Estimation of Higher-order Two-phase Regression Models

Hyunju Son

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington

2020

Committee:

Youyi Fong, Chair

Wei Sun

Program Authorized to Offer Degree:
Biostatistics

University of Washington

Abstract

Estimation of Higher-order Two-phase Regression Models

Hyunju Son

Chair of the Supervisory Committee:
Youyi Fong
Department of Biostatistics

Two-phase regression models are a class of nonlinear regression models that are known for their flexibility and interpretability. An important feature of two-phase regression models is the existence of a threshold at which the relationship between an outcome and a covariate of interest changes. A standard estimation method, such as that used for generalized linear models, cannot be applied to two-phase regression models since the likelihood function is not differentiable with respect to the threshold parameter. We resolve this difficulty by using a grid search method which reduces the problem to a set of well-behaved likelihood functions for given candidate threshold values. Previously, a fast grid search algorithm that dramatically improved computational efficiency over a brute-force grid search was developed for two-phase regression models with linear trends. Here we generalize this algorithm to higher-order two-phase regression models where two separate polynomial regressions, not limited to linear, are used to model each phase (i.e., before and after the threshold). Based on the proposed fast grid search algorithm, we perform Monte Carlo simulations to examine the behavior of the parameter estimates. A real data example is also presented to illustrate the practical use of two-phase regression models.

TABLE OF CONTENTS

Page

List of Figures ii

List of Tables iii

Chapter 1: Introduction 1
1.1 Background 1
1.2 Objectives 2

Chapter 2: Fast grid search algorithms for higher-order two-phase regression models . . . 4
2.1 Algorithms for q-th order two-phase regression models 4
2.2 Discussion of special cases 12

Chapter 3: Simulation studies 21
3.1 Bias and coverage when models are correctly specified 21
3.2 Bias and coverage when models are misspecified 32

Chapter 4: Application to LIDAR data 35

Chapter 5: Discussion 38

LIST OF FIGURES

Figure Number Page

1.1 Mean functions of thirteen two-phase regression models to be discussed later. 3

3.1 Two simulated datasets to evaluate the performance of the estimators under model misspecification. 33

4.1 The LIDAR data. 36

LIST OF TABLES

Table Number Page

3.1 Results from 10,000 MC simulations for M20 and M02. 25
3.2 Results from 10,000 MC simulations for M21 and M12. 26
3.3 Results from 10,000 MC simulations for M21c and M12c. 27
3.4 Results from 10,000 MC simulations for M22 and M22c. 28
3.5 Results from 10,000 MC simulations for M30 and M03. 29
3.6 Results from 10,000 MC simulations for M31 and M13. 30
3.7 Results from 10,000 MC simulations for M33c. 31
3.8 Results from 10,000 MC simulations for M20 and M22 under model misspecification. 34

4.1 Estimation results from the six two-phase regression models for the LIDAR data. . . 37

ACKNOWLEDGMENTS

First, I would like to thank my thesis advisor, Dr. Youyi Fong, for his guidance and continuous support throughout the M.S. program. Without his advice, I would not have been able to complete this thesis. I would also like to thank my committee member, Dr. Wei Sun, for his valuable feedback on this thesis.

I thank my friend, Youngeun, who has been motivating me and encouraging me to keep going. Last but not least, I would like to thank my mom for everything she has done for me and being my biggest supporter.

DEDICATION

to my mom

Chapter 1

INTRODUCTION

1.1 Background

Two-phase regression models are used extensively to model a nonlinear relationship between an outcome and a covariate of interest (e.g., Hinkley, 1971; Gallant and Fuller, 1973). These models describe the nonlinear nature of the relationship more flexibly, by fitting two different polynomial regression functions in the domain of the covariate partitioned by a threshold. Such flexibility and ease of interpretation make them attractive for a wide range of applications such as biology, ecology, economics and physics. In the case of biomedical applications, for example, they can model the relationship between the reciprocal of serum-creatinine and time following a renal transplant (Smith and Cook, 1980). A positive relationship between the two shortly after transplant indicates restored renal function but the change to a negative relationship indicates rejection. Thus, estimating the time at which rejection occurs (i.e., threshold) is of paramount interest for monitoring the renal function of recipients.

The primary focus in this thesis is on continuous two-phase regression models (hereafter referred to as two-phase regression models for brevity) where, at the threshold, the two polynomial regression functions join and no jumps occur. As in the usual regression setting, parameters of two-phase regression models are estimated using the maximum likelihood (ML) method. However, the fact that the likelihood function of a two-phase regression model is non-smooth and non-convex with respect to the threshold parameter poses severe challenges for estimation using the ML method.

One approach to circumvent this problem and find the maximum likelihood estimator (MLE) of the two-phase regression model is smooth approximation (Pastor-Barriuso et al., 2003). The likelihood function is approximated by differentiable transition functions (e.g., logistic function) and the MLE is obtained by an iterative procedure such as Gaussian-Newton algorithms. Although it achieves a fairly fast convergence (Fong et al., 2017a), the solution produced is a local maximizer of the likelihood function and starting values play a crucial role in the local convergence. Fong

(2019) also showed that coverage probabilities of bootstrap confidence intervals obtained from the smooth approximation are far below the nominal confidence level, particularly for small sample sizes.

Grid search is an alternative approach that aims to simplify the problem by conditioning on fixed threshold values. In essence, this approach chooses a grid of candidate threshold values (usually the observed covariate values discarding extreme ones) within the domain of the covariate. Conditioned on any candidate threshold value, the two-phase regression model then can be regarded as the standard regression model in which the likelihood function is differentiable. The MLE of the threshold is the candidate value which gives the largest likelihood. Despite its simplicity and intuitive appeal, however, the grid search method is exhaustive in the sense that it computes the likelihood for every candidate value, making it nearly impractical for large sample sizes if done in a brute force fashion.

Recently, Fong (2019) built a fast grid search algorithm for piecewise linear two-phase regression models (i.e., two lines join at the threshold) to shorten the computation time. Without refitting the submodel for each candidate threshold, recursive formulas in the algorithm compute and update the likelihood of the submodel sequentially by considering two neighboring candidate threshold values. This algorithm was further improved by Elder and Fong (2019), who adopted the inverse formula of a block matrix to explicitly calculate the hat matrix. As a result, computational time was significantly reduced even when the sample size and the number of covariates were both large. The algorithm was implemented only for linear two-phase regression models, still leaving room for further development.

1.2 Objectives

This thesis extends the fast grid search algorithm developed by Elder and Fong (2019) to the general case of two-phase polynomial regression models. All possible polynomials of order q ($q = 0, 1, 2, \dots$), not just line segments, are considered both before and after the threshold. Throughout, for readability, we use the notation "Mqq" and "Mqqc" to denote the model. The first and second numbers following M represent the orders of polynomials before and after the threshold, respectively. For example, M11 denotes a linear two-phase regression model and M01 denotes the hinge model (Fong et al., 2017a). The letter "c" refers to smoothness constraints at the threshold imposed on the

polynomials of order less than q. M12c, for example, denotes a linear-quadratic model constrained to have the same first derivative at the threshold.

We first derive a fast grid search algorithm for the general q-th order two-phase regression model ($q \geq 2$) in Chapter 2. Specifically, six types of q-th order model, i.e., Mq0, M0q, Mq1, Mq1, Mqq and Mqqc, are studied in this thesis rather than exhaustively exploring all possible models. Then we discuss special cases of the q-th order model as shown in Figure 1.1. These models involve quadratic or cubic polynomials and are most likely to be considered when confronted with real data problems. In Chapter 3, the performance of our proposed algorithms is assessed via Monte Carlo simulations. Chapter 4 presents a real data application to demonstrate the use of two-phase regression models. A concluding discussion follows in Chapter 5.

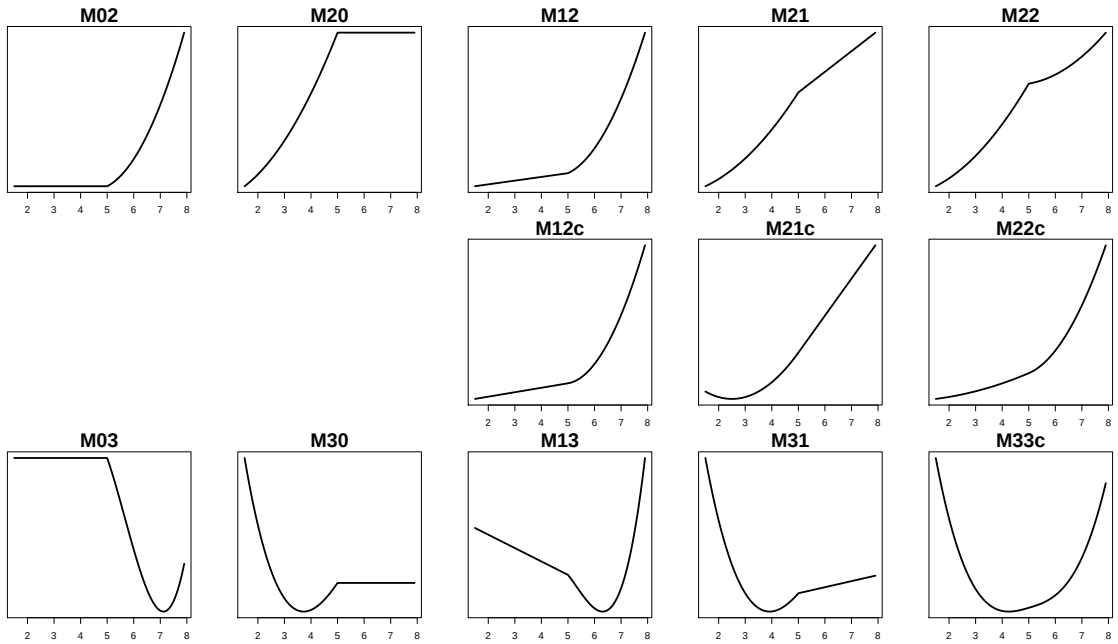

Figure 1.1: Mean functions of thirteen two-phase regression models to be discussed later. The threshold is at $x = 5$ for all models.

Chapter 2

FAST GRID SEARCH ALGORITHMS FOR HIGHER-ORDER TWO-PHASE REGRESSION MODELS

2.1 Algorithms for q-th order two-phase regression models

2.1.1 Fast grid search algorithms for Mq0 and M0q

We start with the model Mq0, which is a generalized version of the upper hinge regression model (corresponding to M10 in this thesis) studied in Elder and Fong (2019). Mq0 can be expressed in terms of the following mean function:

$$\eta(x, \boldsymbol{z}) = \alpha_1 + \boldsymbol{\alpha_2}^T \boldsymbol{z} + \beta_1 (x - e)_- + \beta_2 (x - e)_-^2 + \cdots + \beta_q (x - e)_-^q, \tag{2.1}$$

where x is the covariate of interest, e is the threshold parameter, $\boldsymbol{z}$ is a vector of adjustment covariates of dimension $p - q - 1$ (p is the number of coefficients), and $(x - e)_- = x - e$ if $x < e$ and 0 otherwise. Thus, with the corresponding regression coefficients, the parameter vector is defined as $\boldsymbol{\theta} \equiv (\alpha_1, \boldsymbol{\alpha_2}, \beta_1, \ldots, \beta_q, e)$. As in a classical linear model setting, let n be the sample size, $\boldsymbol{Y} = (y_1, \ldots, y_n)$ be the outcome vector, $\boldsymbol{x} = (x_1, \ldots, x_n)$ be the main covariate vector, $\boldsymbol{Z} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n]$ be the n by $p-q-1$ adjusted covariate matrix, and $\mathbf{1}$ be a vector of ones of length n. Then $\boldsymbol{X}_e \equiv [\mathbf{1}, \boldsymbol{Z}, \boldsymbol{v}_e, \boldsymbol{v}_e^{\circ 2}, \ldots, \boldsymbol{v}_e^{\circ q}]$, where $\boldsymbol{v}_e \equiv (\boldsymbol{x} - \mathbf{1}e)_-$, denotes the n by p design matrix of Mq0 for a fixed value of e. Here, for any vectors $\boldsymbol{a} = (a_1, \ldots, a_n)$ and $\boldsymbol{b} = (b_1, \ldots, b_n)$, we denote them by $\boldsymbol{b} = \boldsymbol{a}^{\circ q}$ if $b_i = a_i^q$ for $i = 1, \ldots, n$.

Conditioned on a given e, Mq0 is reduced to the standard linear model whose log likelihood is proportional to $-\boldsymbol{Y}^T(\boldsymbol{I} - \boldsymbol{H}_e)\boldsymbol{Y} = -\boldsymbol{Y}^T\boldsymbol{Y} + \boldsymbol{Y}^T\boldsymbol{H}_e\boldsymbol{Y}$, where $\boldsymbol{I}$ is the identity matrix and $\boldsymbol{H}_e = \boldsymbol{X}_e(\boldsymbol{X}_e^T\boldsymbol{X}_e)^{-1}\boldsymbol{X}_e^T$ is the hat matrix. The maximum likelihood estimator (MLE) of Mq0 for a given value of e is $\hat{\boldsymbol{\theta}}_e = (\boldsymbol{X}_e^T\boldsymbol{X}_e)^{-1}\boldsymbol{X}_e^T\boldsymbol{Y}$. To find the MLE, the grid search algorithm takes the observed values of $\boldsymbol{x}$ as the candidate threshold values (after discarding extreme values, e.g., 10%) and computes $\boldsymbol{Y}^T\boldsymbol{H}_e\boldsymbol{Y}$ for the submodel corresponding to each candidate e. The MLE $\hat{\boldsymbol{\theta}}_e$ of the

submodel giving the largest log likelihood, along with the corresponding e, is taken as the MLE of Mq0.

Our proposed grid search algorithm aims to enhance the search efficiency by updating $\boldsymbol{Y}^T\boldsymbol{H}_e\boldsymbol{Y}$ sequentially instead of computing it for each e separately. Decomposing the design matrix of the submodel into two parts such that only one of them involves e is the essential step for speeding up computation:

$$\boldsymbol{X}_e \equiv [\mathbf{1}, \boldsymbol{Z}, \boldsymbol{v}_e, \boldsymbol{v}_e^{\circ 2}, \ldots, \boldsymbol{v}_e^{\circ q}] \equiv [\boldsymbol{X}, \boldsymbol{V}_e],$$

where $\boldsymbol{X} \equiv [\mathbf{1}, \boldsymbol{Z}]$ and $\boldsymbol{V}_e \equiv [\boldsymbol{v}_e, \boldsymbol{v}_e^{\circ 2}, \ldots, \boldsymbol{v}_e^{\circ q}]$. Then the submodel hat matrix can be explicitly computed by using the block matrix inversion formula as follows:

$$(\boldsymbol{X}_e^T\boldsymbol{X}_e)^{-1} = \begin{bmatrix} \boldsymbol{X}^T\boldsymbol{X} & \boldsymbol{X}^T\boldsymbol{V}_e \\ \boldsymbol{V}_e^T\boldsymbol{X} & \boldsymbol{V}_e^T\boldsymbol{V}_e \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{A} + \boldsymbol{A}\boldsymbol{X}^T\boldsymbol{V}_e\boldsymbol{C}_e^{-1}\boldsymbol{V}_e^T\boldsymbol{X}\boldsymbol{A}^T & -\boldsymbol{A}\boldsymbol{X}^T\boldsymbol{V}_e\boldsymbol{C}_e^{-1} \\ -\boldsymbol{C}_e^{-1}\boldsymbol{V}_e^T\boldsymbol{X}\boldsymbol{A}^T & \boldsymbol{C}_e^{-1} \end{bmatrix},$$

where $\boldsymbol{A} \equiv (\boldsymbol{X}^T\boldsymbol{X})^{-1}$, $\boldsymbol{C}_e \equiv \boldsymbol{V}_e^T\boldsymbol{V}_e - \boldsymbol{V}_e^T\boldsymbol{H}\boldsymbol{V}_e$ and $\boldsymbol{H} \equiv \boldsymbol{X}\boldsymbol{A}\boldsymbol{X}^T$. Thus, we have

$$\begin{aligned} \boldsymbol{H}_e = \boldsymbol{X}_e(\boldsymbol{X}_e^T\boldsymbol{X}_e)^{-1}\boldsymbol{X}_e^T &= \begin{bmatrix} \boldsymbol{X}, & \boldsymbol{V}_e \end{bmatrix} \begin{bmatrix} \boldsymbol{A} + \boldsymbol{A}\boldsymbol{X}^T\boldsymbol{V}_e\boldsymbol{C}_e^{-1}\boldsymbol{V}_e^T\boldsymbol{X}\boldsymbol{A}^T & -\boldsymbol{A}\boldsymbol{X}^T\boldsymbol{V}_e\boldsymbol{C}_e^{-1} \\ -\boldsymbol{C}_e^{-1}\boldsymbol{V}_e^T\boldsymbol{X}\boldsymbol{A}^T & \boldsymbol{C}_e^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{X}^T \\ \boldsymbol{V}_e^T \end{bmatrix} \\ &= \boldsymbol{H} + [\boldsymbol{H}\boldsymbol{V}_e - \boldsymbol{V}_e]\boldsymbol{C}_e^{-1}[\boldsymbol{H}\boldsymbol{V}_e - \boldsymbol{V}_e]^T. \end{aligned}$$

