derivative. For convenience, we rewrite the above relationship in terms of matrix and vector:

$$\gamma = \mathbf{A}\beta, \tag{3.6}$$

where $\gamma = (\gamma_0, \gamma_1, \ldots, \gamma_{p+K})^T$ and

$$\mathbf{A} = \begin{pmatrix} \mathbf{I}_p & 0 \\ 0 & \mathbf{B} \end{pmatrix}$$

where $\mathbf{I}_p$ denotes an $p \times p$ identity matrix and $\mathbf{B}$ is a $(K+1) \times (K+1)$ matrix defined as

$$\mathbf{B} = p! \begin{pmatrix} 1 & 0 & \ldots & 0 \\ 1 & 1 & \ldots & 0 \\ \vdots & \vdots & \ldots & \vdots \\ 1 & 1 & \ldots & 1 \end{pmatrix}.$$

We refer the matrix $\mathbf{A}$ as the "link matrix", as it connects the original regression spline coefficient vector β with the γ. It is seen that the link matrix is invertible since both $\mathbf{I}_p$ and $\mathbf{B}$ are invertible. In fact,

$$\mathbf{A}^{-1} = \begin{pmatrix} \mathbf{I}_p & 0 \\ 0 & \mathbf{B}^{-1} \end{pmatrix}, \tag{3.7}$$

where

$$\mathbf{B}^{-1} = p!^{-1} \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ -1 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \dots & -1 & 1 \end{pmatrix}.$$

Thus, we can express β in terms of γ:

$$\beta = \mathbf{A}^{-1}\gamma. \tag{3.8}$$

That is,

$$\begin{aligned} \beta_r &= \gamma_r, r = 0, 1, 2, \cdots, p-1, \\ \beta_p &= \gamma_p/p!, \\ \beta_{p+k} &= (\gamma_{p+k+1} - \gamma_{p+k})/p!,\ k = 1, 2, \cdots, K. \end{aligned}$$

From the above expressions, it is easy to obtain an estimate of β, provided we have an estimate of γ.

In terms of vector and matrix, we can rewrite the regression spline model (3.3) as

$$\mathbf{y} = \mathbf{X}\beta + \epsilon, \tag{3.9}$$

where $\mathbf{y} = (y_1, y_2, \dots, y_n)^T$ denotes the response vector, $\beta = (\beta_0, \beta_1, \dots, \beta_{p+K})^T$ denotes the coefficient vector, $\epsilon \sim N(0, \sigma^2\mathbf{I}_n)$ denotes the noise vector, and $\mathbf{X}$ is

the associated design matrix based on the truncated power basis vector, i.e.,

$$\begin{pmatrix} 1 & t_1 & t_1^2 & \dots & t_1^p & (t_1-\tau_1)_+^p & \dots & (t_1-\tau_K)_+^p \\ 1 & t_2 & t_2^2 & \dots & t_2^p & (t_2-\tau_1)_+^p & \dots & (t_2-\tau_K)_+^p \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & t_n & t_n^2 & \dots & t_n^p & (t_n-\tau_1)_+^p & \dots & (t_n-\tau_K)_+^p \end{pmatrix}.$$

Plugging (3.6) into (3.9), we arrive at the transformed model

$$\mathbf{y} = \mathbf{V}\gamma + \epsilon, \tag{3.10}$$

where $\mathbf{V} = \mathbf{X}\mathbf{A}^{-1}$ with $\mathbf{A}^{-1}$ computed using (3.7). Thus, the regression model (3.10) is the re-parameterization of the regression spline model (3.3) in terms of γ.

When β is not very sparse, while γ is sparse, applying the SCAD method of Fan and Li (2001) to the transformed model (3.10) is more efficient than to the original regression spline model (3.3). In this case, once $\hat{\gamma}$ is obtained, $\hat{\beta}$ can be obtained using the relationship (3.8), i.e.,

$$\hat{\beta} = \mathbf{A}^{-1}\hat{\gamma}.$$

We call this newly proposed method as "regression spline smoothing via penalizing derivatives". The performance of this method will be demonstrated in the sections of simulations and real data analysis.

3.2.2 Choice of the Tuning Parameters

In the re-parameterized regression spline model (3.10), there are three tuning parameters which we need to deal with: the order p of the truncated power basis

(3.2), the locations of the knots and the number of the knots. The later two tuning parameters can be selected using the methods described in Chapter 2 since the re-parameterized regression spline model (3.10) is also a linear regression model when the basis vector $\mathbf{\Psi}(t)$ is specified. Here we just consider how to select p.

A naive method is to select p based on the scatter plot of the data. We select p based on whether the pth times derivatives of the function are sparse. The scatter plot of the data may hint what kind of p is able to make the pth times derivatives sparse. For example, the scatter plot of the motorcycle data in Figure 1.1 implies that a cubic ($p = 3$) truncated power basis may be proper since at different ranges, the linear, quadratic or cubic polynomial models can fit the data at the ranges respectively. Therefore, when a cubic truncated power basis is applied, the third times derivatives will be 0 at most of the design time points and hence these derivatives are sparse.

Another method is to select p using the GCV rule. This method is slightly more objective and automatically. It selects p via minimizing the associated GCV score. For each given p, we can compute the best GCV score (2.19) via finding the best number of knots, K. Then comparing all the associated best GCV scores for all the p considered. Select the p such that the associated GCV score is the smallest. Usually, the total number of p's considered is small. For the motorcycle data, we only need consider $p = 1, 2, 3, 4$ since we know that the larger p is not necessary. We will demonstrate how to select the optimal p and K via simulation studies.

3.2.3 Asymptotic Properties

The asymptotic properties of the newly proposed estimator are studied under the general assumption that, as the sample size $n \to \infty$, the dimension of β, i.e., $p + K + 1$, goes to infinity. Since, p is usually taken to be $1, 2, or 3$, we may regard p as pre-determined. Therefore, the above assumption also means that as $n \to \infty$, $K \to \infty$.

First, let us focus on the transformed model (3.10):

$$\mathbf{y} = \mathbf{V}\gamma + \epsilon.$$

The coefficient vector γ is estimated by SCAD with the penalized least squares function (2.24) and is assumed sparse. We denote the estimator of γ as $\hat{\gamma}_n$. We use the subscript n to show that $\hat{\gamma}$ may change with n. We write the nonzero components of γ as γ_1 and zero components as γ_2. For simplicity, γ can be written as

$$\gamma = (\gamma_1^{'}, \gamma_2^{'})^{'},$$

where $\gamma_1^{'} = (\gamma_1, \ldots, \gamma_{h_n})$ and $\gamma_2^{'} = (0, \ldots, 0)$. Here, let $k_n = p + K + 1$ be the dimension of γ, and h_n is the number of nonzero components of γ; $m_n = k_n - h_n$ is the number of zero components of γ. Similar to the partition of γ, $\mathbf{V}$ can be divided into two parts: $V = (V_1, V_2)$ where V_1 and V_2 are $n \times h_n$ and $n \times m_n$ matrices.

Huang and Xie (2007) studied the asymptotic properties of the least squares SCAD estimator. We can employ their conclusions on $\hat{\gamma}_n$. Let $\rho_{n,1}$ be the smallest eigenvalue of $n^{-1}\mathbf{V}'\mathbf{V}$. π_{n,h_n} and ω_{n,m_n} are the largest eigenvalues of $n^{-1}\mathbf{V}_1'\mathbf{V}_1$

and $n^{-1}\mathbf{V}_2'\mathbf{V}_2$ respectively. The following conditions on the design matrix $\mathbf{V}$ are necessary for the conclusions (see Huang and Xie 2007):

A0 (a) The design matrix $\mathbf{V}$ is fixed, and only $\mathbf{y}$ is random; (b) For any $j \in \{1, \ldots, k_n\}$, $||\mathbf{V}_{.j}||^2 = n$; (c) ϵ_i's are i.i.d with mean 0 and variance σ^2.

A1 (a) $\lim_{n\to\infty} \sqrt{h_n}\lambda_n/\sqrt{\rho_{n,1}} = 0$; (b) $\lim_{n\to\infty} \sqrt{k_n}/\sqrt{n\rho_{n,1}} = 0$.

