Fast Grid Search Algorithms for Multi-phase Regression Models

Qianqian Chen

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

Master of Science

University of Washington

2020

Reading Committee:

Youyi Fong, Chair

Ying Huang

Program Authorized to Offer Degree:
Biostatistics - Public Health

University of Washington

Abstract

Fast Grid Search Algorithms for Multi-phase Regression Models

Qianqian Chen

Chair of the Supervisory Committee:
Associate Professor Youyi Fong
Department of Biostatistics

This thesis focuses on a special case of threshold models called continuous multi-phase regression models, which are characterized by the presence of multiple threshold parameters. This type of methods provides a flexible and interpretable way to model nonlinear relationships with multiple phase changes. We develop a fast grid search algorithm for fitting multi-phase regression models with particular attention to three-phase linear models. The proposed algorithm is shown to have significantly greater computational efficiency compared to the brute force grid search procedure. In addition, the finite sample performance of the three-phase model estimators is investigated through two series of Monte Carlo experiments.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Number Page

LIST OF TABLES

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to Dr. Youyi Fong, for being a wonderful advisor and mentor, for giving me patient guidance and an enjoyable learning experience throughout this journey. I would also like to thank Dr. Ying Huang for joining the committee as my second reader and providing comments and feedback with her great expertise.

I am deeply grateful to my manager, Kate Hu, for being understanding and supportive while I was writing my thesis and working at the same time.

I would like to thank my parents for their constant support of my education; my partner and my friends, for providing me the comfort and encouragement whenever I needed them.

DEDICATION

to my family and friends

Chapter 1
INTRODUCTION

Threshold regression models are a type of nonlinear regression models where the relationships between the outcome and the predictor of interest are allowed to change across a threshold parameter, also known as the change point. This type of models is widely used in many fields, including epidemiology (Ulm, 1991) and human immunology (Permar et al., 2015). Compared with nonparametric smoothing methods such as spline models, threshold regression models provide a simple but interpretable way to address nonlinear problems, and allow estimation and inference to be made on the thresholds. In this thesis we focus on a special case of threshold models - the continuous multi-phase regression models. The distinct feature of this type of models is that the threshold covariate has different regression coefficients in each segment while the regression function is continuous at the thresholds.

Many earlier works on threshold models have been focused on the estimation and inference for single-threshold, or two-phase models (e.g. Fong et al., 2017b; Hansen, 2017). In practice, however, a single threshold parameter may not be capable of capturing multiple structural changes, which are commonly seen in many research areas. Under such circumstances, a reliable and computationally efficient estimation method for multi-phase models is desirable. Building upon the existing algorithm introduced in Elder and Fong (2019), we develop a fast grid search algorithm for fitting multi-phase regression models with particular attention to three-phase linear models. Then we demonstrate the computational efficiency of the proposed algorithm by comparing it with the brute force grid search procedure. Furthermore, the finite

sample performance of the three-phase linear model estimators is investigated through two series of Monte Carlo experiments.

The rest of this thesis is organized as follows: In Chapter 2 we review previous works relevant to this topic. In Chapter 3 we formulate the multi-phase regression model, and develop the fast grid search algorithm for three-phase linear regression models. In Chapter 4 simulation studies are carried out to evaluate the computational and statistical performance of the proposed algorithm. We conclude the thesis with a discussion in Chapter 5.

Chapter 2
LITERATURE REVIEW

In this chapter we review relevant literature on change point modeling, in particular the works concerned with estimating multiple change points. As a preliminary, it may be worth making the distinction between two subclasses of change point problems: change-point problems and threshold problems. One important difference from the point of view of estimation is that in many change-point problems, data are partitioned into subsets by the change points and fitting can be done for each subset separately from other subsets. In other words, the full criterion function can be written as a sum of multiple partial criterion functions. In threshold problems, however, this is usually not the case. While this thesis is primarily focused on threshold problems, here we present methodologies that are pertinent to both subclasses in order to give a broad picture of the existing works under this topic.

2.1 Exact methods based on dynamic programming

2.1.1 Bai et al.'s methods for estimating multiple structural change models

A comprehensive framework for the estimation of multiple change points in linear regression is developed in Bai and Perron (2003). In particular, they presented an algorithm based on the principle of dynamic programming (DP), which efficiently computes the estimates of change points (also called 'structural changes' or 'break points' in their context) as global minimizers of the sum of squared residuals. We describe it in more detail below.