Consequently,

$$\begin{aligned} \boldsymbol{Y}^T\boldsymbol{H}_e\boldsymbol{Y} &= \boldsymbol{Y}^T\left\{\boldsymbol{H} + [\boldsymbol{H}\boldsymbol{V}_e - \boldsymbol{V}_e]\boldsymbol{C}_e^{-1}[\boldsymbol{H}\boldsymbol{V}_e - \boldsymbol{V}_e]^T\right\}\boldsymbol{Y} \\ &= \boldsymbol{Y}^T\boldsymbol{H}\boldsymbol{Y} + \boldsymbol{Y}^T[\boldsymbol{H} - \boldsymbol{I}]\boldsymbol{V}_e\boldsymbol{C}_e^{-1}\boldsymbol{V}_e^T\underbrace{[\boldsymbol{H} - \boldsymbol{I}]\boldsymbol{Y}}_{\equiv\, \boldsymbol{r}} \\ &= \boldsymbol{Y}^T\boldsymbol{H}\boldsymbol{Y} + \boldsymbol{r}^T\boldsymbol{V}_e\boldsymbol{C}_e^{-1}\boldsymbol{V}_e^T\boldsymbol{r} \\ &= \boldsymbol{Y}^T\boldsymbol{H}\boldsymbol{Y} + \boldsymbol{r}^T\boldsymbol{V}_e(\boldsymbol{V}_e^T\boldsymbol{V}_e - \boldsymbol{V}_e^T\boldsymbol{H}\boldsymbol{V}_e)^{-1}\boldsymbol{V}_e^T\boldsymbol{r} \\ &= \boldsymbol{Y}^T\boldsymbol{H}\boldsymbol{Y} + \boldsymbol{r}^T\boldsymbol{V}_e(\boldsymbol{V}_e^T\boldsymbol{V}_e - \boldsymbol{V}_e^T\underbrace{\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1/2}}_{\equiv\, \boldsymbol{B}_{n\times(p-q)}}(\boldsymbol{X}^T\boldsymbol{X})^{-1/2}\boldsymbol{X}^T\boldsymbol{V}_e)^{-1}\boldsymbol{V}_e^T\boldsymbol{r} \\ &= \boldsymbol{Y}^T\boldsymbol{H}\boldsymbol{Y} + \boldsymbol{r}^T\boldsymbol{V}_e(\boldsymbol{V}_e^T\boldsymbol{V}_e - \boldsymbol{V}_e^T\boldsymbol{B}\boldsymbol{B}^T\boldsymbol{V}_e)^{-1}\boldsymbol{V}_e^T\boldsymbol{r}. \end{aligned} \tag{2.2}$$

Notice that in equation (2.2), only the three terms, i.e., $\boldsymbol{V}_e^T\boldsymbol{V}_e$, $\boldsymbol{V}_e^T\boldsymbol{r}$ and $\boldsymbol{V}_e^T\boldsymbol{B}$, involve e. Developing recursive formulas for these terms is critical in our fast grid search algorithm, which enables the second term of equation (2.2) to be sequentially updated in computing the log likelihood of the submodel. Suppose that the rows of the design matrix $\boldsymbol{X}_e$ are ordered by increasing values of $\boldsymbol{x}$. Consider two arbitrary consecutive values of e, say, e_t and e_{t+1}, each corresponding to the k^{th} and $k+1^{th}$ ordered values of $\boldsymbol{x}$ in the dataset. Then we obtain

$$\boldsymbol{V}_{e_{t+1}} - \boldsymbol{V}_{e_t} = \left[d_t^{(1)}\boldsymbol{\delta}_t,\ \ldots,\ \left\{ \sum_{s=1}^{q} d_t^{(s)}(-1)^{s+1}\binom{q}{s}\boldsymbol{x}^{\circ(q-s)} \right\} \circ \boldsymbol{\delta}_t \right],$$

where $d_t^{(s)} \equiv e_{t+1}^s - e_t^s$, $\circ$ is the element-wise product operator and $\boldsymbol{\delta}_t$ is a n by 1 vector whose first k elements are -1 and whose remaining elements are zero.

Given a definition $D_{t,i}^{(l)} \equiv \sum_{s=1}^{l} d_t^{(s)}(-1)^s\binom{l}{s}x_i^{l-s}$, it follows that

$$\boldsymbol{V}_{e_{t+1}}^T\boldsymbol{V}_{e_{t+1}} = \boldsymbol{V}_{e_t}^T\boldsymbol{V}_{e_t} + \boldsymbol{W}_1,$$

where $\boldsymbol{W}_1$ is a q by q symmetric matrix whose (l,m)th entry is

$$\sum_{i=1}^{k}\{(x_i - e_t)^m D_{t,i}^{(l)} + (x_i - e_t)^l D_{t,i}^{(m)} + D_{t,i}^{(l)}D_{t,i}^{(m)}\}, \tag{2.3}$$

and

$$\boldsymbol{V}_{e_{t+1}}^T\boldsymbol{r} = \boldsymbol{V}_{e_t}^T\boldsymbol{r} + \boldsymbol{w},$$

where $\boldsymbol{w}$ is a vector of length q whose l-th element is

$$\sum_{i=1}^{k} D_{t,i}^{(l)} r_i, \tag{2.4}$$

r_i is the i-th element of $\boldsymbol{r}$, and lastly,

$$\boldsymbol{V}_{e_{t+1}}^T\boldsymbol{B} = \boldsymbol{V}_{e_t}^T\boldsymbol{B} + \boldsymbol{W}_2,$$

where $\boldsymbol{W}_2$ is a q by $p-q$ matrix whose (l,m)th entry is

$$\sum_{i=1}^{k} D_{t,i}^{(l)} B_{i,m}, \tag{2.5}$$

and $B_{i,m}$ is the (i,m)th entry of $\boldsymbol{B}$.

The complete description of the proposed fast grid search algorithm for Mq0 is given in Algorithm 1.

Algorithm 1 Fast grid search algorithm for higher-order two-phase regression models

1. Sort the observations in ascending order of x_i.
2. Compute and store $\boldsymbol{V}_{e_1}^T\boldsymbol{V}_{e_1}$, $\boldsymbol{V}_{e_1}^T\boldsymbol{r}$ and $\boldsymbol{V}_{e_1}^T\boldsymbol{B}$.
3. Compute and store $\boldsymbol{Y}^T\boldsymbol{H}\boldsymbol{Y} - \boldsymbol{Y}^T\boldsymbol{H}_{e_1}\boldsymbol{Y}$ according to (2.2).
4. For t in 1 to $M-1$:
   (a) update $\boldsymbol{V}_{e_{t+1}}^T\boldsymbol{V}_{e_{t+1}}$, $\boldsymbol{V}_{e_{t+1}}^T\boldsymbol{r}$ and $\boldsymbol{V}_{e_{t+1}}^T\boldsymbol{B}$ according to (2.3), (2.4) and (2.5),
   (b) compute and store $\boldsymbol{Y}^T\boldsymbol{H}\boldsymbol{Y} - \boldsymbol{Y}^T\boldsymbol{H}_{e_{t+1}}\boldsymbol{Y}$ according to (2.2).

Among the candidate values $e_1,\ldots,e_M$, the MLE of the threshold parameter is the one that minimizes $\boldsymbol{Y}^T\boldsymbol{H}\boldsymbol{Y} - \boldsymbol{Y}^T\boldsymbol{H}_e\boldsymbol{Y}$. The MLE of the coefficient parameters follows immediately from the corresponding submodel.

In fact, Algorithm 1 works for any other higher-order two-phase regression models, such as M0q, Mq1, Mqq, Mqqc, by substituting (2.3), (2.4) and (2.5) accordingly. Below we give the detailed derivations of the recursive formulas (2.3), (2.4) and (2.5) adapted to a specific model.

The mean function of M0q is given by

$$\eta(x,\boldsymbol{z}) = \tilde{\alpha}_1 + \tilde{\boldsymbol{\alpha}}_{\boldsymbol{2}}^T\boldsymbol{z} + \tilde{\beta}_1(x-\tilde{e})_+ + \tilde{\beta}_2(x-\tilde{e})_+^2 + \cdots + \tilde{\beta}_q(x-\tilde{e})_+^q$$

where $(x-\tilde{e})_+ = x-\tilde{e}$ if $x > \tilde{e}$ and 0 otherwise. One can easily notice that this is simply a reparameterization of the mean function of Mq0. That is, replacing x with $-x$ in equation (2.1)

yields $\tilde{\alpha}_1 = \alpha_1, \tilde{\boldsymbol{\alpha}}_{\boldsymbol{2}} = \boldsymbol{\alpha}_{\boldsymbol{2}}$, $\tilde{\beta}_s = (-1)^s \beta_s$ for $s = 1, \ldots, q$, and $\tilde{e} = -e$. Hence fitting Mq0 with $-x$ using Algorithm 1 gives the MLE of M0q.

2.1.2 Fast grid search algorithms for Mq1 and M1q

Adding a linear term in x to equation (2.1) gives the following mean function of Mq1:

$$\eta(x, \boldsymbol{z}) = \alpha_1 + \boldsymbol{\alpha}_{\boldsymbol{2}}^T \boldsymbol{z} + \beta_1 (x - e)_- + \beta_2 (x - e)_-^2 + \cdots + \beta_q (x - e)_-^q + \gamma x. \tag{2.6}$$

We observe that the design matrix of Mq1 is decomposed as $\boldsymbol{X}_e = [\boldsymbol{X}, \boldsymbol{V}_e]$, where $\boldsymbol{X} \equiv [\boldsymbol{1}, \boldsymbol{Z}, \boldsymbol{x}]$ and $\boldsymbol{V}_e \equiv [\boldsymbol{v}_e, \boldsymbol{v}_e^{\circ 2}, \ldots, \boldsymbol{v}_e^{\circ q}]$. Since $\boldsymbol{V}_e$ remains the same as in Mq0, so do the formulas (2.3), (2.4) and (2.5).

It immediately follows that the mean function of M1q

$$\eta(x, \boldsymbol{z}) = \tilde{\alpha}_1 + \tilde{\boldsymbol{\alpha}}_{\boldsymbol{2}}^T \boldsymbol{z} + \tilde{\beta}_1 (x - \tilde{e})_+ + \tilde{\beta}_2 (x - \tilde{e})_+^2 + \cdots + \tilde{\beta}_q (x - \tilde{e})_+^q + \tilde{\gamma} x$$

is a reparameterized version of equation (2.6). By substituting $-x$ for x in (2.6), we obtain $\tilde{\alpha}_1 = \alpha_1, \tilde{\boldsymbol{\alpha}}_{\boldsymbol{2}} = \boldsymbol{\alpha}_{\boldsymbol{2}}$, $\tilde{\beta}_s = (-1)^s \beta_s$ for $s = 1, \ldots, q$, $\tilde{\gamma} = -\gamma$, $\tilde{e} = -e$ and the same argument in Section 2.1.1 applies here.

2.1.3 Fast grid search algorithm for Mqq

For Mqq, consider the following mean function:

$$\begin{aligned} \eta(x, \boldsymbol{z}) = \alpha_1 + \boldsymbol{\alpha}_{\boldsymbol{2}}^T \boldsymbol{z} + \beta_{1,-} (x - e)_- + \beta_{1,+} (x - e)_+ + \beta_{2,-} (x - e)_-^2 + \beta_{2,+} (x - e)_+^2 \\ + \cdots + \beta_{q,-} (x - e)_-^q + \beta_{q,+} (x - e)_+^q. \end{aligned} \tag{2.7}$$

The design matrix $\boldsymbol{X}_e$ is now split into $\boldsymbol{X} \equiv [\boldsymbol{1}, \boldsymbol{Z}]$ and $\boldsymbol{V}_e \equiv [\boldsymbol{v}_e, \boldsymbol{u}_e, \ldots, \boldsymbol{v}_e^{\circ q}, \boldsymbol{u}_e^{\circ q}]$, where $\boldsymbol{u}_e \equiv (\boldsymbol{x} - \boldsymbol{1}e)_+$.

With $D_{t,i}^{(l)} \equiv \sum_{s=1}^{l} d_t^{(s)} (-1)^s \binom{l}{s} x_i^{l-s}$ (as defined earlier) and $\boldsymbol{D}_t^{(l)} \equiv (D_{t,1}^{(l)}, \ldots, D_{t,n}^{(l)})$,

$$\boldsymbol{V}_{e_{t+1}} - \boldsymbol{V}_{e_t} = [-\boldsymbol{D}_t^{(1)} \circ \boldsymbol{\delta}_t, -\boldsymbol{D}_t^{(1)} \circ \tilde{\boldsymbol{\delta}}_t, \ldots, -\boldsymbol{D}_t^{(q)} \circ \boldsymbol{\delta}_t, -\boldsymbol{D}_t^{(q)} \circ \tilde{\boldsymbol{\delta}}_t],$$

where $\boldsymbol{\delta}_t$ is defined as before and $\tilde{\boldsymbol{\delta}}_t$ is a n by 1 vector whose first k elements are zero and whose remaining elements are -1. Then we derive the recursive formulas for Algorithm 1 estimating the MLE of Mqq as follows:

$$\boldsymbol{V}_{e_{t+1}}^T \boldsymbol{V}_{e_{t+1}} = \boldsymbol{V}_{e_t}^T \boldsymbol{V}_{e_t} + \boldsymbol{W}_1^*,$$

where $\boldsymbol{W}_1^*$ is a $2q$ by $2q$ symmetric matrix whose (l,m)th entry is

$$\begin{cases} \sum_{i=1}^{k} \{D_{t,i}^{(s)} D_{t,i}^{(\tilde{s})} + (x_i - e_t)^s D_{t,i}^{(\tilde{s})} + (x_i - e_t)^{\tilde{s}} D_{t,i}^{(s)}\}, & \\ \qquad\qquad \text{for } l = 2s-1,\ m = 2\tilde{s}-1 \ (s, \tilde{s} = 1, \ldots, q) & \\ \sum_{i=k+1}^{n} \{D_{t,i}^{(s)} D_{t,i}^{(\tilde{s})} + (x_i - e_t)^s D_{t,i}^{(\tilde{s})} + (x_i - e_t)^{\tilde{s}} D_{t,i}^{(s)}\}, & \\ \qquad\qquad \text{for } l = 2s,\ m = 2\tilde{s} \ (s, \tilde{s} = 1, \ldots, q) & \\ 0 \qquad\qquad \text{o.w.}, & \end{cases}$$

and

$$\boldsymbol{V}_{e_{t+1}}^T \boldsymbol{r} = \boldsymbol{V}_{e_t}^T \boldsymbol{r} + \boldsymbol{w}^*,$$

where $\boldsymbol{w}^*$ is a vector of length $2q$ whose l-th element is

$$\begin{cases} \sum_{i=1}^{k} D_{t,i}^{(s)} r_i, & \text{for } l = 2s-1 \ (s = 1, \ldots, q) \\ \sum_{i=k+1}^{n} D_{t,i}^{(s)} r_i, & \text{for } l = 2s \ (s = 1, \ldots, q), \end{cases}$$

and

$$\boldsymbol{V}_{e_{t+1}}^T \boldsymbol{B} = \boldsymbol{V}_{e_t}^T \boldsymbol{B} + \boldsymbol{W}_2^*,$$

where $\boldsymbol{W}_2^*$ is a $2q$ by $p - 2q$ matrix whose (l,m)th entry is

$$\begin{cases} \sum_{i=1}^{k} D_{t,i}^{(s)} B_{i,m}, & \text{for } l = 2s-1 \ (s = 1, \ldots, q) \\ \sum_{i=k+1}^{n} D_{t,i}^{(s)} B_{i,m}, & \text{for } l = 2s \ (s = 1, \ldots, q). \end{cases}$$