A2 (a) $\lim_{n\to\infty} \sqrt{h_n}\lambda_n/(\sqrt{\rho_{n,1}} \min_{1\le j\le h_n} |\gamma_j|) = 0$;

(b)$\lim_{n\to\infty} \sqrt{k_n}/(\sqrt{n\rho_{n,1}} \min_{1\le j\le h_n} |\gamma_j|) = 0$; (c) $\lim_{n\to\infty} \sqrt{p_n/n}/\rho_{n,1} = 0$.

A3 $\lim_{n\to\infty} \sqrt{max(\pi_{n,h_n}, \omega_{n,m_n})k_n}/(\sqrt{n}\rho_{n,1}\lambda_n) = 0$.

A4 $\lim_{n\to\infty} \max_{1\le i\le n} \mathbf{V}_{i1}'(\sum_{i=1}^n \mathbf{V}_{i1}\mathbf{V}_{i1}')^{-1}\mathbf{V}_{i1} = 0$

Under $A0 - A4$, the asymptotic properties of $\hat{\gamma}_n$ are as follows.

Property 1:

$$||\hat{\gamma}_n - \gamma|| \xrightarrow{P} 0 \quad as\ n \to \infty$$

Property 2:

$$||\hat{\gamma}_n - \gamma|| = O_p(\sqrt{k_n/n}/\rho_{n,1})$$

Property 3:

$$\hat{\gamma}_{2n} = 0_{m_n}$$

with probability tending to 1.

Property 4:

$$\sqrt{n}\Sigma_n^{-1/2}\mathbf{D}_n(\hat{\gamma}_{1n} - \gamma_1) \xrightarrow{D} N(0_d, I_d),$$

where $\mathbf{D}_n, n = 1, 2, \ldots$ are a sequence of matrices of dimension $d \times h_n$ with full row rank and $\sum_n = \sigma^2\mathbf{D}_n(\sum_{i=1}^n \mathbf{V}_{i1}\mathbf{V}_{i1}'/n)^{-1}\mathbf{D}_n{}'$.

Property 1 shows that the least squares SCAD estimator is a consistent estimator: as n grows larger, the estimator converges to the true value with probability tending to 1. In particular, property 2 gives the convergence rate of the estimator. Property 3 and Property 4 together demonstrate the oracle properties of the SCAD estimator, that is, when the true coefficients have some zero components, they are estimated as 0 with probability tending to 1, and the nonzero components are estimated as well as when the correct submodel is known.

Based on Property 1-4 of $\hat{\gamma}_n$, some properties of $\hat{\beta}_n$ can be derived.

Theorem 1:

$$||\hat{\beta}_n - \beta|| \xrightarrow{P} 0 \quad as\ n \to \infty$$

Proof: Let $\hat{\beta}_n$ be the estimator of β, and $\mathbf{A}_n$ be the link matrix described in Subsection 3.2.1. In Property 1, we have

$$||\hat{\gamma}_n - \gamma|| \xrightarrow{P} 0 \quad as\ n \to \infty.$$

Since, $\hat{\beta}_n = \mathbf{A}_n^{-1}\hat{\gamma}_n$ and $\beta = \mathbf{A}_n^{-1}\gamma$, we have,

$$||\hat{\beta}_n - \beta|| = ||\mathbf{A}_n^{-1}(\hat{\gamma}_n - \gamma)||.$$

Here,

$$\mathbf{A}_n^{-1} = \begin{pmatrix} \mathbf{I}_p & 0 \\ 0 & \mathbf{B}^{-1} \end{pmatrix},$$

where

$$\mathbf{B}^{-1} = p!^{-1} \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ -1 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \dots & -1 & 1 \end{pmatrix}.$$

In fact, γ can be divided into two parts: the first part is related to the first p terms of the polynomial function (2.4) and the second part is related to the p-th order derivatives of the regression spline function. Therefore, we can rewrite γ as $(\gamma_1^{*'}, \gamma_2^{*'})'$, where $\gamma_1^{*'} = (\gamma_0, \gamma_1, \dots, \gamma_{p-1})$ and $\gamma_2^{*'} = (\gamma_p, \gamma_{p+1}, \dots, \gamma_{p+K})$. Correspondingly, $\hat{\gamma}_n$, β and $\hat{\beta}_n$ can be divided as $(\hat{\gamma}_{1n}^{*'}, \hat{\gamma}_{2n}^{*'})'$, $(\beta_1^{*'}, \beta_2^{*'})'$, and $(\hat{\beta}_{1n}^{*'}, \hat{\beta}_{2n}^{*'})'$, respectively.

So,

$$||\hat{\beta}_n - \beta|| = \left|\left| \begin{pmatrix} \hat{\beta}_{1n}^* - \beta_1^* \\ \hat{\beta}_{2n}^* - \beta_2^* \end{pmatrix} \right|\right| = \left|\left| \begin{pmatrix} \hat{\gamma}_{1n}^* - \gamma_1^* \\ \mathbf{B}^{-1}(\hat{\gamma}_{2n}^* - \gamma_2^*) \end{pmatrix} \right|\right|. \tag{3.11}$$

As

$$||\hat{\beta}_{1n}^* - \beta_1^*|| = ||\hat{\gamma}_{1n}^* - \gamma_1^*||, \tag{3.12}$$

and

$$||\hat{\beta}_{2n}^* - \beta_2^*|| = p!^{-1} \left|\left| \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ -1 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \dots & -1 & 1 \end{pmatrix} \begin{pmatrix} \hat{\gamma}_p - \gamma_p \\ \vdots \\ \hat{\gamma}_{p+K} - \gamma_{p+K} \end{pmatrix} \right|\right|,$$

we get

$$||\hat{\beta}^*_{2n} - \beta^*_2|| \leq \frac{2}{p!}||\hat{\gamma}^*_{2n} - \gamma^*_2||. \tag{3.13}$$

Combining (3.12) and (3.13), we arrive at the following inequality:

$$||\hat{\beta}_n - \beta|| \leq \begin{cases} 2||\hat{\gamma}_n - \gamma||, & \text{if } p = 1, \\ ||\hat{\gamma}_n - \gamma||, & \text{if } p > 1. \end{cases} \tag{3.14}$$

Since

$$||\hat{\gamma}_n - \gamma|| \xrightarrow{P} 0 \quad as\ n \to \infty,$$

we have

$$||\hat{\beta}_n - \beta|| \xrightarrow{P} 0 \quad as\ n \to \infty.$$

The proof is complete.

Based on (3.14), we arrive at the following theorem, which is obvious from Property 2.

Theorem 2:

$$||\hat{\beta}_n - \beta|| = O_p(\sqrt{p_n/n}/\rho_{n,1})$$

3.3 Simulation Studies

In this section, we shall present two simulation studies to demonstrate the proposed method and compare different estimators. First of all, we state two basic concepts which are useful for simulation studies.

Signal-to-noise ratio (SNR): This concept is used to measure how strong the signal is, compared to the noise. It indicates how difficult the estimation will be. Usually, the smaller the SNR is, the more difficult the estimation is. For the standard nonparametric regression model (3.1), the SNR is defined as $\text{std}(m)/\sigma$, where $\text{std}(m)$ denotes the standard deviation of $m(t)$ when t is regarded as a random variable and σ is the standard deviation of the noise variable ϵ. For computation convenience, we can roughly approximate $\text{std}(m)$ by

$$\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(m(t_i)-\overline{m}(t_i))^2}.$$

In a simulation study, a SNR of $3 \sim 6$ is often used.

Mean of the Squared Errors (MSE): MSE is used to measure how accurate an estimator is. For the standard nonparametric regression model (3.1), the MSE of an estimator $\hat{m}(\cdot)$ may be defined as

$$\frac{1}{n}\sum_{i=1}^{n}\left(\hat{m}(t_i)-m(t_i)\right)^2. \tag{3.15}$$

To compare different estimators, we compare their MSEs. The best is the one with the smallest MSE value.