Consider a linear regression model with a known number of break points, say k (i.e. $k+1$ segments). For each k-partition $(T_1, ..., T_k)$, the associated least squares estimates of regression coefficients are obtained by minimizing the sum of squared residuals. Letting $S_T(T_1, ..., T_k)$ denote the resulting sum of squared residuals, then the estimated break points that minimize the objective function over all partitions can be expressed as:

$$(\hat{T}_1, ..., \hat{T}_k) = \underset{T_1,...,T_k}{\arg\min}\, S_T(T_1, ..., T_k).$$

This guarantees that the break point estimators are global minimizers of the objective function, and the regression coefficient estimates are the corresponding least squares estimates at the k-partition.

While the estimated break points could be obtained via an exhaustive grid search, the computation would be of order $O(n^k)$ given a sample size n and is thus burdensome when $k > 2$. In contrast, Bai's approach makes it feasible to obtain the results with at most least-squares operations of order $O(n^2)$ for any number of break points, which is a considerable reduction in computational complexity relative to the brute force grid search procedure. More specifically, the algorithm starts with constructing a triangular matrix of sums of squared residuals corresponding to all possible segments, whose number is at most $n(n+1)/2$. Thus, the estimated break points, i.e. $(\hat{T}_1, ..., \hat{T}_k)$, will necessarily correspond to a particular linear combination of these sums of squared residuals with a minimal value. One can use a dynamic programming algorithm to efficiently evaluate all possible combinations and determine which partition produces a global minimizer of the overall SSR. Such approach essentially proceeds via a sequential search for optimal one-break partitions until a single optimal k-break partition is attained which yields an overall minimum.

The major advantage of Bai's method is its computation, whose bulky part is in the construction of the triangular matrix of SSRs for all possible segments. Aside from

this, the computational costs associated with the sequential search for the optimal k-partition will be trivial compared to the total computational costs, regardless of the number of break points. This means that it is only marginally longer to obtain global minimizers with five or ten break points as it is with two.

As a supplement to the theoretical insights discussed by Bai and Perron, Zeileis et al. (2003) presents how such framework could be applied in practice and illustrates its implementation in the R package *strucchange*. The package incorporates the aforementioned dynamic programming algorithm for computing estimates of break points that minimizes the SSR as well as provides functionalities for hypothesis testing against multiple structural changes in linear regression, thus helping identify locations and presence of break points from the data.

In addition to the least squares estimation of multiple structural changes, the same author has also considered the method of least absolute deviations (LAD) for this estimation problem (Bai, 1998). Specifically, they demonstrate that LAD method could have superior computational feasibility and efficiency over other robust estimation procedures.

First of all, LAD allows complex computations to be done quickly when k is large, which is usually the case for a multiple change points problem. This is because the optimization procedure of LAD can be carried out via linear programming, as studied earlier in Barrodale and Roberts (1974). Second, the change points can be estimated by LAD with greater efficiency relative to least squares if the underlying distribution is heavy-tailed. Efficiency is further improved through LAD's consistent estimation of the regression coefficients. Moreover, this method allows more relaxed assumptions compared to those typically required by existing frameworks. Namely, it does not require each partition to span a positive fraction of the total sample, and allows an unbounded number of change points.

2.1.2 Hawkins' methods for optimal multi-segmentation of data following an exponential family distribution

Hawkins (2001) proposes an exact approach for finding the maximum likelihood estimator (MLE) of a change-point model when its functional form is within the general exponential family. They show that an optimal multi-segment fit can be found through dynamic programming, as a result of the 'separability' property of the change-point models they study. This method is guaranteed to yield the global optimum with a modest amount of computation.