2.1.4 Fast grid search algorithm for Mqqc

The mean function of Mqqc is expressed as

$$\eta(x, \boldsymbol{z}) = \alpha_1 + \boldsymbol{\alpha}_{\boldsymbol{2}}^T \boldsymbol{z} + \beta_2(x-e)^2 + \cdots + \beta_{q-1}(x-e)^{q-1} + \beta_{q,-}(x-e)_-^q + \beta_{q,+}(x-e)_+^q + \gamma x$$

after imposing the smoothness constraint $\beta_{s,-} = \beta_{s,+} \equiv \beta_s$ for $s = 1, \ldots, q-1$ on equation (2.7) and letting $\beta_1 = \gamma$. In other words, Mqqc satisfies the smoothness condition that the s-th derivatives, with respect to x, of the q-th order polynomials before and after the threshold e are the same when evaluated at $x = e$. Then with $\boldsymbol{X} \equiv [\boldsymbol{1}, \boldsymbol{Z}, \boldsymbol{x}]$ and $\boldsymbol{V}_e \equiv [(\boldsymbol{x} - \boldsymbol{1}e)^{\circ 2}, \ldots, (\boldsymbol{x} - \boldsymbol{1}e)^{\circ(q-1)}, \boldsymbol{v}_e^{\circ q}, \boldsymbol{u}_e^{\circ q}]$, we have the usual decomposition $\boldsymbol{X}_e = [\boldsymbol{X}, \boldsymbol{V}_e]$. It follows that

$$\boldsymbol{V}_{e_{t+1}} - \boldsymbol{V}_{e_t} = [\boldsymbol{D}_t^{(1)}, \ldots, \boldsymbol{D}_t^{(q-1)}, -\boldsymbol{D}_t^{(q)} \circ \boldsymbol{\delta}_t, -\boldsymbol{D}_t^{(q)} \circ \tilde{\boldsymbol{\delta}}_t],$$

where $\boldsymbol{D}_t^{(q)}$, $\boldsymbol{\delta}_t$ and $\tilde{\boldsymbol{\delta}}_t$ are defined as before. This yields the following recursive formulas for Mqqc.

$$\boldsymbol{V}_{e_{t+1}}^T \boldsymbol{V}_{e_{t+1}} = \boldsymbol{V}_{e_t}^T \boldsymbol{V}_{e_t} + \boldsymbol{W}_1^{**},$$

where $\boldsymbol{W}_1^{**}$ is a $q+1$ by $q+1$ symmetric matrix whose (l,m)th entry is

$$\begin{cases} \sum_{i=1}^{n}\{D_{t,i}^{(l)}D_{t,i}^{(m)} + (x_i - e_t)^l D_{t,i}^{(m)} + (x_i - e_t)^m D_{t,i}^{(l)}\}, \\ \qquad\qquad \text{for } 1 \le l \le q-1,\ 1 \le m \le q-1 \\ \sum_{i=1}^{k}\{D_{t,i}^{(l)}D_{t,i}^{(m)} + (x_i - e_t)^l D_{t,i}^{(m)} + (x_i - e_t)^m D_{t,i}^{(l)}\}, \\ \qquad\qquad \begin{cases} \text{for } l = q,\ 1 \le m \le q \\ \text{or } 1 \le l \le q,\ m = q \end{cases} \\ \sum_{i=k+1}^{n}\{D_{t,i}^{(l-1)}D_{t,i}^{(m)} + (x_i - e_t)^{l-1} D_{t,i}^{(m)} + (x_i - e_t)^m D_{t,i}^{(l-1)}\}, \\ \qquad\qquad \text{for } l = q+1,\ 1 \le m \le q-1 \\ \sum_{i=k+1}^{n}\{D_{t,i}^{(l)}D_{t,i}^{(m-1)} + (x_i - e_t)^l D_{t,i}^{(m-1)} + (x_i - e_t)^{m-1} D_{t,i}^{(l)}\}, \\ \qquad\qquad \text{for } 1 \le l \le q-1,\ m = q+1 \\ \sum_{i=k+1}^{n}\{D_{t,i}^{(l-1)}D_{t,i}^{(m-1)} + (x_i - e_t)^{l-1} D_{t,i}^{(m-1)} + (x_i - e_t)^{m-1} D_{t,i}^{(l-1)}\}, \\ \qquad\qquad \text{for } l = m = q+1 \\ 0, \qquad\qquad \begin{cases} \text{for } l = q+1,\ m = q \\ \text{or } l = q,\ m = q+1 \end{cases} \end{cases}$$

and

$$\boldsymbol{V}_{e_{t+1}}^T \boldsymbol{r} = \boldsymbol{V}_{e_t}^T \boldsymbol{r} + \boldsymbol{w}^{**},$$

where $\boldsymbol{w}^{**}$ is a vector of length $q+1$ whose l-th element is

$$\begin{cases} \sum_{i=1}^{n} D_{t,i}^{(l)} r_i, & \text{for } 1 \le l \le q-1 \\ \sum_{i=1}^{k} D_{t,i}^{(l)} r_i, & \text{for } l = q \\ \sum_{i=k+1}^{n} D_{t,i}^{(l-1)} r_i, & \text{for } l = q+1, \end{cases}$$

and

$$\boldsymbol{V}_{e_{t+1}}^T \boldsymbol{B} = \boldsymbol{V}_{e_t}^T \boldsymbol{B} + \boldsymbol{W}_2^{**},$$

where $\boldsymbol{W}_2^{**}$ is a $q+1$ by $p-(q+1)$ matrix whose (l,m)th entry is

$$\begin{cases} \sum_{i=1}^n D_{t,i}^{(l)} B_{i,m}, & \text{for } 1 \le l \le q-1 \\ \sum_{i=1}^k D_{t,i}^{(l)} B_{i,m}, & \text{for } l = q \\ \sum_{i=k+1}^n D_{t,i}^{(l-1)} B_{i,m}, & \text{for } l = q+1. \end{cases}$$

2.2 *Discussion of special cases*

2.2.1 *Quadratic two-phase regression models without smoothness constraints: M20, M02, M21, M12, M22*

In this subsection, we examine the quadratic two-phase regression model with no smoothness constraint, namely M20, M02, M21, M12 and M22. For each model, we express the recursive formulas discussed in Section 2.1 in a closed form for $q = 2$.

For M20 of the form

$$\eta(x, \boldsymbol{z}) = \alpha_1 + \boldsymbol{\alpha_2}^T \boldsymbol{z} + \beta_1 (x-e)_- + \beta_2 (x-e)_-^2,$$

equations (2.3), (2.4) and (2.5) are reduced to

$$\boldsymbol{W}_1 = \begin{bmatrix} c_1 & c_2 \\ c_2 & c_3 \end{bmatrix},$$

where, with $\Sigma_x^k = \sum_{i=1}^k x_i$, $\Sigma_{x^2}^k = \sum_{i=1}^k x_i^2$ and $\Sigma_{x^3}^k = \sum_{i=1}^k x_i^3$,

$$\begin{aligned} c_1 &= -2d_t^{(1)} \Sigma_x^k + 2kd_t^{(1)} e_t + k(d_t^{(1)})^2, \\ c_2 &= -3d_t^{(1)} \Sigma_{x^2}^k + (2(d_t^{(1)})^2 + 4d_t^{(1)} e_t + d_t^{(2)}) \Sigma_x^k - kd_t^{(1)} d_t^{(2)} - kd_t^{(2)} e_t - kd_t^{(1)} e_t^2, \\ c_3 &= -4d_t^{(1)} \Sigma_{x^3}^k + (4(d_t^{(1)})^2 + 8d_t^{(1)} e_t + 2d_t^{(2)}) \Sigma_{x^2}^k + (-4d_t^{(1)} d_t^{(2)} - 4d_t^{(1)} e_t^2 - 4d_t^{(2)} e_t) \Sigma_x^k \end{aligned}$$

$$+ k(d_t^{(2)})^2 + 2kd_t^{(2)}e_t^2,$$

and with $\Sigma_r^k = \sum_{i=1}^k r_i$ and $\Sigma_{xr}^k = \sum_{i=1}^k (x_i r_i)$,

$$\boldsymbol{w} = \begin{bmatrix} -d_t^{(1)}\Sigma_r^k \\ -2d_t^{(1)}\Sigma_{xr}^k + d_t^{(2)}\Sigma_r^k \end{bmatrix},$$

and with $\Sigma_{B_1}^k = \sum_{i=1}^k B_{i,1}$ and $\Sigma_{xB_1}^k = \sum_{i=1}^k B_{i,1}x_i$,

$$\boldsymbol{W}_2 = \begin{bmatrix} -d_t^{(1)}\Sigma_{B_1}^k & , \cdots , & -d_t^{(1)}\Sigma_{B_{p-2}}^k \\ -2d_t^{(1)}\Sigma_{xB_1}^k + d_t^{(2)}\Sigma_{B_1}^k & , \cdots , & -2d_t^{(1)}\Sigma_{xB_{p-2}}^k + d_t^{(2)}\Sigma_{B_{p-2}}^k \end{bmatrix}.$$

These results are directly applicable to estimating M02, M21 and M12 of the following forms by using the argument in Sections 2.1.1 and 2.1.2:

$$\eta(x, \boldsymbol{z}) = \tilde{\alpha}_1 + \tilde{\boldsymbol{\alpha}}_{\boldsymbol{2}}^T \boldsymbol{z} + \tilde{\beta}_1(x - \tilde{e})_+ + \tilde{\beta}_2(x - \tilde{e})_+^2, \tag{M02}$$

$$\eta(x, \boldsymbol{z}) = \alpha_1 + \boldsymbol{\alpha}_{\boldsymbol{2}}^T \boldsymbol{z} + \beta_1(x - e)_- + \beta_2(x - e)_-^2 + \gamma x, \tag{M21}$$

$$\eta(x, \boldsymbol{z}) = \tilde{\alpha}_1 + \tilde{\boldsymbol{\alpha}}_{\boldsymbol{2}}^T \boldsymbol{z} + \tilde{\beta}_1(x - \tilde{e})_+ + \tilde{\beta}_2(x - \tilde{e})_+^2 + \tilde{\gamma}x. \tag{M12}$$

Lastly, M22 of the form

$$\eta(x, \boldsymbol{z}) = \alpha_1 + \boldsymbol{\alpha}_{\boldsymbol{2}}^T \boldsymbol{z} + \beta_{1,-}(x - e)_- + \beta_{1,+}(x - e)_+ + \beta_{2,-}(x - e)_-^2 + \beta_{2,+}(x - e)_+^2,$$

is estimated by Algorithm 1 using the following:

$$\boldsymbol{W}_1^* = \begin{bmatrix} c_1 & 0 & c_2 & 0 \\ 0 & c_4 & 0 & c_5 \\ c_2 & 0 & c_3 & 0 \\ 0 & c_5 & 0 & c_6 \end{bmatrix},$$

where, with $\Sigma_x^{n-k} = \sum_{i=k+1}^{n} x_i$, $\Sigma_{x^2}^{n-k} = \sum_{i=k+1}^{n} x_i^2$, and $\Sigma_{x^3}^{n-k} = \sum_{i=k+1}^{n} x_i^3$,

$$
\begin{aligned}
c_4 &= -2d_t^{(1)}\Sigma_x^{n-k} + 2(n-k)d_t^{(1)}e_t + (n-k)(d_t^{(1)})^2, \\
c_5 &= -3d_t^{(1)}\Sigma_{x^2}^{n-k} + (2(d_t^{(1)})^2 + 4d_t^{(1)}e_t + d_t^{(2)})\Sigma_x^{n-k} - (n-k)d_t^{(1)}d_t^{(2)} - (n-k)d_t^{(2)}e_t \\
&\quad - (n-k)d_t^{(1)}e_t^2, \\
c_6 &= -4d_t^{(1)}\Sigma_{x^3}^{n-k} + (4(d_t^{(1)})^2 + 8d_t^{(1)}e_t + 2d_t^{(2)})\Sigma_{x^2}^{n-k} + (-4d_t^{(1)}d_t^{(2)} - 4d_t^{(1)}e_t^2 - 4d_t^{(2)}e_t)\Sigma_x^{n-k} \\
&\quad + (n-k)(d_t^{(2)})^2 + 2(n-k)d_t^{(2)}e_t^2,
\end{aligned}
$$

and with $\Sigma_r^{n-k} = \sum_{i=k+1}^{n} r_i$, and $\Sigma_{xr}^{n-k} = \sum_{i=k+1}^{n}(x_i r_i)$,

$$
\boldsymbol{w}^* = \begin{bmatrix} -d_t^{(1)}\Sigma_r^k \\ -d_t^{(1)}\Sigma_r^{n-k} \\ -2d_t^{(1)}\Sigma_{xr}^k + d_t^{(2)}\Sigma_r^k \\ -2d_t^{(1)}\Sigma_{xr}^{n-k} + d_t^{(2)}\Sigma_r^{n-k} \end{bmatrix}
$$

and with $\Sigma_{B_1}^{n-k} = \sum_{i=k+1}^{n} B_{i,1}$, and $\Sigma_{xB_1}^{n-k} = \sum_{i=k+1}^{n} B_{i,1}x_i$,

$$
\boldsymbol{W}_2^* = \begin{bmatrix} -d_t^{(1)}\Sigma_{B_1}^k & ,\cdots, & -d_t^{(1)}\Sigma_{B_{p-4}}^k \\ -d_t^{(1)}\Sigma_{B_1}^{n-k} & ,\cdots, & -d_t^{(1)}\Sigma_{B_{p-4}}^{n-k} \\ -2d_t^{(1)}\Sigma_{xB_1}^k + d_t^{(2)}\Sigma_{B_1}^k & ,\cdots, & -2d_t^{(1)}\Sigma_{xB_{p-4}}^k + d_t^{(2)}\Sigma_{B_{p-4}}^k \\ -2d_t^{(1)}\Sigma_{xB_1}^{n-k} + d_t^{(2)}\Sigma_{B_1}^{n-k} & ,\cdots, & -2d_t^{(1)}\Sigma_{xB_{p-4}}^{n-k} + d_t^{(2)}\Sigma_{B_{p-4}}^{n-k} \end{bmatrix}.
$$

2.2.2 Quadratic two-phase regression models with smoothness constraints: M21c, M12c, M22c

We now consider three types of quadratic two-phase regression models with smoothness constraints: M21c, M12c and M22c. Due to its complexity, we have so far focused on developing the fast grid search algorithm only for Mqqc among q^2 possible models under smoothness constraints. Yet the problem becomes simple when $q = 2$, as there exist only two models (M21c and M12c) of interest. These low-order models are also more likely to be useful in modeling real data. Thus we discuss them in detail below along with M22c.

M21c imposes a smoothness condition $\beta_1 = 0$ on M21 so that the first derivatives, evaluated at

the threshold e, of a quadratic curve before e and a line after e are the same. Consequently, the mean function of M21c is expressed as

$$\eta(x, \boldsymbol{z}) = \alpha_1 + \boldsymbol{\alpha_2}^T \boldsymbol{z} + \beta_{2,-}(x-e)_-^2 + \gamma x,$$

and by $\boldsymbol{V}_{e_{t+1}} - \boldsymbol{V}_{e_t} = [2d_t^{(1)}(\boldsymbol{x} \circ \boldsymbol{\delta}_t) - d_t^{(2)}\boldsymbol{\delta}_t]$, we obtain the following recursive formulas for $\boldsymbol{V}_{e_{t+1}}^T \boldsymbol{V}_{e_{t+1}}$, $\boldsymbol{V}_{e_{t+1}}^T \boldsymbol{r}$ and $\boldsymbol{V}_{e_{t+1}}^T \boldsymbol{B}$:

$$\begin{aligned}
\boldsymbol{V}_{e_{t+1}}^T \boldsymbol{V}_{e_{t+1}} &= \boldsymbol{V}_{e_t}^T \boldsymbol{V}_{e_t} + c_3, \\
\boldsymbol{V}_{e_{t+1}}^T \boldsymbol{r} &= \boldsymbol{V}_{e_t}^T \boldsymbol{r} - 2d_t^{(1)}\Sigma_{xr}^k + d_t^{(2)}\Sigma_r^k, \\
\boldsymbol{V}_{e_{t+1}}^T \boldsymbol{B} &= \boldsymbol{V}_{e_t}^T \boldsymbol{B} + \left[-2d_t^{(1)}\Sigma_{xB_1}^k + d_t^{(2)}\Sigma_{B_1}^k \quad, \cdots, \quad -2d_t^{(1)}\Sigma_{xB_{p-1}}^k + d_t^{(2)}\Sigma_{B_{p-1}}^k\right].
\end{aligned}$$

M12c of the form

$$\eta(x, \boldsymbol{z}) = \tilde{\alpha}_1 + \tilde{\boldsymbol{\alpha}}_{\boldsymbol{2}}^T \boldsymbol{z} + \tilde{\beta}_{2,+}(x-\tilde{e})_+^2 + \tilde{\gamma} x,$$

after imposing the same smoothness condition $\beta_1 = 0$ on M12 is estimated by reparameterizing M21c as usual. Substituting $-x$ for x in the mean function of M21c gives $\tilde{\alpha}_1 = \alpha_1, \tilde{\boldsymbol{\alpha}}_{\boldsymbol{2}} = \boldsymbol{\alpha_2}, \tilde{\beta}_{2,+} = \beta_{2,-}, \tilde{\gamma} = -\gamma$ and $\tilde{e} = -e$.