3.3.1 Simulation 1

In this simulation study, we shall use the famous block test function which was first mentioned in Donoho and Johnstone (1994). It is defined as follows.

$$m(t) = \sum h_j g(t-\tau_j),$$

where $g(t)$ is the kernel function, $g(t) = \{1 + sgn(t)\}/2$. The knots here are

$$(\tau_j) = (0.1, 0.13, 0.15, 0.23, 0.25, 0.40, 0.44, 0.65, 0.76, 0.78, 0.81);$$

and the coefficients are

$$(h_j) = (4, -5, 3, -4, 5, -4.2, 2.1, 4.3, -3.1, 2.1, -4.2).$$

The observed data are then generated by

$$y_i = m(t_i) + \epsilon_i, \;\; \epsilon_i \sim N(0, \sigma^2), \; i = 1, \cdots, n, \tag{3.16}$$

where σ is chosen such that the associated SNR is the given one and $t_i, i = 1, 2, \ldots, n$ are randomly drawn from the standard uniform distribution.

This block test function has been employed extensively in the field of wavelet analysis, e.g., in Donoho and Johnstone (1994) and Antoniadis and Fan (2001). However, as Antoniadis and Fan (2001) pointed out that most wavelet applications to statistics are restricted to the models whose design points are evenly spaced and the sample size is a power of 2. In this simulation study, we attempt to use our proposed method,that is, regression spline smoothing via penalizing derivatives, to deal with the data generated from the simulation model (3.16). Figure 3.2 shows the block test function (upper panel) and a noisy sample generated from it (lower panel). It is seen that the block test function is discontinuous at the knots τ and is constant within each block. Therefore, the block test function is a piecewise constant function, i.e., a constant spline. As a result, the first order derivatives

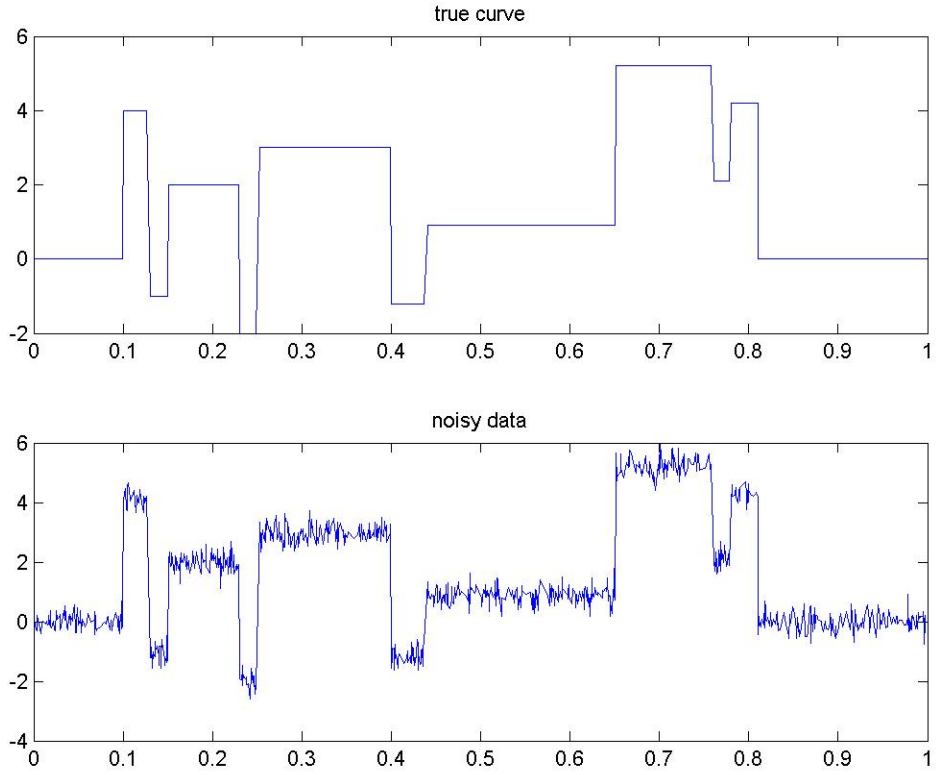

Figure 3.2: *The block test function (upper panel) and a noisy sample (lower panel) generated from the simulation model (3.16) with* $SNR = 6$.

Table 3.1: *The optimal number of knots for different SNRs.*

SNR	Optimal Number of Knots	GCV
3	25	2.4964
4	18	2.4655
5	17	2.1899
6	20	1.9828

will be sparse and hence a linear truncated power basis ($p = 1$) should be used for this simulation study.

Since we set $p = 1$ from an empirical perspective and for simplicity, we choose the uniform knot locating rule (see Subsection 2.2.4) to locate knots, the next step is to determine the number of knots K. In our method, we use the GCV rule to select the optimal number of knots from a reasonable range of K, say, $K \in [15, 50]$. Since different SNR values are considered in our later simulations, we shall select the optimal number of knots for each SNR. This is done by the following procedure: for each SNR, we generate $N = 30$ samples with a sample size of $n = 100$ for each $K \in [15, 50]$ and compute the average values of GCVs for each K. The optimal number of knots is the one leads to the smallest average GCV value. Table 1 shows the results.

As the three tuning parameters which we need to deal with: the order p of the truncated power basis (3.2), the locations of the knots and the number of the knots have been well fixed, we apply our proposed method to the data. Figure 3.3

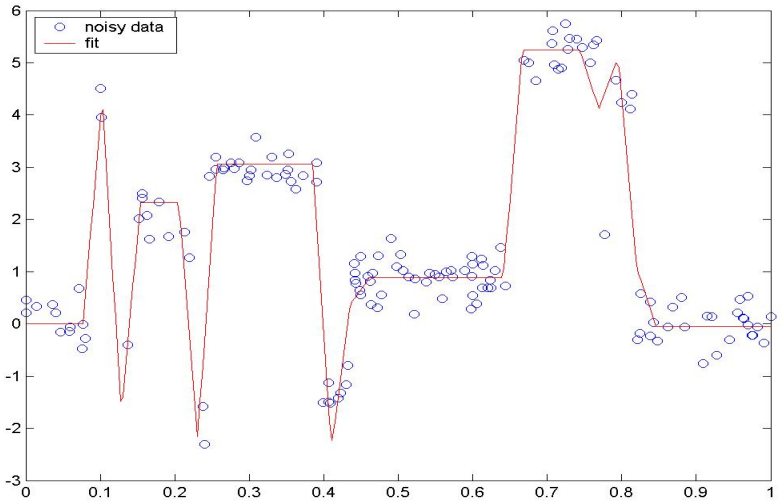

Figure 3.3: *The fit by our proposed method with* $p = 1$ *and* $K = 20$.

displays the fit by applying our proposed method to a noisy sample of size $n = 100$ generated from the simulation model (3.16) with $SNR = 6$. As can be seen from the figure, most blocks are fitted well by our proposed method, except for some small intervals, probably due to the limited data in these small intervals.

Next, we shall compare our proposed method with other methods. We consider four different SNRs: $3, 4, 5$, and 6. For each SNR, we generate $N = 150$ samples $(t_j, y_j), j = 1, 2, \cdots, n$ from the simulation model (3.16) for $n = 100$. For convenience, the design time points $t_j, j = 1, 2, \cdots, n$ are randomly drawn from the standard uniform distribution.