Suppose there is a k-segment model with change points $\tau_1, \tau_2, ..., \tau_{k-1}$ such that the observations X_i $(\tau_{j-1} < i \leq \tau_j)$ follow a particular exponential family distribution with parameter θ_j. In other words, while all segments share the same distributional form, the parameters are allowed to change from one segment to another. Then the log likelihood of the model is given by

$$L(X, \theta, \tau) = \sum_{j=1}^{k} \sum_{i=\tau_{j-1}+1}^{\tau_j} [-\theta_j' X_i + c(X_i) + d(\theta_j)].$$

For arbitrary $0 < h < m \leq n$, the -2 times maximized log likelihood, after dropping the constant term c(.), is derived as

$$Q(h, m) = -2[\hat{\theta}' S(h, m) - (m - h) d(\hat{\theta})],$$

where $S(h, m) = \sum i = h + 1^m X_i$ is the sufficient statistic for θ using the sub-sequence $X_{h+1}, ..., X_m$. Note that the likelihood of change-point models possess an important property that allows Hawkins' method to work - it is separable. Namely, the optimum for dividing cases 1,...,n into k segments consists conceptually of first finding the rightmost change point $\hat{\tau}_k$. After this step, based on the fact that the remaining change points constitute the optimum for splitting cases 1,...,$\hat{\tau}_k$ into $k-1$ segments, they can be found in a recursive way. This was formally introduced by Bellman as

the 'principle of optimality' (Bellman and Dreyfus, 1962). The dynamic programming algorithm by Hawkins can then be outlined as follows:

1. Let $F(r, m)$ denote -2 times maximized log likelihood from fitting an r-segment model to the observations $X_1, ..., X_m$. Calculate $F(1, m) = Q(0, m)$ for each $m = 1, ..., n$.

2. For each subsequent $r = 2, ..., k$, recursively compute the values of $F(r, m)$ using the equation $F(r, m) = \min_{0<h<m}[F(r-1, h) + Q(h, m)]$, $m = 1, ..., n$.

3. Store $H(r, m)$, the value of h that yields the minimum along with each $F(r, m)$ calculated in the step above.

4. Once this recursive procedure is completed, the maximized log likelihood of the resulting k-segment model fitted to the full data set is found as $-\frac{1}{2}F(k, n)$. The estimates of change points $\hat{\tau}_j$ can be solved through DP back-tracing, together with the within-segment $\hat{\theta}_j$.

The computational complexity of this algorithm is $O(kn^2)$. As it has a computational complexity linear in k, it would not require intensive computation when dealing with large numbers of change points. However, the computation time needed to fill up Q may be the time limiting step.

Prior to Hawkins, a similar approach based on the the principle of dynamic programming has been proposed in Bellman and Roth (1969), with special attention to fitting continuous piecewise regression models. Hawkins' DP formulation can be considered as a generalization of Bellman and Roth's algorithm. We do not go into details here as the key idea of these two methods are essentially the same.

2.1.3 Remarks

Through the use of dynamic programming, the aforementioned exact methods are able to find an optimal multi-segmentation of data in a reasonably fast manner. Here

we would like to comment on the applicability of these methods to the multi-phase regression models studied in this thesis. As pointed out by Hawkins, the separability property of change-point models is a prerequisite for such dynamic programming formulation, which leads to improved efficiency of the estimation procedure. On the other hand, this property does not hold for the multi-phase regression models that we are concerned with because in our case different segments are required to share coefficients for covariates not subject to threshold effect.

Furthermore, in the multiple change-point models as described in previous subsections, the data usually have a natural order based on time or space, and the model has a different set of parameters for each regimen divided by the change points along the natural axis. It is possible that the fitted lines might not join at the estimated change points. In addition, the model does not allow a common effect of additional covariates to be shared across regimens. However, in the multi-phase regression models we study, a natural ordering of the data typically does not exist, and this type of regression is mainly used to model nonlinear relationships between the outcome and the predictor. Thus, there is limitation in extending their methodology to the estimation of multi-phase regression models that our thesis is focused on.

2.2 Suboptimal methods - application of greedy algorithms in change point modeling

Greedy algorithm is a commonly used strategy in optimization problems. It breaks one problem into many components, then find the optimal solution for each component in isolation. A major advantage of greedy approach is that it can be computationally much cheaper than algorithms that solve for the global optimum, and the quality of a greedy solution is often acceptable if not optimal. However, combining the individually optimal components is not guaranteed to yield an optimal complete solution.

In the context of change point modeling, this type of methods has been adopted to reduce the computational burden when the number of change points is large. We present two examples here.

2.2.1 Friedman et al.'s adaptive stepwise procedure for piecewise linear fitting

The stepwise procedure proposed in Friedman and Silverman (1989) makes it feasible to fit piecewise linear models with knots that depend on the data, thus providing a simple but flexible approach for modeling nonlinear relationships between the response and explanatory variables. Note that, while there exists terminological difference, the idea of a 'knot' is essentially equivalent to that of a threshold or change point referred to in the thesis.