After imposing a smoothness condition $\beta_{1,-} = \beta_{1,+}$ on M22 to have the same first derivatives of quadratic curves before and after the threshold, we have M22c of the form

$$\eta(x, \boldsymbol{z}) = \alpha_1 + \boldsymbol{\alpha_2}^T \boldsymbol{z} + \beta_{2,-}(x-e)_-^2 + \beta_{2,+}(x-e)_+^2 + \gamma x.$$

Then the argument in Section 2.1.4 applies to M22c as follows.

$$\boldsymbol{W}_1^{**} = \begin{bmatrix} c_7 & c_2 & c_5 \\ c_2 & c_3 & 0 \\ c_5 & 0 & c_6 \end{bmatrix},$$

where, with $\Sigma_x^n = \sum_{i=1}^n x_i$,

$$c_7 = -2d_t^{(1)}\Sigma_x^n + 2nd_t^{(1)}e_t + n(d_t^{(1)})^2,$$

and with $\Sigma_r^n = \sum_{i=1}^n r_i$,

$$\boldsymbol{w}^{**} = \begin{bmatrix} -d_t^{(1)}\Sigma_r^n \\ -2d_t^{(1)}\Sigma_{xr}^k + d_t^{(2)}\Sigma_r^k \\ -2d_t^{(1)}\Sigma_{xr}^{n-k} + d_t^{(2)}\Sigma_r^{n-k} \end{bmatrix},$$

and with $\Sigma_{B_1}^n = \sum_{i=1}^n B_{i,1}$,

$$\boldsymbol{W}_2^{**} = \begin{bmatrix} -d_t^{(1)}\Sigma_{B_1}^n & ,\cdots, & -d_t^{(1)}\Sigma_{B_{p-3}}^n \\ -2d_t^{(1)}\Sigma_{xB_1}^k + d_t^{(2)}\Sigma_{B_1}^k & ,\cdots, & -2d_t^{(1)}\Sigma_{xB_{p-3}}^k + d_t^{(2)}\Sigma_{B_{p-3}}^k \\ -2d_t^{(1)}\Sigma_{xB_1}^{n-k} + d_t^{(2)}\Sigma_{B_1}^{n-k} & ,\cdots, & -2d_t^{(1)}\Sigma_{xB_{p-3}}^{n-k} + d_t^{(2)}\Sigma_{B_{p-3}}^{n-k} \end{bmatrix}.$$

Not surprisingly, the last two rows of $\boldsymbol{w}^{**}$ and $\boldsymbol{W}_2^{**}$ remain the same as $\boldsymbol{w}^*$ and $\boldsymbol{W}_2^*$ for M22.

2.2.3 Cubic two-phase regression models without smoothness constraints: M30, M03, M31, M13, M33

Here we explicitly give the recursive formulas for the cubic two-phase regression model without smoothness constraints ($q = 3$). There are five models to consider:

$$\eta(x, \boldsymbol{z}) = \alpha_1 + \boldsymbol{\alpha_2}^T\boldsymbol{z} + \beta_1(x-e)_- + \beta_2(x-e)_-^2 + \beta_3(x-e)_-^3, \tag{M30}$$

$$\eta(x, \boldsymbol{z}) = \tilde{\alpha}_1 + \tilde{\boldsymbol{\alpha}}_{\boldsymbol{2}}^T\boldsymbol{z} + \tilde{\beta}_1(x-\tilde{e})_+ + \tilde{\beta}_2(x-\tilde{e})_+^2 + \tilde{\beta}_3(x-\tilde{e})_+^3, \tag{M03}$$

$$\eta(x, \boldsymbol{z}) = \alpha_1 + \boldsymbol{\alpha_2}^T\boldsymbol{z} + \beta_1(x-e)_- + \beta_2(x-e)_-^2 + \beta_3(x-e)_-^3 + \gamma x, \tag{M31}$$

$$\eta(x, \boldsymbol{z}) = \tilde{\alpha}_1 + \tilde{\boldsymbol{\alpha}}_{\boldsymbol{2}}^T\boldsymbol{z} + \tilde{\beta}_1(x-\tilde{e})_+ + \tilde{\beta}_2(x-\tilde{e})_+^2 + \tilde{\beta}_3(x-e)_+^3 + \tilde{\gamma}x. \tag{M13}$$

$$\begin{aligned}\eta(x, \boldsymbol{z}) = \alpha_1 + \boldsymbol{\alpha_2}^T\boldsymbol{z} + \beta_1(x-e)_- + \beta_2(x-e)_+ + \beta_3(x-e)_-^2 + \beta_4(x-e)_+^2 \\ + \beta_5(x-e)_-^3 + \beta_6(x-e)_+^3\end{aligned} \tag{M33}$$

For M30, the formulas take the form

$$\boldsymbol{W}_1 = \begin{bmatrix} c_1 & c_2 & c_8 \\ c_2 & c_3 & c_9 \\ c_8 & c_9 & c_{10} \end{bmatrix},$$

where, with $\Sigma^k_{x^4} = \sum_{i=1}^k x_i^4$ and $\Sigma^k_{x^5} = \sum_{i=1}^k x_i^5$,

$$\begin{aligned}
c_8 &= -4d_t^{(1)}\Sigma^k_{x^3} + (6d_t^{(1)}e_t + 3d_t^{(2)} + 3(d_t^{(1)})^2)\Sigma^k_{x^2} \\
&\quad + (-3d_t^{(1)}e_t^2 - 3d_t^{(2)}e_t - d_t^{(3)} - 3d_t^{(1)}d_t^{(2)})\Sigma^k_x + kd_t^{(1)}e_t^3 + kd_t^{(3)}e_t + kd_t^{(1)}d_t^{(3)}, \\
c_9 &= -5d_t^{(1)}\Sigma^k_{x^4} + (12d_t^{(1)}e_t + 4d_t^{(2)} + 6(d_t^{(1)})^2)\Sigma^k_{x^3} \\
&\quad + (-9d_t^{(1)}e_t^2 - 9d_t^{(2)}e_t - d_t^{(3)} - 9d_t^{(1)}d_t^{(2)})\Sigma^k_{x^2} \\
&\quad + (2d_t^{(1)}e_t^3 + 6d_t^{(2)}e_t^2 + 2d_t^{(3)}e_t + 2d_t^{(1)}d_t^{(3)} + 3(d_t^{(2)})^2)\Sigma^k_x \\
&\quad - kd_t^{(2)}e_t^3 - kd_t^{(3)}e_t^2 - kd_t^{(2)}d_t^{(3)}, \\
c_{10} &= -6d_t^{(1)}\Sigma^k_{x^5} + (18d_t^{(1)}e_t + 6d_t^{(2)} + 9(d_t^{(1)})^2)\Sigma^k_{x^4} \\
&\quad + (-18d_t^{(1)}e_t^2 - 18d_t^{(2)}e_t - 2d_t^{(3)} - 18d_t^{(1)}d_t^{(2)})\Sigma^k_{x^3} \\
&\quad + (6d_t^{(1)}e_t^3 + 18d_t^{(2)}e_t^2 + 6d_t^{(3)}e_t + 6d_t^{(1)}d_t^{(3)} + 9(d_t^{(2)})^2)\Sigma^k_{x^2} \\
&\quad + (-6d_t^{(2)}e_t^3 - 6d_t^{(3)}e_t^2 - 6d_t^{(2)}d_t^{(3)})\Sigma^k_x + 2kd_t^{(3)}e_t^3 + k(d_t^{(3)})^2,
\end{aligned}$$

and with $\Sigma^k_{x^2r} = \sum_{i=1}^k x_i^2 r_i$,

$$\boldsymbol{w} = \begin{bmatrix} -d_t^{(1)}\Sigma^k_r \\ -2d_t^{(1)}\Sigma^k_{xr} + d_t^{(2)}\Sigma^k_r \\ -3d_t^{(1)}\Sigma^k_{x^2r} + 3d_t^{(2)}\Sigma^k_{xr} - d_t^{(3)}\Sigma^k_r \end{bmatrix},$$

and with $\Sigma^k_{x^2B_1} = \sum_{i=1}^k x_i^2 B_{i,1}$,

$$\boldsymbol{W}_2 = \begin{bmatrix} -d_t^{(1)}\Sigma^k_{B_1} & ,\cdots, & -d_t^{(1)}\Sigma^k_{B_{p-3}} \\ -2d_t^{(1)}\Sigma^k_{xB_1} + d_t^{(2)}\Sigma^k_{B_1} & ,\cdots, & -2d_t^{(1)}\Sigma^k_{xB_{p-3}} + d_t^{(2)}\Sigma^k_{B_{p-3}} \\ -3d_t^{(1)}\Sigma^k_{x^2B_1} + 3d_t^{(2)}\Sigma^k_{xB_1} - d_t^{(3)}\Sigma^k_{B_1} & ,\cdots, & -3d_t^{(1)}\Sigma^k_{x^2B_{p-3}} + 3d_t^{(2)}\Sigma^k_{xB_{p-3}} - d_t^{(3)}\Sigma^k_{B_{p-3}} \end{bmatrix}.$$

With these in hand, we can readily estimate the parameters of M03, M31 and M13 according to Sections 2.1.1 and 2.1.2.

The recursive formulas are much more complicated for M33 since it contains the largest number of parameters to be estimated. That is,

$$\boldsymbol{W}_1^* = \begin{bmatrix} c_1 & 0 & c_2 & 0 & c_8 & 0 \\ 0 & c_4 & 0 & c_5 & 0 & c_{11} \\ c_2 & 0 & c_3 & 0 & c_9 & 0 \\ 0 & c_5 & 0 & c_6 & 0 & c_{12} \\ c_8 & 0 & c_9 & 0 & c_{10} & 0 \\ 0 & c_{11} & 0 & c_{12} & 0 & c_{13} \end{bmatrix},$$

where, with $\Sigma_{x^4}^{n-k} = \sum_{i=k+1}^{n} x_i^4$ and $\Sigma_{x^5}^{n-k} = \sum_{i=k+1}^{n} x_i^5$,

$$\begin{aligned}
c_{11} &= -4d_t^{(1)}\Sigma_{x^3}^{n-k} + (6d_t^{(1)}e_t + 3d_t^{(2)} + 3(d_t^{(1)})^2)\Sigma_{x^2}^{n-k} \\
&\quad + (-3d_t^{(1)}e_t^2 - 3d_t^{(2)}e_t - d_t^{(3)} - 3d_t^{(1)}d_t^{(2)})\Sigma_x^{n-k} \\
&\quad + (n-k)d_t^{(1)}e_t^3 + (n-k)d_t^{(3)}e_t + (n-k)d_t^{(1)}d_t^{(3)}, \\
c_{12} &= -5d_t^{(1)}\Sigma_{x^4}^{n-k} + (12d_t^{(1)}e_t + 4d_t^{(2)} + 6(d_t^{(1)})^2)\Sigma_{x^3}^{n-k} \\
&\quad + (-9d_t^{(1)}e_t^2 - 9d_t^{(2)}e_t - d_t^{(3)} - 9d_t^{(1)}d_t^{(2)})\Sigma_{x^2}^{n-k} \\
&\quad + (2d_t^{(1)}e_t^3 + 6d_t^{(2)}e_t^2 + 2d_t^{(3)}e_t + 2d_t^{(1)}d_t^{(3)} + 3(d_t^{(2)})^2)\Sigma_x^{n-k} \\
&\quad - (n-k)d_t^{(2)}e_t^3 - (n-k)d_t^{(3)}e_t^2 - (n-k)d_t^{(2)}d_t^{(3)}, \\
c_{13} &= -6d_t^{(1)}\Sigma_{x^5}^{n-k} + (18d_t^{(1)}e_t + 6d_t^{(2)} + 9(d_t^{(1)})^2)\Sigma_{x^4}^{n-k} \\
&\quad + (-18d_t^{(1)}e_t^2 - 18d_t^{(2)}e_t - 2d_t^{(3)} - 18d_t^{(1)}d_t^{(2)})\Sigma_{x^3}^{n-k} \\
&\quad + (6d_t^{(1)}e_t^3 + 18d_t^{(2)}e_t^2 + 6d_t^{(3)}e_t + 6d_t^{(1)}d_t^{(3)} + 9(d_t^{(2)})^2)\Sigma_{x^2}^{n-k} \\
&\quad + (-6d_t^{(2)}e_t^3 - 6d_t^{(3)}e_t^2 - 6d_t^{(2)}d_t^{(3)})\Sigma_x^{n-k} + 2(n-k)d_t^{(3)}e_t^3 + (n-k)(d_t^{(3)})^2,
\end{aligned}$$

and with $\Sigma_{x^2r}^{n-k} = \sum_{i=k+1}^{n} x_i^2 r_i$,

$$\boldsymbol{w}^* = \begin{bmatrix} -d_t^{(1)}\Sigma_r^k \\ -d_t^{(1)}\Sigma_r^{n-k} \\ -2d_t^{(1)}\Sigma_{xr}^k + d_t^{(2)}\Sigma_r^k \\ -2d_t^{(1)}\Sigma_{xr}^{n-k} + d_t^{(2)}\Sigma_r^{n-k} \\ -3d_t^{(1)}\Sigma_{x^2r}^k + 3d_t^{(2)}\Sigma_{xr}^k - d_t^{(3)}\Sigma_r^k \\ -3d_t^{(1)}\Sigma_{x^2r}^{n-k} + 3d_t^{(2)}\Sigma_{xr}^{n-k} - d_t^{(3)}\Sigma_r^{n-k} \end{bmatrix},$$

and with $\Sigma_{x^2B_1}^{n-k} = \sum_{i=k+1}^{n} x_i^2 B_{i,1}$,

$$\boldsymbol{W}_2^* = \begin{bmatrix} -d_t^{(1)}\Sigma_{B_1}^k & ,\cdots, & -d_t^{(1)}\Sigma_{B_{p-6}}^k \\ -d_t^{(1)}\Sigma_{B_1}^{n-k} & ,\cdots, & -d_t^{(1)}\Sigma_{B_{p-6}}^{n-k} \\ -2d_t^{(1)}\Sigma_{xB_1}^k + d_t^{(2)}\Sigma_{B_1}^k & ,\cdots, & -2d_t^{(1)}\Sigma_{xB_{p-6}}^k + d_t^{(2)}\Sigma_{B_{p-6}}^k \\ -2d_t^{(1)}\Sigma_{xB_1}^{n-k} + d_t^{(2)}\Sigma_{B_1}^{n-k} & ,\cdots, & -2d_t^{(1)}\Sigma_{xB_{p-6}}^{n-k} + d_t^{(2)}\Sigma_{B_{p-6}}^{n-k} \\ -3d_t^{(1)}\Sigma_{x^2B_1}^k + 3d_t^{(2)}\Sigma_{xB_1}^k - d_t^{(3)}\Sigma_{B_1}^k & ,\cdots, & -3d_t^{(1)}\Sigma_{x^2B_{p-6}}^k + 3d_t^{(2)}\Sigma_{xB_{p-6}}^k - d_t^{(3)}\Sigma_{B_{p-6}}^k \\ -3d_t^{(1)}\Sigma_{x^2B_1}^{n-k} + 3d_t^{(2)}\Sigma_{xB_1}^{n-k} - d_t^{(3)}\Sigma_{B_1}^{n-k} & ,\cdots, & -3d_t^{(1)}\Sigma_{x^2B_{p-6}}^{n-k} + 3d_t^{(2)}\Sigma_{xB_{p-6}}^{n-k} - d_t^{(3)}\Sigma_{B_{p-6}}^{n-k} \end{bmatrix}.$$

2.2.4 Cubic two-phase regression models with smoothness constraints: M33c

Under smoothness constraints $\beta_{1,-} = \beta_{1,+}$ and $\beta_{2,-} = \beta_{2,+}$ on M33, we obtain the following mean function of M33c:

$$\eta(x, \boldsymbol{z}) = \alpha_1 + \boldsymbol{\alpha_2}^T \boldsymbol{z} + \beta_2 (x-e)^2 + \beta_{3,-}(x-e)_-^3 + \beta_{3,+}(x-e)_+^3 + \gamma x. \tag{2.8}$$