For each sample, three smoothing methods are applied: (a) the forward selection method without penalizing derivatives; (b) the SCAD method without penalizing derivatives; (c) regression spline smoothing via penalizing derivatives (our proposed

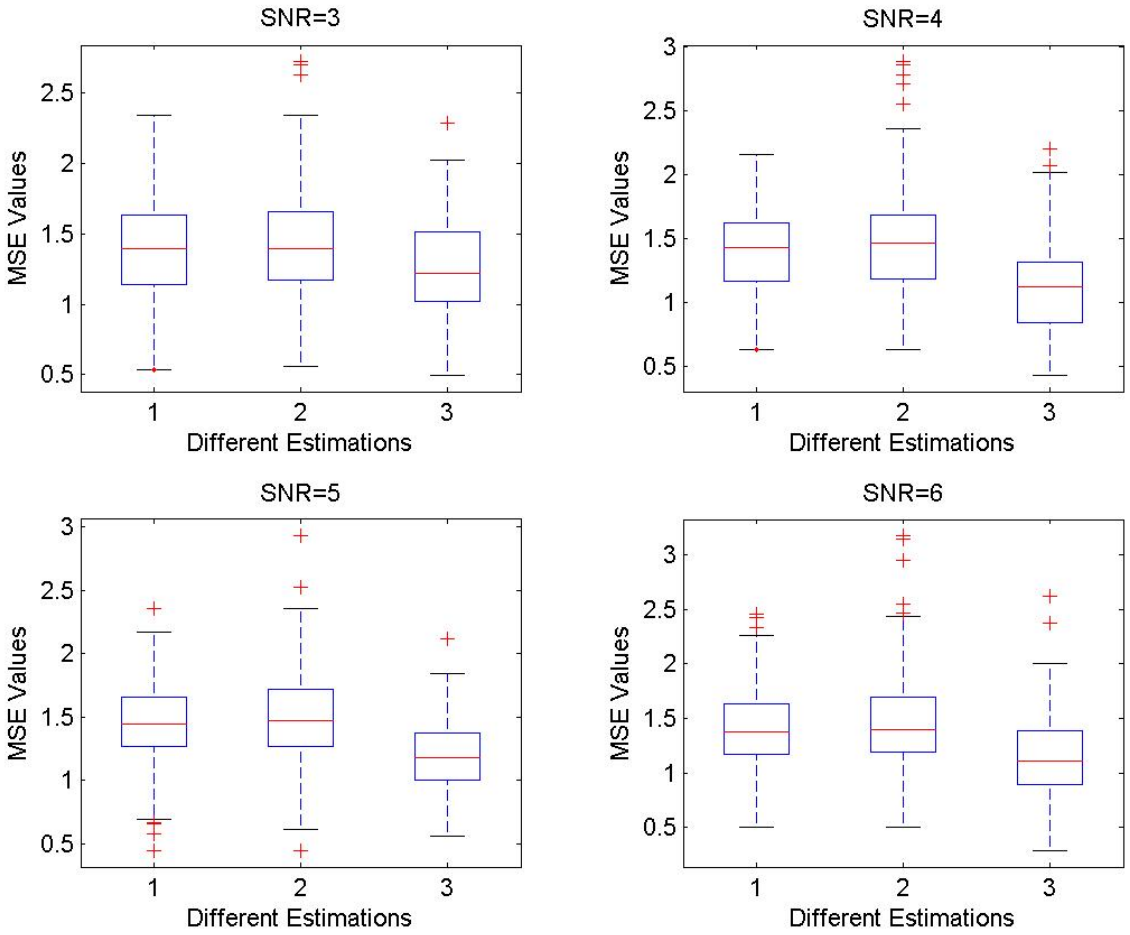

Figure 3.4: *Boxplots of the MSEs for different methods. From left to right: (a) the forward selection method without penalizing derivatives; (b) the SCAD method without penalizing derivatives; (c) regression spline smoothing via penalizing derivatives (our proposed method). Different SNRs were considered for different panels.*

method). We calculate and record the associated MSE for each of the methods. The box plots of MSEs are displayed in Figure 3.4.

By the boxplots, it is seen that among the three methods, our proposed method appears to be the best. This is possibly because it utilizes the information that the first derivatives of the block test function are sparse.

3.3.2 Simulation 2

In the previous simulation study, the underlying function is a constant spline. We now consider a new case where the underlying function is defined by a linear spline:

$$m(t) = \mathbf{\Psi}(t)^T \beta, \tag{3.17}$$

where $\mathbf{\Psi}(t)$ is a linear spline basis vector defined as

$$\Psi(t) = (1, t, (t-\tau_1)_+), (t-\tau_2)_+, \cdots, (t-\tau_7)_+)^T,$$

with the inner knots $\tau_1, \tau_2, \ldots, \tau_7$ evenly scattered in [0,1]. We carefully chose the corresponding coefficient vector as

$$\beta = (2, 0, 2, -2, 2, 2, -4, 2, -2)^T,$$

so that the coefficients of the truncated power basis functions are $(2, -2, 2, 2, -4, 2, -2)^T$, which are not sparse. However, a careful observation of the underlying function $m(t)$ in Figure 3.5 shows that the function has many constant intervals, which means the first derivatives are sparse. In fact, the first derivatives within the dif-

ferent intervals are

$$2, 0, 2, 4, 0, 2, 0$$

respectively. Therefore, if we re-parameterize the underlying function (3.17) in terms of its first derivatives, the associated coefficients are sparse while the original ones are not. A simulated sample can be generated using the simulated model (3.16) with the new underlying function defined in (3.17). The upper panel in Figure 3.5 displays the true underlying function and a noisy sample generated from it with $SNR = 5$. The lower panel is the new regression spline fit by our proposed method with $p = 1$ and $K = 7$. As can be seen, although there exists noise in the data, by penalizing the first derivatives of the regression spline function, we are able to produce a fit which is almost the same as the true underlying function.

Next, we shall compare our proposed method with other estimation methods. The data generation process for the comparison in different methods is the same as in simulation 1. Since we fix $K = 7$ and $p = 1$ in our model, the number of the coefficients to be estimated is only 9, which is quite small. Therefore, we can apply the ordinary least square method directly to the original model (3.9) or the transformed model (3.10) to estimate. Actually, applying OLS to (3.9) and (3.10) will get the same estimation results of β.

Figure 3.6 displays the boxplots of MSEs for the OLS method and our proposed method with varying SNRs. It is seen that, in terms of the mean values of MSEs, our method outperforms the OLS method, probably because our proposed method

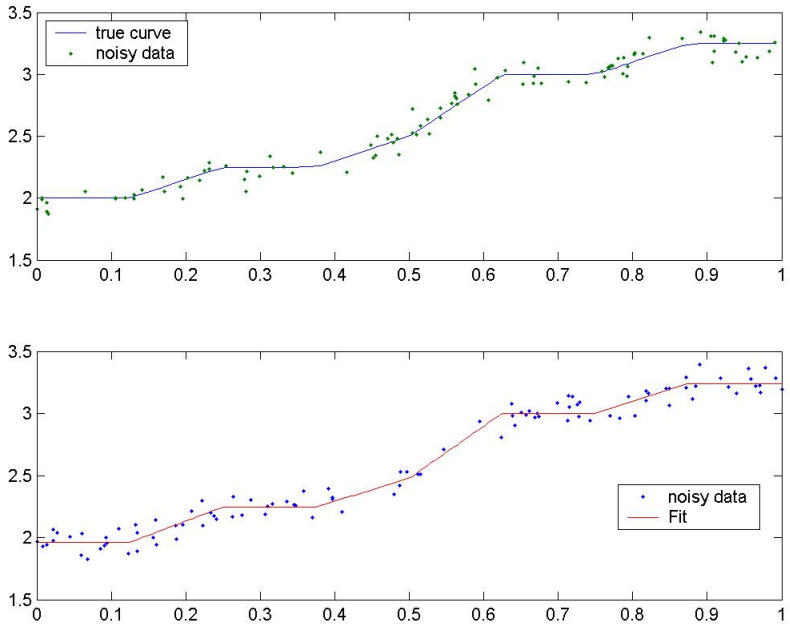

Figure 3.5: *The underlying function (The upper panel) and the regression spline fit by our proposed method with* $p = 1$ *and* $K = 7$ *(The lower panel).*