For a fixed number of knots K, Friedman et al.'s method aims to place the knots and construct the corresponding piecewise linear fit that minimizes the average squared residual (ASR). This is defined by:

$$ASR = \frac{1}{N}\sum_{i=1}^{N}[y_i - f(x_i)]^2,$$

where estimates $f(x)$ are required to be continuous and piecewise linear with the given knots. The curve estimate is then taken to be

$$f(x) = a_0 + \sum_{k=1}^{K} a_k b_k(x),$$

where the values of the coefficients $a_0, ..., a_k$ correspond to the piecewise linear curve that optimizes the ASR. Here, the set of basis functions $b_k(x)$, $1 \leq k \leq K$, is chosen to be $b_k(x) = (x - t_k)^+$, with t_k representing the location of the kth knot and the plus superscript representing the nonnegative part. What makes their choice convenient for computation is that each basis function is parameterized by a single knot. Thus, any operations on the position of a knot would affect only one basis function. Based

on this property, Friedman et al. derived a rapidly computable updating formula, which we describe below.

To start the stepwise procedure for knot placement, the first knot (k = 1) is placed at the position that yields the best corresponding piecewise linear fit. Then each additional knot is placed at the position that yields the best piecewise linear fit involving itself and the $k-1$ previous placed knots. That is, for every k, a least squares fit must be performed at each of the potential new knot positions to obtain the ASR. This process stops when some maximum number of knots are positioned. Assume the knot-placement increment is M, there will be $N/M - k + 1$ potential locations for the kth knot, resulting in approximately N/M least squares fit performed to place each knot (out of K). This would be computationally expensive if done by brute force. To deal with the computational burden, the trick here is to visit the potential knot positions in descending abscissa order and take advantage of an updating formula associated with the aforementioned basis function set $b_k(x)$:

for $t' \geq t''$,

$$(x - t'')^+ - (x - t')^+ = \begin{cases} 0, & x \leq t'' \\ x - t'', & t'' \leq x \leq t' \\ t' - t'', & x > t' \end{cases} \tag{2.1}$$

The linear least squares fit for the kth knot, at $t_k = t''$, can be solved by the normal equations

$$Ba = c, \tag{2.2}$$

where B is the $k \times k$ covariance matrix of the k basis functions, and c is the k-dimensional covariance vector of the response with each basis function, given by

$$B_{jl} = \sum_{i=1}^{N} b_l(x_i)[b_j(x_i) - \bar{b}_j],$$

$$c_j = \sum_{i=1}^{N} (y_i - \overline{y}) b_j(x_i),$$

respectively.

Note that only B_{jk} and c_k $(1 \leq j \leq k)$ need to be computed repeatedly since only the position of the kth knot is changing, and this saves the computation by a factor of k. Then, with the updating formula (2.1), if B_{jk} and c_k have been computed for a knot located at $t_k = t'$, then the corresponding quantities for a knot at $t_k = t''$ $(t' > t'')$ can be done through a simple series of updates. Thus, the corresponding quantities at all potential knot locations could be obtained with total computation of order kN.

Now that all values needed for solving the normal equations (2.2) are ready, the remaining task is to enter these values to perform each linear least squares fit for the approximately N/M potential locations for knot replacement. This is done with the Cholesky decomposition of the basis covariance matrix, which requires computation proportional to k^2.

Following the forward procedure for knot placement, the second half of Friedman et al.'s procedure adopts a backward deletion strategy, where the model found from previous step is subject to a knot deletion process. Here, the term 'generalized cross-validation' (GCV) is brought in as an estimate of future prediction error, which is used as the model-selection criterion to be minimized. Let $d(K)$ be a suitable increasing function representing the number of knots in the fitted model. The GCV score is given by

$$GCV = \frac{1}{N} \sum_{i=1}^{N} [y_i - f(x_i)]^2 / [1 - \frac{d(K)}{N}]^2$$

During the backward stepwise procedure, each of the K knots is deleted and the corresponding $(K-1)$-knot model is fitted. If any of these models results in improvement in GCV, the one with the smallest GCV will be chosen and this corresponding

knot will be permanently deleted. This procedure will stop when the deletion of any remaining knot results in no improvement in GCV. It is used to address the question of model selection among the placed knots.