The formulas in Section 2.1.4 are simplified as

$$\boldsymbol{W}_1^{**} = \begin{bmatrix} c_7 & c_{14} & c_8 & c_{11} \\ c_{14} & c_{15} & c_9 & c_{12} \\ c_8 & c_9 & c_{10} & 0 \\ c_{11} & c_{12} & 0 & c_{13} \end{bmatrix},$$

where, with $\Sigma^n_{x^2} = \sum_{i=1}^n x_i^2$ and $\Sigma^n_{x^3} = \sum_{i=1}^n x_i^3$,

$$
\begin{aligned}
c_{14} &= -3d_t^{(1)}\Sigma^n_{x^2} + (4d_t^{(1)}e_t + d_t^{(2)} + 2(d_t^{(1)})^2)\Sigma^n_x - nd_t^{(1)}e_t^2 - nd_t^{(2)}e_t - nd_t^{(1)}d_t^{(2)}, \\
c_{15} &= -4d_t^{(1)}\Sigma^n_{x^3} + (8d_t^{(1)}e_t + 2d_t^{(2)} + 4(d_t^{(1)})^2)\Sigma^n_{x^2} + (-4d_t^{(1)}e_t^2 - 4d_t^{(2)}e_t - 4d_t^{(1)}d_t^{(2)})\Sigma^n_x \\
&\quad + 2nd_t^{(2)}e_t^2 + n(d_t^{(2)})^2,
\end{aligned}
$$

and with $\Sigma^n_{xr} = \sum_{i=1}^n x_i r_i$,

$$
\boldsymbol{w}^{**} = \begin{bmatrix} -d_t^{(1)}\Sigma^n_r \\ -2d_t^{(1)}\Sigma^n_{xr} + d_t^{(2)}\Sigma^n_r \\ -3d_t^{(1)}\Sigma^k_{x^2r} + 3d_t^{(2)}\Sigma^k_{xr} - d_t^{(3)}\Sigma^k_r \\ -3d_t^{(1)}\Sigma^{n-k}_{x^2r} + 3d_t^{(2)}\Sigma^{n-k}_{xr} - d_t^{(3)}\Sigma^{n-k}_r \end{bmatrix},
$$

and with $\Sigma^n_{xB_1} = \sum_{i=1}^n x_i B_{i,1}$,

$$
\boldsymbol{W}_2^{**} = \begin{bmatrix} -d_t^{(1)}\Sigma^n_{B_1} & , \cdots , & -d_t^{(1)}\Sigma^n_{B_{p-4}} \\ -2d_t^{(1)}\Sigma^n_{xB_1} + d_t^{(2)}\Sigma^n_{B_1} & , \cdots , & -2d_t^{(1)}\Sigma^n_{xB_{p-4}} + d_t^{(2)}\Sigma^n_{B_{p-4}} \\ -3d_t^{(1)}\Sigma^k_{x^2B_1} + 3d_t^{(2)}\Sigma^k_{xB_1} - d_t^{(3)}\Sigma^k_{B_1} & , \cdots , & -3d_t^{(1)}\Sigma^k_{x^2B_{p-4}} + 3d_t^{(2)}\Sigma^k_{xB_{p-4}} - d_t^{(3)}\Sigma^k_{B_{p-4}} \\ -3d_t^{(1)}\Sigma^{n-k}_{x^2B_1} + 3d_t^{(2)}\Sigma^{n-k}_{xB_1} - d_t^{(3)}\Sigma^{n-k}_{B_1} & , \cdots , & -3d_t^{(1)}\Sigma^{n-k}_{x^2B_{p-4}} + 3d_t^{(2)}\Sigma^{n-k}_{xB_{p-4}} - d_t^{(3)}\Sigma^{n-k}_{B_{p-4}} \end{bmatrix}.
$$

M33c is of particular interest because of its similarity to cubic spline models. According to Wakefield (2013), the cubic spline function with L knots ($\xi_1 < \cdots < \xi_L$) is defined as

$$
f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \sum_{l=1}^{L} b_l (x - \xi_l)_+^3. \tag{2.9}
$$

In both equations (2.8) and (2.9), the first and second derivatives are continuous, particularly at the knots (or the threshold). The difference is that compared to M33c, the cubic spline model has more parameters to estimate, including multiple knots, when $L > 1$. In addition, M33c avoids the need for choosing the optimal number and position of the knots as in the cubic spline model, since the threshold is directly estimated from the data in M33c based on the fast grid search algorithm.

Chapter 3

SIMULATION STUDIES

To study the behavior of the estimators and bootstrap confidence intervals obtained from the proposed fast grid search algorithm in finite samples, we conduct Monte Carlo (MC) experiments. Bias and coverage of estimation are calculated in two simulation settings: under correctly specified models and under misspecified models.

3.1 Bias and coverage when models are correctly specified

In this subsection, we specifically focus on correctly specified higher-order two-phase regression models with known parameters. The mean functions of the thirteen models we consider are given below with the true parameter values. These are also described in Figure 1.1.

$$\begin{aligned}\eta &= \alpha_1 + \boldsymbol{\alpha_2^T z} + \beta_1 (x-e)_- + \beta_2 (x-e)_-^2 \\ &= \log(1.4)z + 10\,(x-5)_- + (x-5)_-^2 \end{aligned} \tag{M20}$$

$$\begin{aligned}\eta &= \alpha_1 + \boldsymbol{\alpha_2^T z} + \beta_1 (x-e)_- + \beta_2 (x-e)_-^2 + \beta_3 (x-e)_-^3 \\ &= \log(1.4)z + 10\,(x-5)_- + 2\,(x-5)_-^2 - (x-5)_-^3 \end{aligned} \tag{M30}$$

$$\begin{aligned}\eta &= \alpha_1 + \boldsymbol{\alpha_2^T z} + \gamma_1 x + \beta_1 (x-e)_- + \beta_2 (x-e)_-^2 \\ &= \log(1.4)z + 5x + 5\,(x-5)_- + (x-5)_-^2 \end{aligned} \tag{M21}$$

$$\begin{aligned}\eta &= \alpha_1 + \boldsymbol{\alpha_2^T z} + \gamma_1 x + \beta_2 (x-e)_-^2 \\ &= \log(1.4)z + 5x + (x-5)_-^2 \end{aligned} \tag{M21c}$$

$$\begin{aligned}\eta &= \alpha_1 + \boldsymbol{\alpha_2^T z} + \gamma_1 x + \beta_1 (x-e)_- + \beta_2 (x-e)_-^2 + \beta_3 (x-e)_-^3 \\ &= \log(1.4)z + 2x + 8\,(x-5)_- + 3\,(x-5)_-^2 - (x-5)_-^3 \end{aligned} \tag{M31}$$

$$\begin{aligned}\eta &= \alpha_1 + \boldsymbol{\alpha_2^T z} + \beta_1 (x-e)_+ + \beta_2 (x-e)_+^2 \\ &= \log(1.4)z + 10\,(x-5)_+ + 10\,(x-5)_+^2 \end{aligned} \tag{M02}$$

$$\eta = \alpha_1 + \boldsymbol{\alpha_2^T z} + \beta_1 (x-e)_+ + \beta_2 (x-e)_+^2 + \beta_3 (x-e)_+^3$$
$$= \log(1.4)z - 5(x-5)_+ - 2(x-5)_+^2 + (x-5)_+^3 \quad \text{(M03)}$$

$$\eta = \alpha_1 + \boldsymbol{\alpha_2^T z} + \gamma_1 x + \beta_1 (x-e)_+ + \beta_2 (x-e)_+^2$$
$$= \log(1.4)z + 3x + 7(x-5)_+ + 10(x-5)_+^2 \quad \text{(M12)}$$

$$\eta = \alpha_1 + \boldsymbol{\alpha_2^T z} + \gamma_1 x + \beta_2 (x-e)_+^2$$
$$= \log(1.4)z + 3x + 10(x-5)_+^2 \quad \text{(M12c)}$$

$$\eta = \alpha_1 + \boldsymbol{\alpha_2^T z} + \gamma_1 x + \beta_1 (x-e)_+ + \beta_2 (x-e)_+^2 + \beta_3 (x-e)_+^3$$
$$= \log(1.4)z - 2x - 2(x-5)_+ - 2(x-5)_+^2 + 2(x-5)_+^3 \quad \text{(M13)}$$

$$\eta = \alpha_1 + \boldsymbol{\alpha_2^T z} + \beta_{1,-} (x-e)_- + \beta_{1,+} (x-e)_+ + \beta_{2,-} (x-e)_-^2 + \beta_{2,+} (x-e)_+^2$$
$$= \log(1.4)z + 10(x-5)_- + (x-5)_+ + (x-5)_-^2 + (x-5)_+^2 \quad \text{(M22)}$$

$$\eta = \alpha_1 + \boldsymbol{\alpha_2^T z} + \gamma_1 x + \beta_{2,-} (x-e)_-^2 + \beta_{2,+} (x-e)_+^2$$
$$= \log(1.4)z + 10x + (x-5)_-^2 + 10(x-5)_+^2 \quad \text{(M22c)}$$

$$\eta = \alpha_1 + \boldsymbol{\alpha_2^T z} + \gamma_1 x + \gamma_2 (x-e)^2 + \beta_{3,-} (x-e)_-^3 + \beta_{3,+} (x-e)_+^3$$
$$= \log(1.4)z + 10x + (x-5)^2 - 5(x-5)_-^3 + 5(x-5)_+^3 \quad \text{(M33c)}$$

Here, for all models, the intercept α_1 is zero, the coefficient α_2 of the one-dimensional adjusted covariate z is $\log(1.4) = 0.34$, z is standard normally distributed and the covariate of interest x is uniformly distributed between 1.5 and 7.9. Then the outcome for each model is generated by $y = \eta + \epsilon$, where $\epsilon \sim N(0, \sigma = 3)$, assuming homoscedastic errors.

To obtain confidence intervals (CIs), we use a bootstrap procedure. Specifically, we consider three types of bootstrap CIs, i.e., percentile, basic (Efron and Tibshirani, 1993) and symmetric percentile (Hansen, 2017), and compare their performances. The percentile bootstrap CI is an intuitive one, constructed by taking the $100\alpha/2$-th and $100(1-\alpha/2)$-th percentiles of the bootstrap distribution of the estimate $\hat{\theta}$ as the $100(1-\alpha)\%$ confidence limits. The basic (aka, inverse percentile) bootstrap CI is computed as $[2\hat{\theta} - \hat{G}^{-1}(1-\alpha/2), 2\hat{\theta} - \hat{G}^{-1}(\alpha/2)]$, where $\hat{G}$ is the bootstrap cumulative distribution of the estimate $\hat{\theta}$ and thus $\hat{G}^{-1}(1-\alpha/2)$ and $\hat{G}^{-1}(\alpha/2)$ are the $100(1-\alpha/2)$-th and $100\alpha/2$-th percentiles of the bootstrap distribution. These two CIs may perform badly for

skewed bootstrap distributions. In such case, the symmetric percentile bootstrap CI can be a good alternative, which is computed as $[\hat{\theta} - q^*_{1-\alpha}, \hat{\theta} + q^*_{1-\alpha}]$, where $q^*_{1-\alpha}$ is the $(1-\alpha)$ quantile of $|\hat{\theta} - \hat{\theta}^*|$ and $\hat{\theta}^*$ is the bootstrap estimate.

For each model listed above, 10,000 MC simulations are conducted. In each MC simulation, 1,000 bootstrap samples are generated to compute the three 95% bootstrap CIs. Estimates and CIs are obtained for four sample sizes: 100, 200, 500 and 2000. We report the mean parameter estimates from the MC simulations along with their biases. The bias is calculated by subtracting the MC mean from the true parameter values. In addition, empirical coverage of the three 95% bootstrap CIs for the parameters and their MC median width are provided.

Simulation results are given in Tables 3.1 - 3.7. What we observe from all the thirteen models is that as the sample size n increases, the bias is reduced and the empirical coverage approaches the nominal coverage of 95%, with a higher precision represented by a narrower MC median width. Further, the estimated coefficient $\hat{\alpha}_2$ of the adjusted covariate is always well behaved in terms of bias and coverage: it has almost no bias and has coverage close to the nominal coverage even for the small sample size. Among the three bootstrap CIs, the percentile bootstrap CI performs best for all models with respect to coverage. The symmetric percentile bootstrap CI is second-best in that it is slightly wider than the percentile bootstrap CI. The basic bootstrap CI performs worst, tending to grossly undercover the true threshold parameter and the coefficients under no smoothness constraints.

The finite-sample behavior of the estimated parameters related to threshold effects depends on the shape of the two-phase regression model. To illustrate the effect of different model shapes, consider the simplest examples in our simulation: M20 and M02 in Figure 1.1. Table 3.1 shows that estimating M02 is much harder than estimating M20, as indicated by a larger bias, a lower coverage (for small samples), and a wider CI for β_1 and e. Notice that in Figure 1.1, a gradual transition occurs in M02 from the flat line to the quadratic curve near the neighborhood of the threshold e; uncertainty arises as to where exactly the transition occurs in this case, and thus estimation of e is harder than in M20. On the other hand, a sudden transition near e in M20 allows a relatively easy estimation of e. This also explains the reason for a large bias and a wide bootstrap CI for β_1, since β_1 is the right derivative of M02 at $x = e$ (denoted by $\partial_+\eta(e)$), or equivalently, is the left derivative of M20 at $x = e$ (denoted by $\partial_-\eta(e)$). The same argument can be applied to the M21

and M12 examples in Figure 1.1: M12 is harder to estimate than M21, particularly for β_1 and e due to a gradual change around e (Table 3.2).

The cubic two-phase regression model exhibits similar behavior as the quadratic two-phase regression model in Figure 1.1, in that M03 (or M13) is harder to estimate than M30 (or M31). In M03 and M13, the biases of the estimates of β_1, β_2 and e are still large when n is 5,000 as compared to their counterparts in M30 and M31. The corresponding bootstrap CIs are extremely wide in M03 and M13, reflecting a high level of uncertainty (or less precision) in the estimates. Even worse, the bootstrap CIs for M13 tend to overcover and do not achieve the nominal coverage for almost all parameters in the sample sizes tested. These results are again due to the shape of the model governed by the parameter values. For example, the transition from a cubic curve to a line (or vice versa) is more gradual in M13, as expected from a smaller value of $\beta_1 (= \partial_+\eta(e) - \partial_-\eta(e))$. The rate of change in this transition at $x = e$ is $\partial_+^2\eta(e) = 2\beta_2$ in M13 and $\partial_-^2\eta(e) = 2\beta_2$ in M31. A smaller value of β_2 in M13 indicates that M13 has a slower rate of change in its transition than M31. Moreover, β_3 determines the width of a cubic curve and hence we see that M13 has a narrower curve than M31. All these findings contribute to the observed poorer performance of the estimators for M13 and M03 than that for M31 and M30.

Finally, comparing the higher-order two-phase regression models with and without smoothness constraints (i.e., M22c vs. M22, M21c vs. M21, and M12c vs. M12), the former are much easier to estimate with almost unbiased and precise estimates and good coverage of CIs (Tables 3.3 and 3.4). The smoothness constraint relieves the burden of estimating the coefficients of $(x - e)_+$ and/or $(x - e)_-$, and therefore makes the estimation of the rest of the parameters more stable. The performance of the estimators for M33c is also found to be good as shown in Table 3.7.

Table 3.1: Results from 10,000 MC simulations for M20 and M02. Mean parameter estimates from the MC simulations and their biases are presented in the column labeled "Est(Bias)". Empirical coverage of the three 95% bootstrap CIs for the parameters (with MC median width in parenthesis) is also given. "Perc", "Symm" and "Basic" below refer to the percentile, symmetric percentile and basic bootstrap CI, respectively.

	M20				M02			
n	Est(Bias)	Perc	Symm	Basic	Est(Bias)	Perc	Symm	Basic
α_2								
100	0.34 (0.00)	94.6 (1.20)	94.8 (1.20)	94.5 (1.20)	0.34 (0.00)	94.6 (1.20)	94.6 (1.20)	94.3 (1.20)
200	0.34 (0.00)	94.4 (0.83)	94.4 (0.84)	94.2 (0.83)	0.34 (0.00)	94.3 (0.83)	94.5 (0.84)	94.4 (0.83)
500	0.34 (0.00)	94.2 (0.52)	94.4 (0.53)	94.4 (0.52)	0.34 (0.00)	94.2 (0.53)	94.3 (0.53)	94.2 (0.53)
5000	0.34 (0.00)	94.6 (0.17)	94.6 (0.17)	94.6 (0.17)	0.34 (0.00)	94.4 (0.17)	94.7 (0.17)	94.7 (0.17)
β_1								
100	10.03 (0.03)	96.0 (6.48)	96.9 (6.72)	93.4 (6.48)	5.44 (-4.56)	93.5 (27.88)	90.7 (38.36)	58.0 (27.88)
200	10.02 (0.02)	95.9 (4.30)	96.4 (4.41)	93.4 (4.30)	6.42 (-3.58)	94.9 (24.43)	92.5 (34.74)	67.4 (24.43)
500	10.00 (0.00)	95.7 (2.60)	96.2 (2.64)	94.5 (2.60)	8.16 (-1.84)	95.2 (20.51)	95.2 (30.12)	82.2 (20.51)
5000	10.00 (0.00)	95.1 (0.79)	95.1 (0.79)	94.4 (0.79)	9.99 (-0.01)	94.7 (2.72)	96.1 (2.77)	94.6 (2.72)
β_2								
100	1.03 (0.03)	96.0 (1.97)	96.3 (2.03)	92.0 (1.97)	9.93 (-0.07)	96.8 (2.61)	96.4 (2.76)	85.9 (2.61)
200	1.02 (0.02)	95.8 (1.34)	96.2 (1.37)	93.5 (1.34)	9.99 (-0.01)	97.0 (1.95)	95.5 (2.05)	83.5 (1.95)
500	1.00 (0.00)	95.6 (0.81)	95.9 (0.83)	94.2 (0.81)	10.01 (0.01)	96.2 (1.30)	94.5 (1.34)	85.0 (1.30)
5000	1.00 (0.00)	95.0 (0.25)	95.1 (0.25)	94.4 (0.25)	10.00 (0.00)	95.3 (0.40)	95.8 (0.41)	95.5 (0.40)
e								
100	5.01 (0.01)	95.1 (0.65)	96.3 (0.65)	87.8 (0.65)	4.77 (-0.23)	90.3 (1.16)	90.0 (1.68)	59.9 (1.16)
200	5.00 (0.00)	94.5 (0.42)	96.1 (0.45)	90.7 (0.42)	4.83 (-0.17)	94.8 (1.03)	93.4 (1.54)	64.7 (1.03)
500	5.00 (0.00)	94.7 (0.24)	96.1 (0.26)	92.0 (0.24)	4.91 (-0.09)	95.1 (0.87)	94.8 (1.36)	79.0 (0.87)
5000	5.00 (0.00)	94.8 (0.07)	95.4 (0.07)	93.8 (0.07)	5.00 (0.00)	94.5 (0.08)	95.8 (0.08)	93.8 (0.08)

Table 3.2: Results from 10,000 MC simulations for M21 and M12. Mean parameter estimates from the MC simulations and their biases are presented in the column labeled "Est(Bias)". Empirical coverage of the three 95% bootstrap CIs for the parameters (with MC median width in parenthesis) is also given. "Perc", "Symm" and "Basic" below refer to the percentile, symmetric percentile and basic bootstrap CI, respectively.