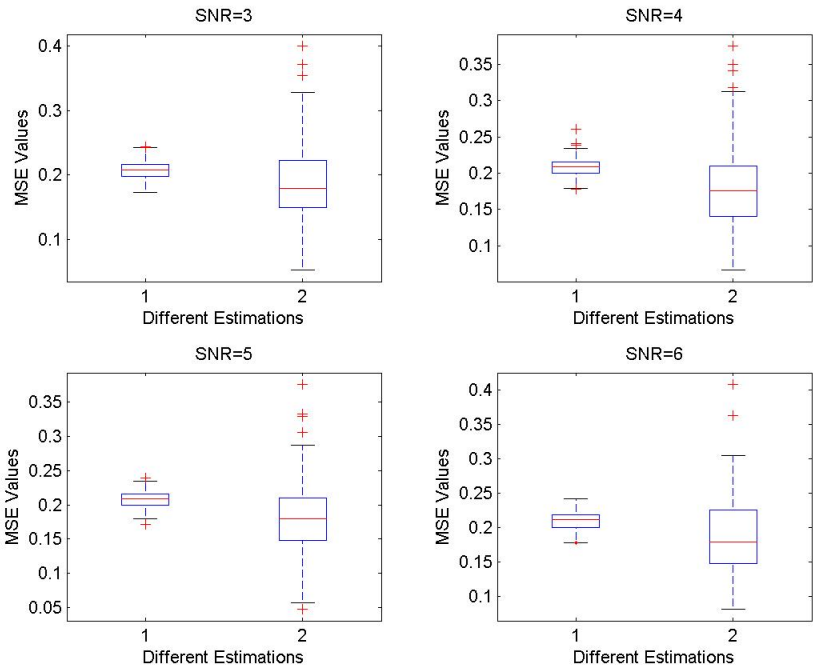

Figure 3.6: *Boxplots of MSEs for the OLS method (left) and our proposed method (right).*

smoothes the noisy data by penalizing the first derivative of the regression spline function. However, it is seen that the OLS method leads to a smaller variance of MSEs. It is because the OLS method is more stable than other methods when the dimension of the estimator is small.

3.4 Real Data Analysis

In this section, two real data applications are presented to illustrate our proposed method. The first data set investigated is the motorcycle data set. It has been introduced briefly in Chapter 1 which motivates the methodology of this thesis. The second data set is the fuel consumption data. Both data sets are widely used

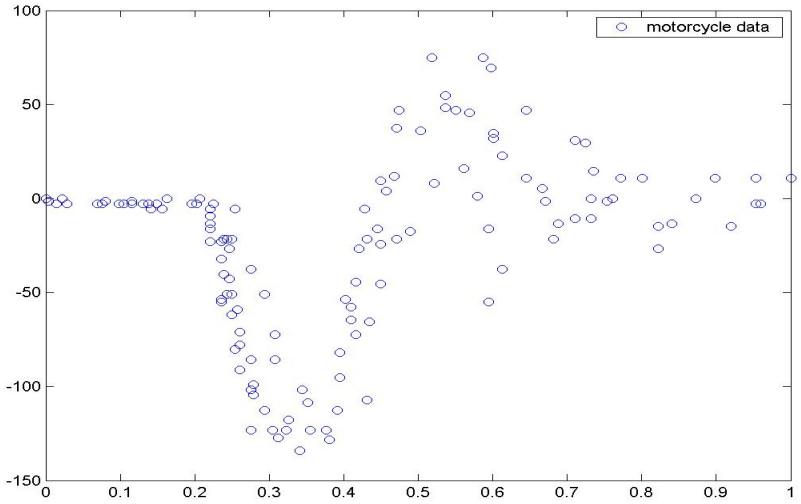

Figure 3.7: *The motorcycle data.*

to illustrate different methods in statistics.

3.4.1 The Motorcycle Data

The motorcycle data set was first studied by Silverman (1985). The data set has 133 observations showing the effects of motorcycle crashes on victims' heads. The dependant variable is the time after a simulated impact with motorcycles and the response variable is the head acceleration of a PTMO (post mortem human test object), which captures the crash effects. The data set is displayed in Figure 3.7. For easy presentation, we re-scaled the design points $t_i^{'}s$ to $[0, 1]$.

From Figure 3.7, it is seen that different polynomials may be fit to the data within different intervals. In fact, within the interval, $[0, 0.2]$, a constant line may fit the data well; within the interval $(0.2, 0.5]$, another quadratic polynomial may

fit the data well; within the interval $(0.5, 1]$, the data are more scattered, thus the underlying pattern is not clear.

Thus, a simple polynomial model is insufficient to fit the data. In the literature, several nonparametric methods have been applied to the motorcycle data. For example, Silverman (1995) adopted the spline smoothing techniques. Hall and Turlach (1997) used the classical wavelet thresholding method. Kovac and Silverman (1999) improved the wavelet thresholding method by filtering the outliers, etc. One major problem of these methods is that, the data in the left range of the support cannot be fitted sufficiently well.

Notice that the second derivative of a quadratic polynomial is constant , while the second derivative of a linear or constant polynomial equals zero. Therefore, if we use a quadratic truncated power basis to fit the data, the second derivatives of the fitted regression spline should be sparse. Meanwhile, due to the feature of the motorcycle data, if we use a cubic truncated power basis, the third derivatives of the constant, linear and quadratic parts of the fitted regression spline are all zero. Therefore, we shall try the quadratic and the cubic truncated power basis to fit the data using our proposed method respectively, and examine which one is better.

We employ the method discussed in Subsection 3.3.2 to choose the two tuning parameters: p and K. First, we compute the best GCV (BIC) scores via finding the best number of knots for $p = 2$ and $p = 3$, respectively. Then, we compare the two associated GCV (BIC) scores and choose the optimal p which leads to the

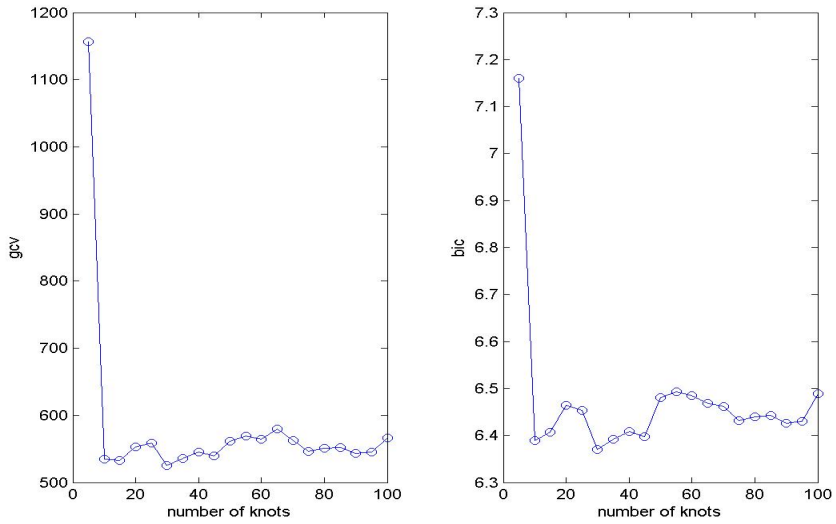

Figure 3.8: *GCV and BIC curves against various number of initial knots for the proposed method using a quadratic truncated power basis.*

smaller GCV (BIC).

Figure 3.8 and 3.9 show the plots of GCV and BIC values when $p = 2$ and 3, respectively. Notice the number of the initial knots are chosen as $5, 10, \ldots, 100$.

It is seen that, when $p = 2$, the optimal number of knots is 30, with the associated $GCV = 525.1421$ and $BIC = 6.3709$. When $p = 3$, the optimal number of knots is 45, with the associated $GCV = 538.8862$ and $BIC = 6.4388$. A simple comparison indicates that a quadratic truncated power basis with 30 knots is the best choice. Figure 3.10 displays the fit by our proposed method using a quadratic truncated power basis and a group of $K = 30$ knots which are equally spaced in $[0, 1]$. It is seen that the data are well fit, even within the left end of the support.