	M21				M12			
n	Est(Bias)	Perc	Symm	Basic	Est(Bias)	Perc	Symm	Basic
α_2								
100	0.34 (0.00)	94.9 (1.22)	95.0 (1.23)	94.5 (1.22)	0.34 (0.00)	94.5 (1.20)	94.7 (1.21)	94.5 (1.20)
200	0.34 (0.00)	94.3 (0.84)	94.5 (0.84)	94.2 (0.84)	0.34 (0.00)	94.4 (0.84)	94.4 (0.84)	94.1 (0.84)
500	0.34 (0.00)	94.3 (0.53)	94.6 (0.53)	94.3 (0.53)	0.34 (0.00)	94.2 (0.53)	94.3 (0.53)	94.1 (0.53)
5000	0.34 (0.00)	94.5 (0.17)	94.6 (0.17)	94.5 (0.17)	0.34 (0.00)	94.4 (0.17)	94.6 (0.17)	94.7 (0.17)
γ_1								
100	4.79 (-0.21)	96.9 (4.45)	98.5 (4.26)	93.6 (4.45)	3.15 (0.15)	94.1 (1.98)	96.0 (2.04)	94.3 (1.98)
200	4.92 (-0.08)	96.5 (2.14)	97.7 (2.19)	94.7 (2.14)	3.08 (0.08)	95.0 (1.35)	96.6 (1.40)	94.2 (1.35)
500	4.97 (-0.03)	95.9 (1.06)	96.5 (1.08)	94.0 (1.06)	3.04 (0.04)	95.9 (0.83)	97.1 (0.85)	94.9 (0.83)
5000	5.00 (0.00)	95.2 (0.30)	95.4 (0.30)	94.8 (0.30)	3.00 (0.00)	96.1 (0.24)	96.6 (0.24)	95.5 (0.24)
β_1								
100	5.43 (0.43)	93.5 (16.53)	96.9 (15.10)	96.3 (16.53)	2.56 (-4.44)	94.4 (29.10)	93.2 (37.80)	53.9 (29.10)
200	5.21 (0.21)	95.1 (4.53)	97.0 (4.67)	95.5 (4.53)	2.86 (-4.14)	94.9 (24.42)	93.2 (32.84)	56.2 (24.42)
500	5.06 (0.06)	95.4 (2.61)	95.5 (2.64)	93.9 (2.61)	3.73 (-3.27)	95.2 (19.81)	93.2 (27.95)	62.5 (19.81)
5000	5.00 (0.00)	95.0 (0.82)	95.2 (0.82)	94.9 (0.82)	6.70 (-0.30)	95.2 (9.26)	96.9 (4.91)	92.7 (9.26)
β_2								
100	1.20 (0.20)	98.1 (11.12)	99.0 (8.06)	93.3 (11.12)	9.82 (-0.18)	95.7 (2.65)	96.8 (2.79)	92.1 (2.65)
200	1.10 (0.10)	97.1 (1.94)	98.1 (1.96)	92.8 (1.94)	9.92 (-0.08)	96.4 (1.89)	97.0 (1.99)	91.3 (1.89)
500	1.02 (0.02)	96.3 (0.91)	96.9 (0.93)	93.2 (0.91)	9.99 (-0.01)	97.2 (1.21)	96.5 (1.27)	88.5 (1.21)
5000	1.00 (0.00)	95.4 (0.25)	95.4 (0.26)	94.4 (0.25)	10.00 (0.00)	95.9 (0.42)	95.5 (0.43)	93.2 (0.42)
e								
100	4.99 (-0.01)	97.6 (3.62)	98.2 (4.27)	84.2 (3.62)	4.76 (-0.24)	91.7 (1.23)	91.6 (1.68)	54.7 (1.23)
200	5.01 (0.01)	96.2 (1.80)	97.6 (1.93)	87.8 (1.80)	4.79 (-0.21)	94.3 (1.06)	93.1 (1.48)	54.6 (1.06)
500	5.00 (0.00)	95.0 (0.73)	96.5 (0.80)	88.1 (0.73)	4.84 (-0.16)	94.7 (0.88)	93.5 (1.28)	61.4 (0.88)
5000	5.00 (0.00)	94.2 (0.18)	95.6 (0.19)	91.6 (0.18)	4.99 (-0.01)	94.8 (0.40)	96.4 (0.19)	91.5 (0.40)

Table 3.3: Results from 10,000 MC simulations for M21c and M12c. Mean parameter estimates from the MC simulations and their biases are presented in the column labeled "Est(Bias)". Empirical coverage of the three 95% bootstrap CIs for the parameters (with MC median width in parenthesis) is also given. "Perc", "Symm" and "Basic" below refer to the percentile, symmetric percentile and basic bootstrap CI, respectively.

	M21c				M12c			
n	Est(Bias)	Perc	Symm	Basic	Est(Bias)	Perc	Symm	Basic
α_2								
100	0.34 (0.00)	94.4 (1.19)	94.6 (1.19)	94.3 (1.19)	0.34 (0.00)	94.4 (1.19)	94.7 (1.20)	94.5 (1.19)
200	0.34 (0.00)	94.2 (0.83)	94.2 (0.84)	94.2 (0.83)	0.34 (0.00)	94.3 (0.83)	94.4 (0.83)	94.2 (0.83)
500	0.34 (0.00)	94.3 (0.52)	94.5 (0.53)	94.5 (0.52)	0.34 (0.00)	94.3 (0.52)	94.4 (0.53)	94.2 (0.52)
5000	0.34 (0.00)	94.5 (0.17)	94.6 (0.17)	94.6 (0.17)	0.34 (0.00)	94.4 (0.17)	94.7 (0.17)	94.8 (0.17)
γ_1								
100	5.14 (0.14)	95.6 (2.26)	95.3 (2.40)	85.6 (2.26)	3.00 (0.00)	96.2 (1.50)	96.2 (1.53)	95.1 (1.50)
200	5.08 (0.08)	95.3 (1.61)	96.3 (1.60)	90.9 (1.61)	2.99 (-0.01)	95.7 (1.03)	95.6 (1.04)	94.9 (1.03)
500	5.02 (0.02)	95.2 (0.86)	96.2 (0.87)	94.5 (0.86)	3.00 (0.00)	95.4 (0.64)	95.6 (0.64)	95.2 (0.64)
5000	5.00 (0.00)	94.8 (0.25)	94.9 (0.25)	94.8 (0.25)	3.00 (0.00)	95.1 (0.20)	95.2 (0.20)	95.1 (0.20)
β_2								
100	1.14 (0.14)	95.4 (1.94)	96.3 (2.01)	80.8 (1.94)	10.02 (0.02)	96.8 (2.39)	96.8 (2.44)	95.0 (2.39)
200	1.05 (0.05)	95.3 (1.13)	95.9 (1.18)	85.7 (1.13)	10.01 (0.01)	96.0 (1.60)	95.9 (1.62)	94.7 (1.60)
500	1.02 (0.02)	95.1 (0.66)	95.6 (0.68)	91.3 (0.66)	10.01 (0.01)	95.5 (0.98)	95.4 (0.98)	95.0 (0.98)
5000	1.00 (0.00)	94.9 (0.20)	95.1 (0.20)	94.7 (0.20)	10.00 (0.00)	95.3 (0.30)	95.4 (0.30)	95.3 (0.30)
e								
100	5.12 (0.12)	94.6 (3.17)	93.4 (3.50)	77.9 (3.17)	5.00 (0.00)	96.6 (0.39)	97.3 (0.39)	95.0 (0.39)
200	5.08 (0.08)	94.6 (2.48)	94.2 (2.57)	86.7 (2.48)	5.00 (0.00)	95.7 (0.26)	95.7 (0.26)	94.3 (0.26)
500	5.02 (0.02)	94.7 (1.37)	95.7 (1.39)	93.3 (1.37)	5.00 (0.00)	95.5 (0.15)	95.6 (0.15)	94.9 (0.15)
5000	5.00 (0.00)	94.8 (0.41)	95.0 (0.41)	94.9 (0.41)	5.00 (0.00)	95.3 (0.05)	95.4 (0.05)	95.4 (0.05)

Table 3.4: Results from 10,000 MC simulations for M22 and M22c. Mean parameter estimates from the MC simulations and their biases are presented in the column labeled "Est(Bias)". Empirical coverage of the three 95% bootstrap CIs for the parameters (with MC median width in parenthesis) is also given. "Perc", "Symm" and "Basic" below refer to the percentile, symmetric percentile and basic bootstrap CI, respectively.

	M22				M22c			
n	Est(Bias)	Perc	Symm	Basic	Est(Bias)	Perc	Symm	Basic
α_2								
100	0.34 (0.00)	95.1 (1.24)	95.2 (1.25)	94.5 (1.24)	0.34 (0.00)	94.6 (1.20)	94.7 (1.20)	94.5 (1.20)
200	0.34 (0.00)	94.5 (0.85)	94.7 (0.85)	94.5 (0.85)	0.34 (0.00)	94.2 (0.84)	94.3 (0.84)	94.2 (0.84)
500	0.34 (0.00)	94.3 (0.53)	94.6 (0.53)	94.3 (0.53)	0.34 (0.00)	94.2 (0.53)	94.4 (0.53)	94.4 (0.53)
5000	0.34 (0.00)	94.5 (0.17)	94.6 (0.17)	94.6 (0.17)	0.34 (0.00)	94.6 (0.17)	94.7 (0.17)	94.6 (0.17)
γ_1								
100					9.98 (-0.02)	95.3 (6.39)	95.6 (6.47)	94.1 (6.39)
200					9.99 (-0.01)	95.5 (4.41)	95.5 (4.45)	94.8 (4.41)
500					10.00 (0.00)	94.8 (2.75)	94.8 (2.76)	94.5 (2.75)
5000					10.00 (0.00)	95.0 (0.86)	95.0 (0.87)	94.9 (0.86)
$\beta_{1,-}$								
100	9.40 (-0.60)	98.6 (26.94)	98.6 (30.00)	94.3 (26.94)				
200	9.98 (-0.02)	97.6 (8.26)	98.1 (6.72)	96.2 (8.26)				
500	10.03 (0.03)	96.0 (2.67)	96.6 (2.70)	95.5 (2.67)				
5000	10.00 (0.00)	95.3 (0.79)	95.4 (0.80)	94.9 (0.79)				
$\beta_{1,+}$								
100	0.92 (-0.08)	98.1 (17.26)	98.3 (19.12)	91.2 (17.26)				
200	0.89 (-0.11)	96.7 (9.71)	97.7 (10.25)	95.3 (9.71)				
500	0.92 (-0.08)	96.1 (3.76)	97.2 (3.82)	95.5 (3.76)				
5000	1.00 (0.00)	95.3 (1.08)	95.5 (1.08)	94.8 (1.08)				
$\beta_{2,-}$								
100	0.53 (-0.47)	98.9 (24.01)	99.3 (23.01)	96.5 (24.01)	0.97 (-0.03)	95.1 (1.56)	95.7 (1.57)	94.2 (1.56)
200	0.99 (-0.01)	97.8 (3.22)	98.5 (2.51)	96.8 (3.22)	0.99 (-0.01)	95.1 (1.08)	95.4 (1.09)	94.7 (1.08)
500	1.01 (0.01)	96.3 (0.88)	96.9 (0.90)	94.9 (0.88)	0.99 (-0.01)	94.8 (0.67)	95.0 (0.68)	94.8 (0.67)
5000	1.00 (0.00)	95.2 (0.25)	95.3 (0.25)	94.7 (0.25)	1.00 (0.00)	94.9 (0.21)	95.1 (0.21)	95.0 (0.21)
$\beta_{2,+}$								
100	1.26 (0.26)	98.3 (7.97)	98.8 (6.90)	91.3 (7.97)	10.04 (0.04)	95.1 (2.42)	95.6 (2.45)	94.1 (2.42)
200	1.09 (0.09)	97.0 (2.95)	97.9 (3.08)	94.1 (2.95)	10.02 (0.02)	94.9 (1.67)	95.1 (1.68)	94.2 (1.67)
500	1.04 (0.04)	96.6 (1.44)	97.4 (1.47)	95.1 (1.44)	10.01 (0.01)	95.1 (1.04)	95.5 (1.05)	95.3 (1.04)
5000	1.00 (0.00)	95.5 (0.41)	95.7 (0.41)	94.6 (0.41)	10.00 (0.00)	95.1 (0.33)	95.2 (0.33)	95.1 (0.33)
e								
100	4.87 (-0.13)	97.6 (3.81)	99.1 (4.78)	86.8 (3.81)	5.00 (0.00)	95.2 (0.58)	95.8 (0.65)	93.7 (0.58)
200	4.97 (-0.03)	96.5 (2.32)	98.2 (2.06)	89.8 (2.32)	5.00 (0.00)	95.0 (0.42)	95.1 (0.39)	94.4 (0.42)
500	5.00 (0.00)	95.2 (0.49)	96.8 (0.54)	88.5 (0.49)	5.00 (0.00)	95.3 (0.26)	95.3 (0.26)	95.0 (0.26)
5000	5.00 (0.00)	94.2 (0.12)	95.6 (0.13)	91.5 (0.12)	5.00 (0.00)	95.3 (0.08)	95.4 (0.08)	95.3 (0.08)

Table 3.5: Results from 10,000 MC simulations for M30 and M03. Mean parameter estimates from the MC simulations and their biases are presented in the column labeled "Est(Bias)". Empirical coverage of the three 95% bootstrap CIs for the parameters (with MC median width in parenthesis) is also given. "Perc", "Symm" and "Basic" below refer to the percentile, symmetric percentile and basic bootstrap CI, respectively.

	M30				M03			
n	Est(Bias)	Perc	Symm	Basic	Est(Bias)	Perc	Symm	Basic
α_2								
100	0.34 (0.00)	94.7 (1.21)	95.0 (1.22)	94.4 (1.21)	0.34 (0.00)	94.8 (1.22)	94.9 (1.22)	94.5 (1.22)
200	0.34 (0.00)	94.3 (0.84)	94.4 (0.84)	94.2 (0.84)	0.34 (0.00)	94.2 (0.84)	94.4 (0.84)	94.0 (0.84)
500	0.34 (0.00)	94.3 (0.53)	94.6 (0.53)	94.5 (0.53)	0.34 (0.00)	94.2 (0.53)	94.4 (0.53)	94.1 (0.53)
5000	0.34 (0.00)	94.6 (0.17)	94.6 (0.17)	94.5 (0.17)	0.34 (0.00)	94.4 (0.17)	94.7 (0.17)	94.8 (0.17)
β_1								
100	7.67 (-2.33)	97.3 (26.55)	95.7 (35.85)	74.6 (26.55)	-2.88 (2.12)	98.7 (28.88)	97.0 (36.83)	48.7 (28.88)
200	8.69 (-1.31)	96.6 (21.78)	95.9 (29.31)	84.8 (21.78)	-2.77 (2.23)	98.1 (22.47)	96.2 (28.91)	52.9 (22.47)
500	9.73 (-0.27)	96.2 (12.15)	97.2 (9.35)	93.7 (12.15)	-2.91 (2.09)	96.9 (16.63)	94.6 (21.99)	57.9 (16.63)
5000	10.00 (0.00)	95.3 (2.00)	95.7 (2.03)	94.5 (2.00)	-4.58 (0.42)	95.8 (9.22)	96.3 (11.87)	88.8 (9.22)
β_2								
100	1.02 (-0.98)	97.4 (17.98)	95.6 (22.02)	74.3 (17.98)	-1.59 (0.41)	98.9 (22.88)	97.9 (26.52)	61.5 (22.88)
200	1.38 (-0.62)	96.7 (14.13)	95.8 (17.46)	84.7 (14.13)	-2.24 (-0.24)	98.2 (15.98)	96.6 (18.92)	53.8 (15.98)
500	1.88 (-0.12)	96.2 (9.12)	97.4 (8.42)	94.0 (9.12)	-2.61 (-0.61)	97.2 (10.63)	94.3 (12.71)	57.6 (10.63)
5000	2.00 (0.00)	95.3 (1.76)	95.8 (1.79)	94.2 (1.76)	-2.17 (-0.17)	95.8 (4.93)	96.2 (5.81)	90.0 (4.93)
β_3								
100	-0.95 (0.05)	97.8 (2.25)	96.7 (2.40)	76.8 (2.25)	0.56 (-0.44)	96.3 (5.03)	98.7 (5.37)	94.0 (5.03)
200	-0.99 (0.01)	96.8 (1.58)	95.5 (1.67)	82.5 (1.58)	0.78 (-0.22)	97.1 (3.13)	98.5 (3.39)	92.8 (3.13)
500	-0.99 (0.01)	96.3 (1.01)	96.0 (1.05)	92.1 (1.01)	0.93 (-0.07)	97.6 (1.80)	98.0 (1.95)	88.3 (1.80)
5000	-1.00 (0.00)	95.3 (0.28)	95.8 (0.29)	94.8 (0.28)	1.00 (0.00)	95.9 (0.59)	94.9 (0.61)	87.5 (0.59)
e								
100	5.24 (0.24)	95.5 (2.00)	95.7 (2.85)	68.6 (2.00)	4.58 (-0.42)	96.2 (2.65)	96.0 (3.36)	59.0 (2.65)
200	5.13 (0.13)	93.9 (1.64)	96.0 (2.38)	81.9 (1.64)	4.67 (-0.33)	96.7 (1.96)	95.9 (2.64)	58.8 (1.96)
500	5.03 (0.03)	94.7 (0.86)	97.1 (0.54)	90.7 (0.86)	4.75 (-0.25)	95.4 (1.41)	94.6 (1.95)	60.6 (1.41)
5000	5.00 (0.00)	94.6 (0.10)	95.4 (0.10)	92.7 (0.10)	4.96 (-0.04)	94.9 (0.78)	95.8 (1.07)	86.2 (0.78)

Table 3.6: Results from 10,000 MC simulations for M31 and M13. Mean parameter estimates from the MC simulations and their biases are presented in the column labeled "Est(Bias)". Empirical coverage of the three 95% bootstrap CIs for the parameters (with MC median width in parenthesis) is also given. "Perc", "Symm" and "Basic" below refer to the percentile, symmetric percentile and basic bootstrap CI, respectively.

	M31				M13			
n	Est(Bias)	Perc	Symm	Basic	Est(Bias)	Perc	Symm	Basic
α_2								
100	0.34 (0.00)	95.1 (1.24)	95.2 (1.25)	94.5 (1.24)	0.34 (0.00)	95.1 (1.25)	95.2 (1.26)	94.5 (1.25)
200	0.34 (0.00)	94.3 (0.85)	94.4 (0.85)	94.0 (0.85)	0.34 (0.00)	94.4 (0.85)	94.5 (0.85)	94.2 (0.85)
500	0.34 (0.00)	94.3 (0.53)	94.7 (0.53)	94.5 (0.53)	0.34 (0.00)	94.2 (0.53)	94.5 (0.53)	94.3 (0.53)
5000	0.34 (0.00)	94.6 (0.17)	94.6 (0.17)	94.5 (0.17)	0.34 (0.00)	94.4 (0.17)	94.7 (0.17)	94.7 (0.17)
γ_1								
100	2.37 (0.37)	97.4 (6.15)	98.8 (6.68)	94.0 (6.15)	-2.29 (-0.29)	95.0 (3.21)	98.2 (3.34)	94.6 (3.21)
200	2.12 (0.12)	97.6 (3.48)	98.8 (3.65)	95.3 (3.48)	-2.16 (-0.16)	95.0 (1.86)	97.4 (1.95)	94.1 (1.86)
500	2.00 (0.00)	97.2 (1.40)	98.0 (1.34)	95.9 (1.40)	-2.07 (-0.07)	95.6 (1.01)	97.4 (1.07)	94.9 (1.01)
5000	2.00 (0.00)	95.1 (0.30)	95.4 (0.30)	94.8 (0.30)	-2.01 (-0.01)	96.3 (0.27)	97.0 (0.27)	95.0 (0.27)
β_1								
100	3.53 (-4.47)	97.4 (40.49)	97.3 (53.59)	81.4 (40.49)	3.57 (6.57)	99.2 (69.17)	99.2 (75.44)	79.8 (69.17)
200	6.18 (-1.82)	97.0 (28.39)	97.0 (39.09)	87.5 (28.39)	0.07 (3.07)	99.5 (43.16)	99.3 (45.78)	77.9 (43.16)
500	7.68 (-0.32)	96.3 (16.53)	97.0 (13.96)	94.2 (16.53)	-1.07 (1.93)	99.3 (19.06)	98.5 (24.57)	65.8 (19.06)
5000	8.00 (0.00)	95.3 (1.84)	95.5 (1.85)	94.8 (1.84)	-1.79 (1.21)	96.3 (8.62)	95.0 (12.14)	67.7 (8.62)
β_2								
100	1.41 (-1.59)	98.3 (27.56)	98.3 (31.84)	79.6 (27.56)	-7.24 (-5.24)	99.3 (127.85)	99.7 (135.43)	86.1 (127.85)
200	2.30 (-0.70)	97.9 (18.04)	97.5 (22.47)	83.7 (18.04)	-2.95 (-0.95)	99.6 (61.56)	99.6 (62.08)	79.9 (61.56)
500	2.85 (-0.15)	96.9 (11.79)	97.6 (13.38)	92.9 (11.79)	-2.65 (-0.65)	99.3 (17.90)	98.6 (22.19)	62.5 (17.90)
5000	3.01 (0.01)	95.7 (1.88)	96.2 (1.91)	94.0 (1.88)	-2.80 (-0.80)	96.1 (6.76)	94.1 (9.00)	66.1 (6.76)
β_3								
100	-1.19 (-0.19)	98.7 (8.06)	99.2 (7.98)	91.7 (8.06)	6.45 (4.45)	98.4 (80.20)	99.3 (85.19)	95.7 (80.20)
200	-0.99 (0.01)	98.0 (2.55)	98.5 (2.49)	89.6 (2.55)	2.74 (0.74)	97.6 (33.75)	99.2 (31.88)	96.5 (33.75)
500	-0.98 (0.02)	97.0 (1.10)	96.9 (1.15)	91.6 (1.10)	1.85 (-0.15)	96.5 (2.99)	98.7 (3.22)	96.3 (2.99)
5000	-1.00 (0.00)	95.7 (0.29)	96.0 (0.29)	94.6 (0.29)	1.99 (-0.01)	97.1 (0.56)	96.9 (0.59)	86.0 (0.56)
e								
100	5.21 (0.21)	98.5 (3.36)	98.6 (4.01)	71.1 (3.36)	4.96 (-0.04)	99.5 (3.68)	99.2 (4.91)	55.0 (3.68)
200	5.15 (0.15)	96.9 (2.54)	98.0 (3.28)	82.6 (2.54)	4.85 (-0.15)	99.6 (3.09)	99.2 (3.86)	61.1 (3.09)
500	5.04 (0.04)	95.2 (1.54)	97.7 (1.72)	89.8 (1.54)	4.78 (-0.22)	99.0 (1.87)	97.7 (2.44)	59.2 (1.87)
5000	5.00 (0.00)	94.6 (0.14)	95.8 (0.15)	91.0 (0.14)	4.87 (-0.13)	95.0 (0.85)	94.3 (1.22)	65.7 (0.85)

Table 3.7: Results from 10,000 MC simulations for M33c. Mean parameter estimates from the MC simulations and their biases are presented in the column labeled "Est(Bias)". Empirical coverage of the three 95% bootstrap CIs for the parameters (with MC median width in parenthesis) is also given. "Perc", "Symm" and "Basic" below refer to the percentile, symmetric percentile and basic bootstrap CI, respectively.

	M33c			
n	Est(Bias)	Perc	Symm	Basic
α_2				
100	0.34 (0.00)	94.5 (1.21)	94.8 (1.22)	94.7 (1.21)
200	0.34 (0.00)	94.3 (0.84)	94.3 (0.84)	94.1 (0.84)
500	0.34 (0.00)	94.3 (0.53)	94.5 (0.53)	94.4 (0.53)
5000	0.34 (0.00)	94.5 (0.17)	94.7 (0.17)	94.6 (0.17)
γ_1				
100	9.97 (-0.03)	96.2 (7.77)	96.2 (7.85)	95.1 (7.77)
200	9.97 (-0.03)	96.0 (5.27)	95.9 (5.32)	95.2 (5.27)
500	10.00 (0.00)	95.0 (3.25)	95.1 (3.27)	94.8 (3.25)
5000	10.00 (0.00)	94.4 (1.01)	94.5 (1.02)	94.5 (1.01)
γ_2				
100	1.00 (0.00)	95.2 (2.58)	95.4 (2.60)	95.0 (2.58)
200	1.00 (0.00)	94.9 (1.80)	95.1 (1.81)	94.8 (1.80)
500	0.99 (-0.01)	94.8 (1.14)	95.0 (1.14)	94.8 (1.14)
5000	1.00 (0.00)	94.9 (0.36)	94.9 (0.36)	94.8 (0.36)
$\beta_{3,-}$				
100	-5.01 (-0.01)	95.9 (1.31)	96.1 (1.33)	94.4 (1.31)
200	-5.01 (-0.01)	95.5 (0.90)	95.6 (0.91)	95.0 (0.90)
500	-5.00 (0.00)	95.1 (0.56)	95.2 (0.56)	94.9 (0.56)
5000	-5.00 (0.00)	94.4 (0.18)	94.5 (0.18)	94.5 (0.18)
$\beta_{3,+}$				
100	5.03 (0.03)	96.0 (2.31)	96.3 (2.35)	94.5 (2.31)
200	5.02 (0.02)	95.4 (1.56)	95.7 (1.58)	94.7 (1.56)
500	5.01 (0.01)	94.8 (0.97)	95.2 (0.97)	94.9 (0.97)
5000	5.00 (0.00)	94.5 (0.30)	94.6 (0.30)	94.6 (0.30)
e				
100	5.00 (0.00)	95.7 (0.45)	95.6 (0.52)	94.2 (0.45)
200	5.00 (0.00)	95.4 (0.32)	95.4 (0.32)	94.5 (0.32)
500	5.00 (0.00)	95.0 (0.19)	95.0 (0.18)	94.8 (0.19)
5000	5.00 (0.00)	94.4 (0.06)	94.5 (0.06)	94.5 (0.06)

3.2 Bias and coverage when models are misspecified

We further study the finite sample behavior of the estimators under model misspecification. The impact of model misspecification on parameter estimation is assessed in two scenarios. In the first scenario, the mean of the two-phase regression model is correctly specified but the variance is misspecified. We specifically focus on a situation where the variability of y depends on the magnitude of y itself. That is, the variance of y is greater for smaller values of y, which frequently happens when small y is more susceptible to measurement error. Data are simulated from the following mean function of M20 (same as in Section 3.1) but with heteroscedastic errors inversely proportional to y:

$$\begin{aligned}\eta &= \alpha_1 + \boldsymbol{\alpha_2^T z} + \beta_1 (x-e)_- + \beta_2 (x-e)_-^2 \\ &= \log(1.4)z + 10\,(x-5)_- + (x-5)_-^2\,, \\ \epsilon &\sim N(0, 1.5 + 1.5\sqrt{|\eta|}),\end{aligned} \tag{M20}$$

where $z \sim N(0,1)$ and $x \sim U(1.5, 7.9)$.

Figure 3.1 (left panel) shows the simulated heteroscedastic data and Table 3.8 shows the corresponding MC simulation results. The estimates are found to be robust to variance misspecification. The bias is negligible and the coverage of the percentile bootstrap CI is close to the nominal level for all sample sizes. However, the CIs are wider particularly for the small sample size, as compared to those from M20 with homoscedastic errors (Table 3.1).

In the second scenario, the variance is correctly specified but the mean is misspecified. Data are simulated from the following homoscedastic M33c, but instead of fitting M33c, we fit M22:

$$\begin{aligned}\eta &= \alpha_1 + \boldsymbol{\alpha_2^T z} + \gamma_1 x + \gamma_2 (x-e)^2 + \beta_{3,-} (x-e)_-^3 + \beta_{3,+} (x-e)_+^3 \\ &= 0 \times z + 0 \times x + (x-0)^2 + (x-0)_-^3 - (x-0)_+^3\,, \\ \epsilon &\sim N(0,1),\end{aligned} \tag{M33c}$$

where $x \sim U(-3.2, 3.2)$. The true parameter values of M22 are approximated by averaging parameter estimates from 200 MC simulations using datasets of sample size $n = 100,000$. This,

together with the symmetry of the models (discussed in 2.1.1), yields $\alpha_2 = 0, \beta_{1,-} = -\beta_{1,+} = -6.145101, \beta_{2,-} = \beta_{2,+} = -3.800270$ and $e = 0$. The mean functions of the data-generating model M33c and the fitted model M22 are both illustrated in Figure 3.1 (right panel). The simulation results displayed in Table 3.8 show that our estimates and percentile bootstrap CI are robust to misspecification of the mean. A low bias, good coverage and high precision all indicate their good performance even for the misspecified mean model.

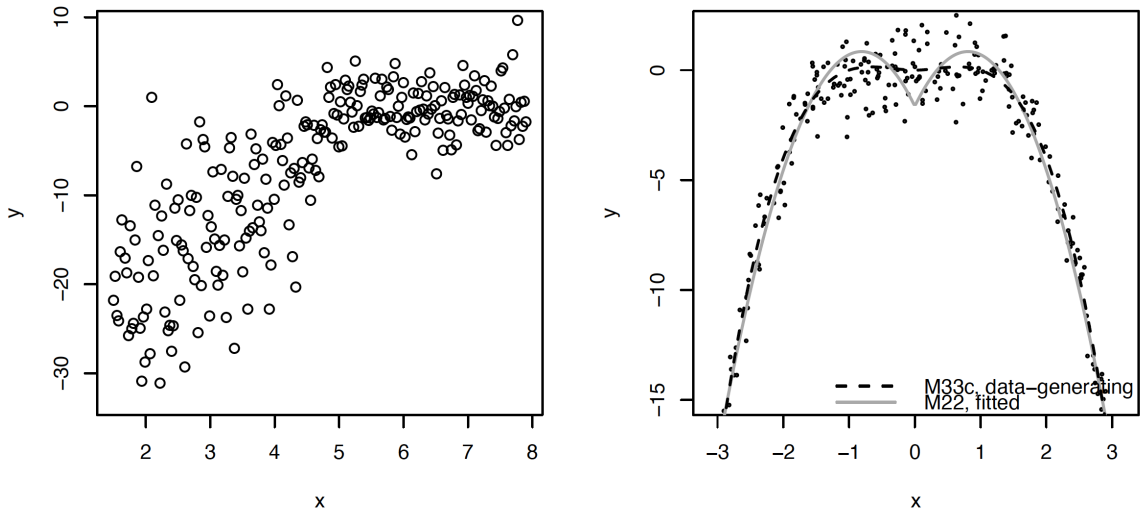

Figure 3.1: Two simulated datasets to evaluate the performance of the estimators under model misspecification. *Left panel*: A dataset generated from the mean function of M20 with heteroscedastic errors inversely proportional to y. The threshold is at $x = 5$. *Right panel*: A dataset generated from the mean function of M33C (dashed line) with homoscedastic errors. M22 (solid line) was fit to this dataset. The threshold is at $x = 0$.

Table 3.8: Results from 10,000 MC simulations for M20 and M22 under model misspecification. Mean parameter estimates from the MC simulations and their biases are presented in the column labeled "Est(Bias)". Empirical coverage of the three 95% bootstrap CIs for the parameters (with MC median width in parenthesis) is also given. "Perc", "Symm" and "Basic" below refer to the percentile, symmetric percentile and basic bootstrap CI, respectively.