Notice that different methods may be applied to fit the transformed model (3.10) for the motorcycle data, including (a) the OLS method, (b) the forward selection

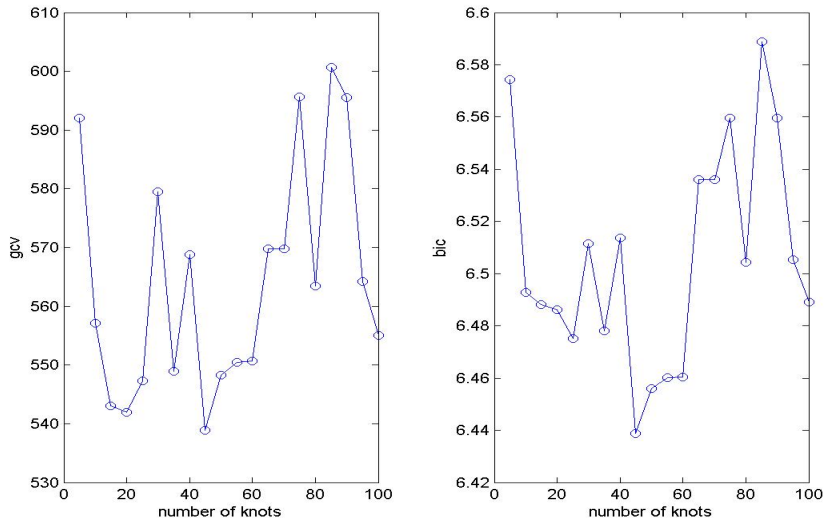

Figure 3.9: *GCV and BIC curves against various number of initial knots for the proposed method using a cubic truncated power basis.*

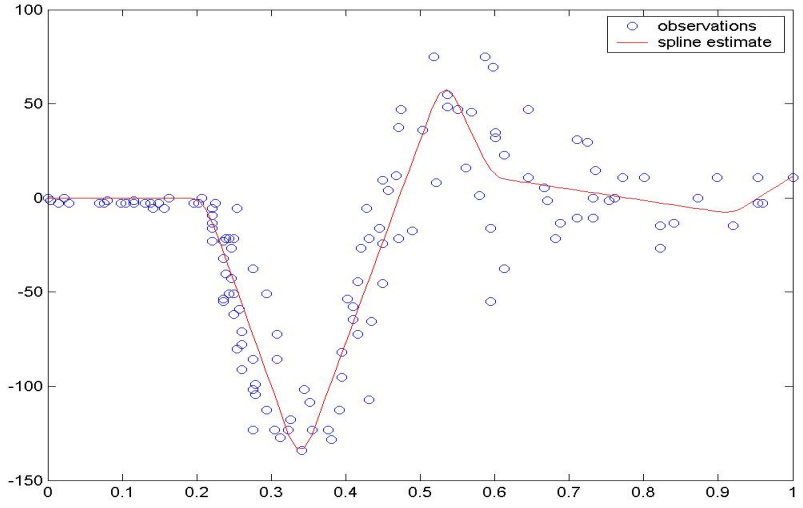

Figure 3.10: *The new regression spline fit to the motorcycle data by the proposed method using a quadratic truncated power basis with the number of knots,* $K = 30$.

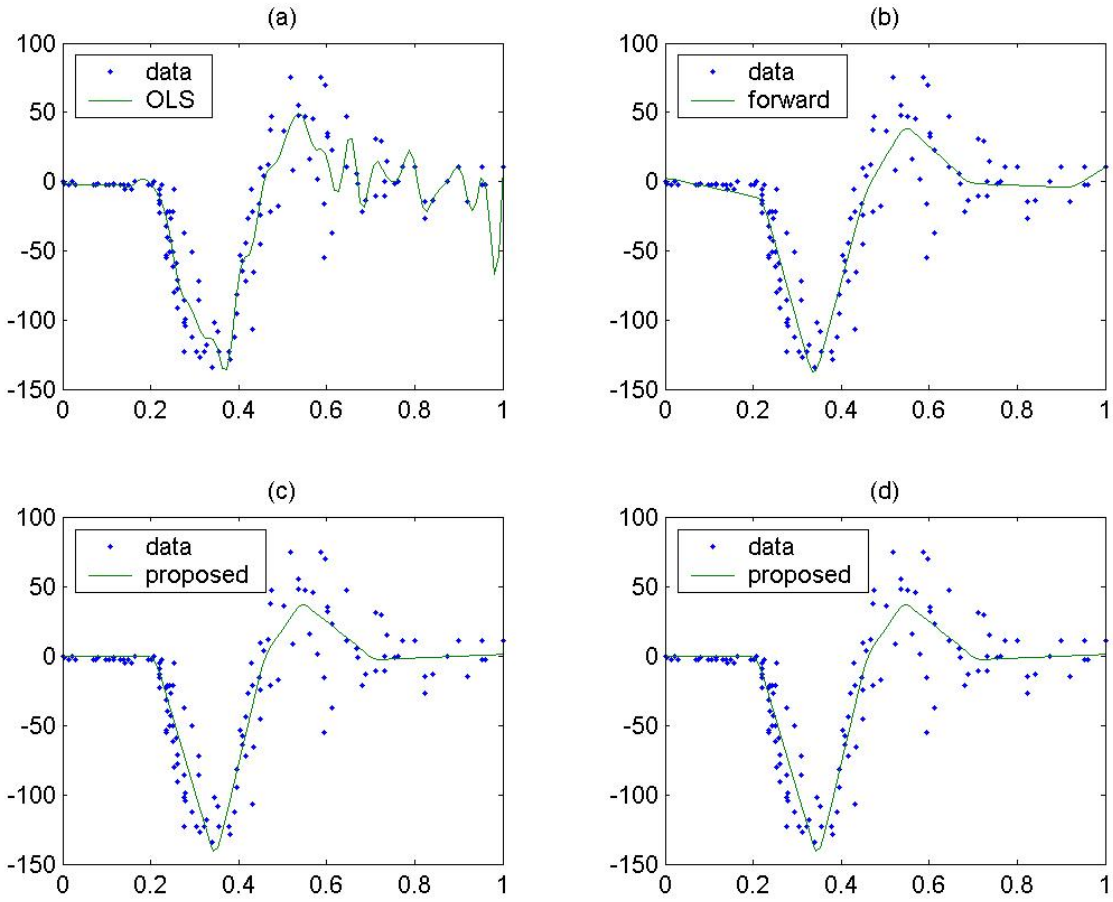

Figure 3.11: *Various fits to the motorcycle data by applying different estimation methods to the transformed model (3.10). A quadratic truncated power basis with* $K = 35$ *initial knots is used for all the methods. The knots are evenly spaced in [0,1].*

based method, and (c) the SCAD method. The resulting fits are displayed in Figure 3.11 $(a) - (c)$. As we can see, the OLS fit has too many wiggles in the right region; the forward selection shows a small wiggle at the left end; while the SCAD fit seems to perform well. This shows the advantage of SCAD method over the OLS and forward selection based methods.

What would happen if we apply the estimation methods to the original model

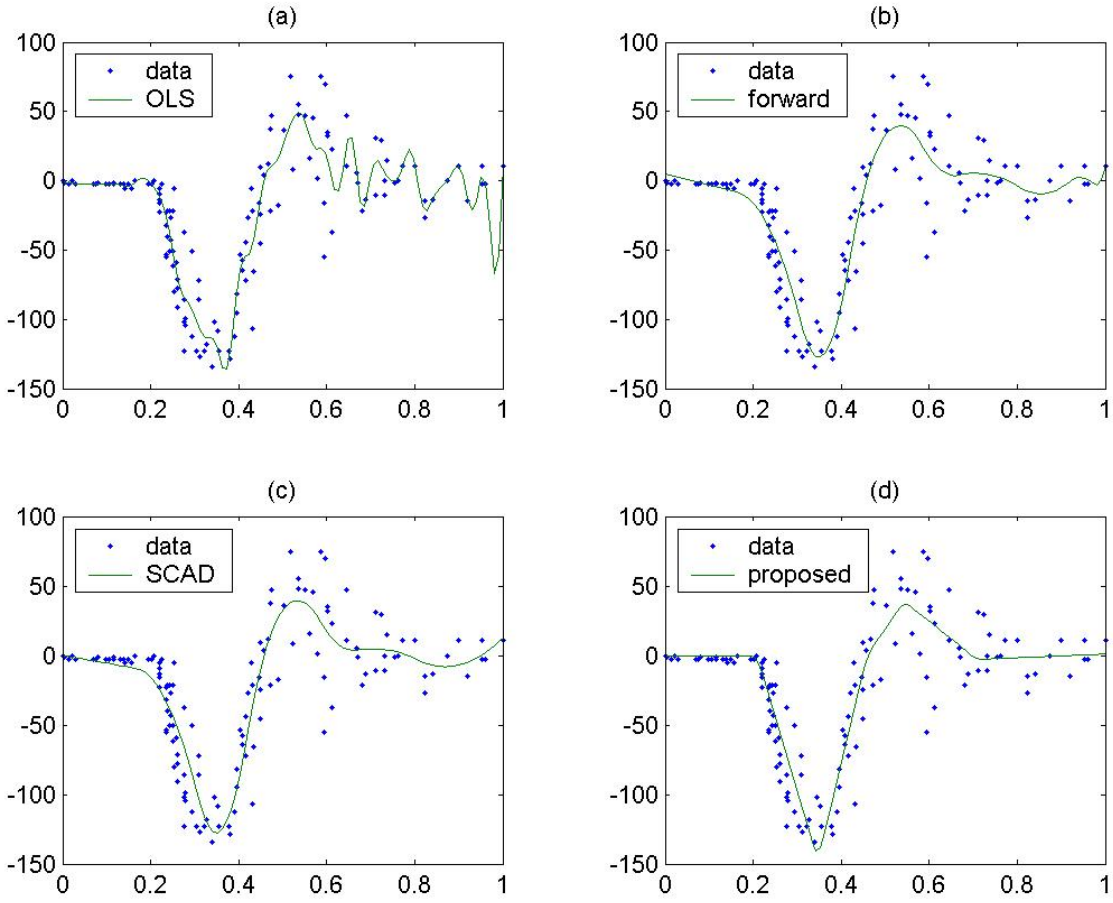

Figure 3.12: *Various fits to the motorcycle data by applying different estimation methods to the original model (3.9).*

(3.9) directly? The associated fits are presented in Figure 3.11 $(a) - (c)$, all of which are not satisfactory enough: the OLS fit is too wiggly in the right region; the forward selection fit performs poorly on the left end region with a decreasing curve; the SCAD fit improves the forward selection fit, but still does not fit the left region well. Therefore, we may conclude that by penalizing the derivative of $m(t)$, we improve the fit to the motorcycle data.

3.4.2 The Fuel Consumption Data

In this subsection, we shall apply our proposed method to the fuel consumption data. The data set records the fuel usage for vehicles of different weights. The response variable is the measurement of fuel usage (in miles per gallon) and the predictor variable is the weight (in pounds) of the vehicle. The data are displayed in Figure 3.13 as dots.

From the scatter plot, one may guess that a single quadratic or cubic polynomial model may fit the data well. However, we employ the truncated power basis model to fit the data here. It makes sense as the regression spline is a natural generalization of polynomial functions. Since the second or third derivatives of the underlying regression function may be sparse, and the third derivatives of the underlying regression spline function may be more sparse. Thus, we here use a cubic truncated power basis to fit the data.

To select the tuning parameter of K, i.e., the number of knots, we compute

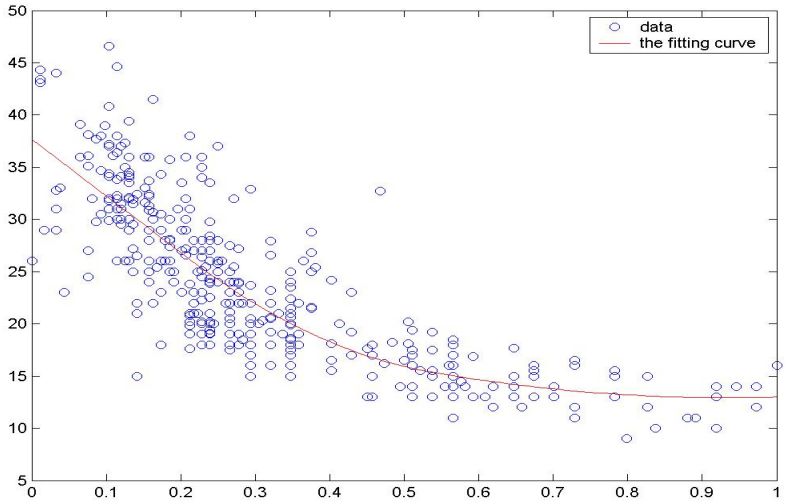

Figure 3.13: *The fuel consumption data (dots) and the new regression spline fit by the proposed method using a cubic truncated power basis with* $K = 35$ *knots.*

the GCV scores against the number of knots. Similar to the motorcycle data, we only consider K as $5, 10, \ldots, 100$. Figure 3.14 shows the GCV curve against the number of initial knots. It seems that the size of knots has little influence on the GCV scores, which lie between 19.14 and 19.26. But we can still conclude from the figure that a group of 35 initial knots is the optimal choice to fit the model.

Figure 3.13 displays the new regression spline fit by the proposed method using a cubic truncated power basis with $K = 35$ knots. It is seen that the data are well fitted.

Similar to the motorcycle data, we can fit the fuel consumption data by the forward selection method or the SCAD method directly applied to the original model (3.9). We found that the fits are similar to the fit produced by the SCAD method applied to the transformed model (3.10) (See Figure 3.15). The result is

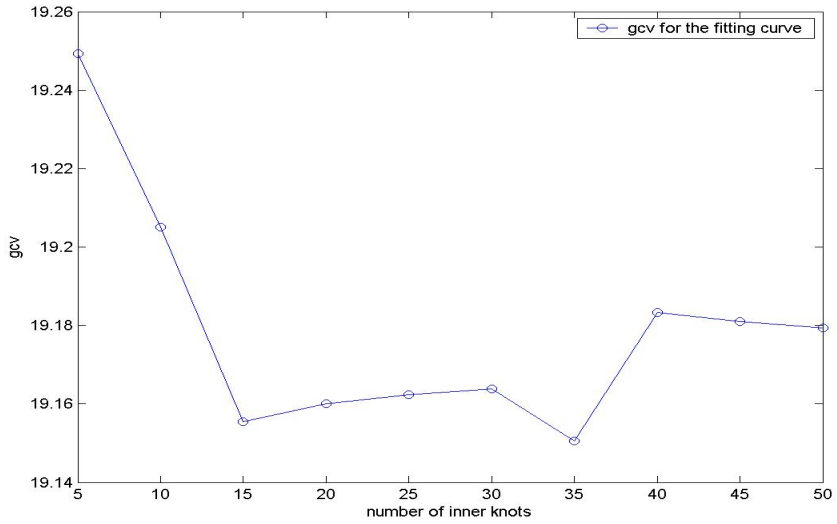

Figure 3.14: *GCV curve against the number of initial knots when the proposed method using a cubic truncated power basis is applied to fit the fuel consumption data.*

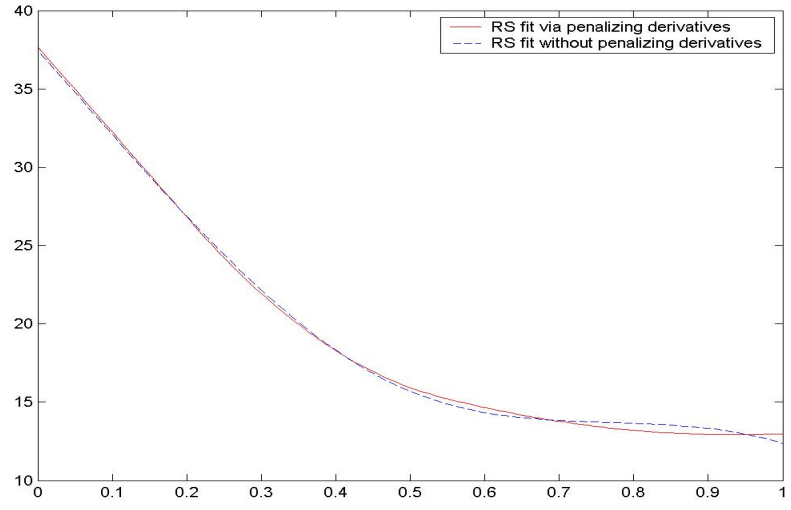

Figure 3.15: *The new regression spline fit by our proposed method (solid curve) and the usual regression spline fit (dashed curve).*