	M20				Data-generating: M33c			
	Heteroscedastic errors				Fit: M22			
n	Est(Bias)	Perc	Symm	Basic	Est(Bias)	Perc	Symm	Basic
	α_2				α_2			
100	0.34 (0.01)	94.3 (2.11)	94.8 (2.12)	94.8 (2.11)	0.00 (0.00)	96.2 (0.49)	95.6 (0.50)	93.6 (0.49)
200	0.34 (0.00)	94.2 (1.49)	94.4 (1.49)	94.5 (1.49)	0.00 (0.00)	95.3 (0.34)	95.1 (0.34)	93.7 (0.34)
500	0.34 (0.00)	94.2 (0.94)	94.5 (0.94)	94.5 (0.94)	0.00 (0.00)	95.2 (0.21)	95.0 (0.21)	94.4 (0.21)
5000	0.34 (0.00)	94.5 (0.30)	94.6 (0.30)	94.5 (0.30)	0.00 (0.00)	95.0 (0.07)	95.1 (0.07)	94.9 (0.07)
	β_1				$\beta_{1,-}$			
100	10.48 (0.48)	95.1 (15.24)	96.6 (15.72)	95.0 (15.24)	-6.20 (-0.06)	96.3 (2.52)	96.0 (2.58)	92.2 (2.52)
200	10.23 (0.23)	95.2 (9.64)	97.5 (9.73)	97.0 (9.64)	-6.21 (-0.06)	95.5 (1.76)	95.3 (1.79)	92.9 (1.76)
500	10.06 (0.06)	95.3 (5.56)	96.2 (5.61)	95.7 (5.56)	-6.18 (-0.04)	94.8 (1.08)	95.0 (1.10)	93.7 (1.08)
5000	10.00 (0.00)	94.7 (1.67)	95.1 (1.68)	95.1 (1.67)	-6.15 (0.00)	95.4 (0.34)	95.6 (0.34)	95.1 (0.34)
					$\beta_{1,+}$			
					6.20 (0.05)	96.7 (2.54)	96.4 (2.61)	92.0 (2.54)
					6.21 (0.06)	95.6 (1.75)	95.5 (1.78)	93.4 (1.75)
					6.18 (0.04)	95.3 (1.09)	95.5 (1.11)	94.3 (1.09)
					6.15 (0.00)	95.3 (0.34)	95.3 (0.35)	94.9 (0.34)
	β_2				$\beta_{2,-}$			
100	1.24 (0.24)	94.7 (4.97)	95.9 (5.01)	89.4 (4.97)	-3.85 (-0.05)	96.9 (1.97)	96.8 (2.16)	82.3 (1.97)
200	1.10 (0.10)	95.0 (3.15)	96.0 (3.17)	93.8 (3.15)	-3.83 (-0.03)	96.8 (1.38)	96.6 (1.49)	83.4 (1.38)
500	1.03 (0.03)	94.9 (1.86)	95.7 (1.87)	94.7 (1.86)	-3.82 (-0.02)	96.0 (0.87)	95.3 (0.94)	83.0 (0.87)
5000	1.00 (0.00)	95.0 (0.56)	95.3 (0.56)	95.2 (0.56)	-3.80 (0.00)	95.5 (0.30)	94.8 (0.31)	86.4 (0.30)
					$\beta_{2,+}$			
					-3.85 (-0.05)	96.9 (1.98)	97.0 (2.18)	82.4 (1.98)
					-3.84 (-0.04)	97.2 (1.37)	96.7 (1.49)	83.1 (1.37)
					-3.82 (-0.02)	95.9 (0.87)	95.3 (0.94)	82.6 (0.87)
					-3.80 (0.00)	95.6 (0.30)	94.9 (0.31)	87.2 (0.30)
	e				e			
100	5.00 (0.00)	94.8 (1.23)	96.7 (1.29)	91.6 (1.23)	0.00 (0.00)	95.8 (0.82)	96.2 (0.95)	73.9 (0.82)
200	4.99 (-0.01)	94.9 (0.71)	97.5 (0.77)	94.7 (0.71)	0.00 (0.00)	96.2 (0.58)	96.0 (0.66)	75.8 (0.58)
500	5.00 (0.00)	94.9 (0.40)	96.8 (0.41)	95.1 (0.40)	0.00 (0.00)	95.6 (0.38)	94.8 (0.43)	76.3 (0.38)
5000	5.00 (0.00)	94.9 (0.11)	95.7 (0.11)	95.4 (0.11)	0.00 (0.00)	94.9 (0.13)	94.7 (0.14)	83.6 (0.13)

Chapter 4

APPLICATION TO LIDAR DATA

In this chapter, we apply two-phase regression models to the LIDAR data using our fast grid search algorithm and demonstrate their practical utility. Light detection and ranging (LIDAR) is a remote sensing tool that measures the range (or distance) between the source and a target by measuring the return time of a laser pulse emitted from the source. Air pollution monitoring is one of the most popular applications of LIDAR (e.g., Sigrist, 1994; Ragnarson, 1994). Using the LIDAR technique, Holst et al. (1996) measured the concentration of atmospheric mercury in the plume from the geothermal power plant in Italy. The LIDAR data we analyze here consist of 221 observations from their LIDAR experiment. The range (in meters), which is defined above, is the covariate of interest. The outcome is the logarithm of the ratio of reflected laser pulses from two laser sources: one on the resonance frequency of mercury and the other off resonance. Other adjustment variables are not considered since the data only contain these two variables.

Figure 4.1 shows the association between the log ratio and range. Due to a marked nonlinear relationship observed between the two, the LIDAR data have appeared in the literature regarding polynomial regression models and smoothing splines (Ruppert, 1997; Ruppert et al., 2003; Wakefield, 2013). The data also suggest existence of a threshold around the range of 550 m, after which the association between the log ratio and range appears to be strongly negative. Thus two-phase regression models are well-suited to describing this nonlinear relationship present in the LIDAR data. We specifically explore the following six models: M01, M02, M03, M11, M12 and M13. The first three models are chosen because Figure 4.1 shows no sign of association before the threshold. Moreover, higher-order polynomials, such as quadratic and cubic, are considered for the association after the threshold because of the observed nonlinearity. The last three models are chosen to explicitly test the association before the threshold. We use the percentile bootstrap confidence interval (CI) for inference, which showed the best performance in our simulation study.

The parameter estimates and the corresponding percentile bootstrap CIs from the six fitted

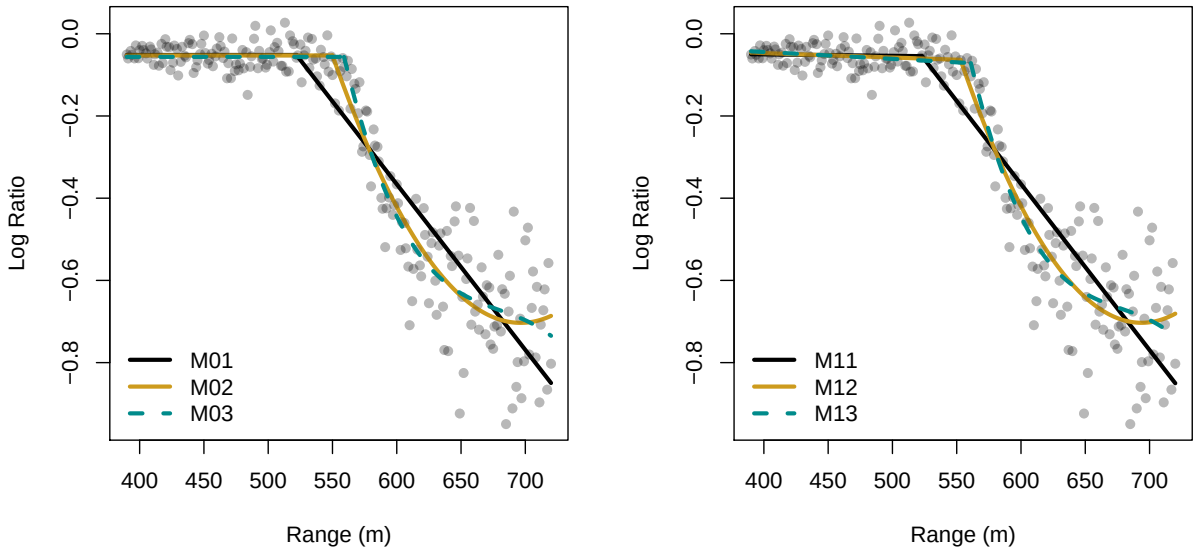

Figure 4.1: The LIDAR data. *Left panel*: Fitted curves from M01, M02 and M03. *Right panel*: Fitted curves from M11, M12 and M13.

models are summarized in Table 4.1 and the model fits are illustrated in Figure 4.1. As expected from Figure 4.1, we find no evidence of an association between the log ratio and range before the threshold. The 95% percentile bootstrap CIs for the range slope in M11, M12 and M13 contain zero, indicating a non-significant linear association before the threshold. The results also suggest a quadratic relationship between the log ratio and range after the threshold, as the 95% percentile bootstrap CI for $(\text{range} - e)_+^2$ in M02 does not contain zero. However, the 95% percentile bootstrap CI for $(\text{range} - e)_+^3$ in M03 does contain zero, which in turn implies that a cubic relationship after the threshold is not well established.

In conclusion, our analyses suggest that M02 best describes the nonlinear relationship between the log ratio and range apparent in the LIDAR data. The threshold is estimated to be 550 m with the 95% percentile bootstrap CI of 544 - 558 m. As a final note, it should be mentioned that we rely on the significance of individual coefficients in comparing the models instead of formal model selection procedures (which need further study). Still, it offers a meaningful insight on the important features of the data as an exploratory analysis.

Table 4.1: Estimation results from the six two-phase regression models for the LIDAR data. The estimated coefficients of the variables, estimated threshold and corresponding 95% percentile bootstrap confidence intervals (in parentheses) are presented. All coefficient estimates and their confidence intervals are multiplied by 10^3 for readability.

	M01	M02	M03	M11	M12	M13
intercept	-51.6	-52.7	-56.2	-36.0	-4.4	22.6
	(-58.9, -45.4)	(-60.7, -46.9)	(-62.9, -49.6)	(-106.7, 31.8)	(-85.9, 69.8)	(-53.2, 80.5)
range				-0.035	-0.11	-0.17
				(-0.189, 0.123)	(-0.27, 0.08)	(-0.30, 0.005)
$(\text{range-}e)_+$	-4.0	-8.9	-13.0	-4.0	-9.0	-13.0
	(-4.4, -3.7)	(-10.5, -7.6)	(-16.0, -8.9)	(-4.4, -3.6)	(-10.3, -7.5)	(-15.6, -9.3)
$(\text{range-}e)_+^2$		0.030	0.099		0.032	0.103
		(0.022, 0.043)	(0.029, 0.153)		(0.021, 0.043)	(0.033, 0.155)
$(\text{range-}e)_+^3$			-0.0003			-0.0003
			(-0.0005, 0.00002)			(-0.0005, -0.00001)
e	522	550	559	523	553	561
	(514, 531)	(544, 558)	(549, 564)	(513, 531)	(546, 559)	(550, 565)

Chapter 5

DISCUSSION

In this thesis, we developed a fast grid search algorithm for estimating parameters of higher-order two-phase regression models. The key idea of our algorithm is to reduce the search time for the threshold value giving the maximum likelihood, based on the recursive formulas sequentially updating the likelihood for each threshold value. This avoids computing the likelihood from scratch each time and thus ease the computational burden. The algorithms for the two-phase regression models presented in Figure 1.1 have been implemented in the R package "chngpt" (Fong et al., 2017a).

Our simulation studies showed that the proposed fast grid algorithm to find the MLE performed satisfactorily, and that the shape (i.e., parameter values) of the model affects the bias and precision of the estimators. Moreover, estimates and bootstrap-based confidence intervals were robust to either misspecified mean or variance models. We concluded with the application of our estimation method to the LIDAR data, which illustrated how the two-phase regression model could be used to describe a nonlinear relationship.

One limitation of our work is that in the grid search, we selected the observed values of the covariate of interest as candidate values for the threshold. We cannot exclude the possibility that the threshold value maximizing the likelihood lies between the observed values or beyond the range of the grid. Though, with increasing sample sizes, the grid becomes denser and this usually provides a good balance between computational complexity and statistical performance. Highly accurate solutions can also be obtained using much denser grids than the observed values.

Our work can be extended in several directions. We only focused on two-phase regression models with an identity link function, but other link functions can be considered as well (e.g., two-phase logistic regression models with a logit link function for binary data and two-phase Poisson regression models with a log link for count data). Estimation methods for such models can be developed in future studies to accommodate non-normal data in a generalized linear model (GLM)

framework (McCullagh and Nelder, 1989). Previously, Elder and Fong (2019) discussed that the fast grid search algorithm is only available for two-phase regression models with an identity link and thus estimation for other link functions should be done in a different way. They used the smooth approximation (Fong et al., 2017b) to estimate the parameters and relied on asymptotic theory to construct confidence intervals. A similar approach can be employed to higher-order two-phase GLMs or a novel grid search method can be developed in future work for practical use.

Also, a further extension can be made by allowing random effects. If some of the parameters in the two-phase regression model are considered random and we are interested in the variability of those random effects as well as fixed effects, two-phase GLMMs (generalized linear mixed models; e.g., McCulloch (2001)) will be appropriate for describing both effects. A vast amount of future work is necessary to develop methods for estimating the two-phase GLMM, because the presence of variance components complicates the problem.

Finally, this thesis has dealt with the estimation of higher-order two-phase regression models, but not much attention has been paid to inference or hypothesis testing. We tested individual parameters based on the bootstrap confidence interval in Chapter 4, which was sufficient for illustration purposes. However, a more formal approach to model testing and selection (e.g., likelihood ratio test) should be introduced and implemented in future studies, in order to guide practitioners in choosing the best model.

BIBLIOGRAPHY

Efron, B. and Tibshirani, R.J. (1993), *An Introduction to the Bootstrap*, Chapman and Hall, New York, NY.

Elder, A. and Fong, Y. (2019), "Estimation and Inference for Upper Hinge Regression Models," *Environmental and Ecological Statistics*, 26, 287–302.

Fong, Y. (2019), "Fast Bootstrap Confidence Intervals for Continuous Threshold Linear Regression," *Journal of Computational and Graphical Statistics*, 28, 466–470.

Fong, Y., Huang, Y., Gilbert, P. and Permar, S. (2017a), "chngpt: threshold regression model estimation and inference," *BMC Bioinformatics*, 18, 454–460.

Fong, Y., Chong, D., Huang, Y. and Gilbert, P. (2017b), "Model-robust Inference for Continuous Threshold Regression Models," *Biometrics*, 73, 452–462.

Gallant, A.R. and Fuller, W.A. (1973), "Fitting segmented polynomial regression models whose join points have to be estimated," *Journal of the American Statistical Association*, 68, 144–147.

Hansen, B.E. (2017), "Regression Kink with an Unknown Threshold," *Journal of Business and Economic Statistics*, 35, 228–240.

Hinkley, D.V. (1971), "Inference in two-phase regression," *Journal of the American Statistical Association*, 66, 736–743.

Holst, U., Hössjer, O., Björklund, C., Ragnarson, P. and Edner, H. (1996), "Locally weighted least squares kernel regression and statistical evaluation of LIDAR measurements," *Environmetrics*, 7, 401–416.

McCullagh, P. and Nelder, J. (1989), *Generalized Linear Models, Second Edition*, Monographs on Statistics and Applied Probability, Taylor & Francis.

McCulloch, C.E. (2001), *Generalized, linear, and mixed models*, Wiley series in probability and statistics., John Wiley & Sons, New York.

Pastor-Barriuso, R., Guallar, E. and Coresh, J. (2003), "Transition models for change-point estimation in logistic regression," *Statistics in Medicine*, 22, 1141–1162.

Ragnarson, P. (1994), "Optical techniques for measurement of atmospheric trace gases," Ph.D. thesis, Department of physics, Lund institute of technology.

Ruppert, D. (1997), "Local polynomial regression and its applications in environmental statistics." in *Statistics for the Environment*, eds. V. Barnett and F. Turkman, vol. 3, Wiley, Chichester.

Ruppert, D., Wand, M.P. and Carroll, R.J. (2003), *Semiparametric Regression*, Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press.

Sigrist, M. (1994), *Air monitoring by spectroscopic techniques*, vol. 127 of *Chemical analysis*, Wiley, New York.

Smith, A. and Cook, D. (1980), "Straight lines with a change-point: A Bayesian analysis of some renal transplant data," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 29, 180–189.

Wakefield, J. (2013), *Bayesian and Frequentist Regression Methods*, Springer Series in Statistics Series, Springer Verlag, New York.

VITA

Hyunju Son graduated from Yonsei University in 2014 with a Bachelor of Arts double major in Economics and Applied Statistics. Following graduation, she entered a graduate program in Statistics at Yonsei University and received a Master's degree in 2016. Soon after, Hyunju came to the U.S. to attend the University of Washington, where she graduated with her Master's degree in Biostatistics in 2020.