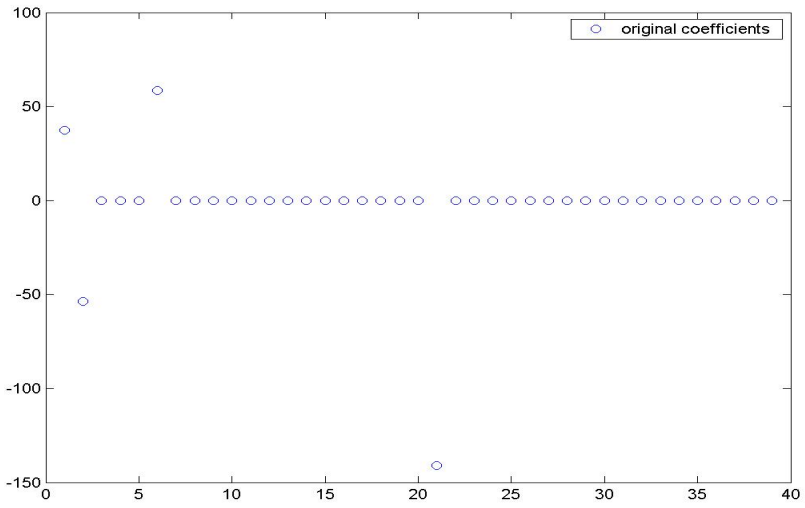

Figure 3.16: *The SCAD estimator of the original coefficient vector which is sparse enough.*

not surprise and it can be explained as follows. As we mentioned previously based on Figure 3.13, the fuel consumption data can be fitted well by a single quadratic or cubic polynomial without any knots. Therefore, when we use a cubic TPB model, the original coefficient vector is sparse enough even without considering the second or third derivatives of the regression spline function. Figure 3.16 displays the SCAD estimator of the original coefficients without penalizing the derivatives. It can be seen that only 4 out of 39 coefficients are non-zero components. Therefore, the original coefficients are very sparse and thus our proposed method performs similarly as the traditional methods.

3.5 Conclusion and Discussion

In this thesis, we proposed a new regression spline smoothing method to fit the standard regression model (3.3), which we call "regression spline smoothing via penalizing derivatives". The method is based on the classical regression spline model with a pth order truncated power basis. When the regression spline coefficients may not be sparse but the pth times derivatives of the regression spline function are sparse, our proposed method performs better than the usual regression spline smoothing methods. The key idea is to re-parameterize the original coefficient vector into a new vector, whose last $K + 1$ terms are the pth times derivatives of the regression spline function. Under the assumption that the pth times derivatives of the function are zero for most design time points, we then apply the SCAD method of Fan and Li (2001) to fit the transformed model. It is equivalent to attaching a smoothly clipped absolute deviation penalty to the pth times derivatives of the regression spline function. Methods for selecting proper tuning parameters are presented in this thesis. Simulation studies are conducted to demonstrate the good performance of the proposed method. Two real data applications are used to illustrate the proposed method and compare with other estimation methods for regression spline models. It has been shown that the newly proposed method is more accurate than the usual regression spline methods when the true curve is piecewise with different orders of polynomials at different segments.

Further work needs to be done to extend the proposed method to other non-

parametric models, as well as semiparametric models (Cox 1972, Bickel 1982) and varying coefficients models (Hastie and Tibshirani 1993). Moreover, theories should be developed to investigate more asymptotic properties of the proposed method.

Some previous work in this area includes Tibshirani (2005), attaching a L_1 Lasso type penalty to both the coefficients and the first times derivatives; James and Zhu (2007) imposes sparsity on the dth times derivatives of the coefficient function of the functional linear regression models, and develops a new model called FLiRTI model. Our regression spline smoothing method via penalizing derivatives is similar to them, but attaches a SCAD penalty to the pth times derivatives of the regression spline function.

Bibliography

[1] Akaike, H. (1974). A new look at the statistical model identification, *IEEE Transactions On Automatic Control*, **AC-19**, 716-723.

[2] Antoniadis, A. and Fan, J. (2001). Regularization of wavelets approximations. *J. Amer. Statist. Assoc.*, **96**, 939-967.

[3] Breiman, L. (1995). Better subset regression using the nonnegative garrote. *Technometrics*, **37**, 373-384.

[4] Craven, P. and Wahba, G. (1979). Smoothing noisy data with spline functions. *Numer. Math.*, **31**, 377-403.

[5] Donoho, D.L. and Johnstone, I.M. (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrics*, **81**, 425-455.

[6] Eubank, R.L. (1988). *Spline Smoothing and Nonparametric Regression*, Dekker, New York.

[7] Fan, J. and Gijbels, I. (1996). *Local Polynomial Modeling and its Applications*, Chapman and Hall, London.

[8] Fan, J. and Li, R. (2001). Variable selection via nonconcave penalised likelihood and its oracle properties. *Ann. Statist.*, **96**, 1348-60.

[9] Friedman, J.H., and Silverman, B.W. (1989). Flexible parsimonious smoothing and additive modeling. *Technometrics*, **31**, 3-39.

[10] Gasser, Th. and Muller, H.G.(1984). Estimating regression functions and their derivatives by the kernel method. *Scandinavian J. Statistics*, **11**, 171-184.

[11] Green, P. and Silverman, B.W. (1994). *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach*, Chapman and Hall, London.

[12] Hastie, T. and Tibshirani, R. (1993). Varying-Coefficient Models. *J. Royal Statist. Soc.*, **B**, **55(4)**, 757-796.

[13] Huang, J. and Xie, H. (2007). Asymptotic oracle properties of SCAD-penalized least squares estimators. *IMS Lecture Notes*, **55**, 149-166.

[14] Hunter, D.R. and Li, R. (2005). Variable selection using MM algorithms. *Ann. Statist.*, **33**, 1617-1642.

[15] Fu, J. (1998). Penalized regressions: the bridge versus the lasso. *J.Statist. Comput. Simul.*, **72(8)**, 647-663.

[16] Jeffrey, S. Simonoff, (1996). *Smoothing Methods in Statistics*, Springer, New York.

[17] Lee, T.C.M. (2000). Regression spline smoothing using the minimum description length principle. *Statist. & Prob. Letters.*, **48**, 71-82.

[18] Michael G.Schimek (2000). *Smoothing and Regression*, Jonh Wiley and Sons.

[19] Nadaraya, E. A. (1964). On estimating regression. *Theory of Probability and its Applications*, **9 (1)**, 141-142.

[20] Rosset, S., Zhu, J. and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *J.R.Statist.SOC.B.*, **67**, 91-108.

[21] Ruppert, D. and Carroll, R.J. (1997). Penalized regression splines, Technical Report, Department of Operations Research and Industrial Engineering, Cornell University, USA.

[22] Silverman, B.W. (1985). Some aspects of the spline smoothing approach to non-parametric regression curve fitting (with discussion). *J. Royal Statist. Soc.*, **B**, **47**, 1-52.

[23] Schwarz, G. (1978). Estimating the dimension of a model. *Ann. Statist.*, **6**, 461-464.

[24] Tibshirani, R. (1996). Regression shrinkage and selection via the LASSO. *J. Royal Statist. Soc.*, **B**, **58**, 267-288.

[25] Tibshirani, R., Saunders, M., Rosset, S., and Zhu, J. (2005). Sparsity and smoothness via the fused lasso. *J. Royal Statist. Soc.*, **B**, **67(1)**, 91-108.

[26] Wahba, G. (1990). *Spline Models for Observational Data*. SIAM, Philadelphia, PA.

[27] Wang, H., Li, R. and TSAI, C. (2007). Tuning parameter selectors for the smoothly clipped absolute deviation method. *Biometrika*, **94**, 553-568.

[28] Wand, M.P. (2000). A comparison of regression spline smoothing procedures. *Compuational Statistics*, **15**, 443-462.

[29] Watson, G.S. (1964). Smooth regression analysis. *Sankhya - The Indian Journal of Statistics*, **26**, 359-372.

[30] Wu, H. and Zhang, J-T. (2006). *Nonparametric Regression Methods for Longitudinal Data Analysis*. Wiley Series in Probability and Statistics.