2.2.2 Regression trees: Recursive binary splitting

The regression trees approach (Breiman et al., 1984) is a well-known example of change-point modeling in data analysis, which essentially addresses the change-point problem for the case of normal means. In Breiman et al.'s implementation, the algorithm starts by finding a single change point that leads to maximized separation between two sub-sequences of the dataset, then fixes this point and applies the same binary splitting to each of the sub-sequences it uncovers. This procedure is repeated recursively until the sub-sequences cannot be usefully subdivided. The resulting algorithm has a computational complexity linear in n and is thus very fast, but it generally does not produce the global optimum if there are more than two change points because the true optimal change points are not necessarily hierarchical. As is often the case with greedy algorithms, this situation leads to a tradeoff between less extensive computations and an approximate solution.

2.3 Gradient-based approximation methods

2.3.1 Approximation through Taylor expansions

The non-smooth and non-convex nature of the likelihood function with respect to the threshold parameter results in difficulties in finding the MLE of a threshold regression model. To overcome this problem, Muggeo (2003) proposes a linear reparameterization technique that approximates the log-likelihood function through Taylor expansions, thus translating the problem into a standard linear framework. The proposed method is implemented in the R package *segmented* (Muggeo, 2008). We briefly

describe it here.

Assume that Y is the response variable, Z is a covariate vector, and X is the threshold variable. A possible parameterization to model multi-phase piecewise linear relationships is given by:

$$Y = \gamma^T Z + \alpha X + \sum_{k=1}^{K} \beta_k (X - \lambda_k)_+ + e, \tag{2.3}$$

where K denote the number of thresholds, and $\lambda \equiv (\lambda_1, \lambda_2, ..., \lambda_K)$ are the locations of the thresholds. γ is the vector of coefficients for Z, α is the slope of X before the appearance of a threshold, β_k is the difference in slopes between kth and $(k+1)$th thresholds, and e is the error term. In estimating the kth threshold, for example, it is possible to approximate the non-differentiable term $\beta_k(X - \lambda_k)_+$ by a first-order Taylor expansion around the initial value $\lambda_k^{(0)}$, provided that the $\lambda_k^{(0)}$ should be close to the true value as much as possible. This can be expressed as:

$$(X - \lambda_k)_+ \approx (X - \lambda_k^{(0)})_+ + (\lambda_k - \lambda_k^{(0)})(-1)I(X > \lambda_k^{(0)}),$$

where $(-1)I(X > \lambda_k^{(0)})$ is the first derivative of $(X - \lambda_k)_+$ at $\lambda_k^{(0)}$.

If we apply the above linearization structure to all change points, equation (2.3) can be approximated by

$$Y = \gamma^T Z + \alpha X + \sum_{k=1}^{K} \beta_k \widetilde{U}_k + \sum_{k=1}^{K} \delta_k \widetilde{V}_k + e,$$

where $\delta_k = \beta_k(\lambda_k - \lambda_k^{(0)})$. $\widetilde{U}_k = (X - \lambda_k^{(0)})_+$, and $\widetilde{V}_k = -I(X > \lambda_k^{(0)})$ are two new covariates with coefficients β_k and δ_k, respectively. The $\hat{\delta}_k$ is called 'working' coefficient in Muggeo and Adelfio (2010), due to the fact that it measures the gap between the two fitted lines (i.e. before and after $\hat{\delta}_k$). At each iteration, a linear model is fitted and the estimate for λ_k is updated via $\hat{\lambda}_k^{(1)} = \hat{\lambda}_k^{(0)} + \hat{\delta}_k/\hat{\beta}_k$. This iterative estimation procedure proceeds until a possible convergence criterion holds, for example, when $\hat{\delta}$ drops below a tolerance level of 10^{-4}.

2.3.2 Remarks

Muggeo's methods bypass the non-smoothness and non-convexity issue by approximating the true parametric form of the threshold model. Since the original problem is reduced to simple iterative fitting of standard linear regression, such approximation methods would not require extensive computational effort even with large sample size n and/or large number of change points k.

On the other hand, this algorithm does not guarantee to find the global optimal solutions because the likelihood function is not always concave and thus local maxima may exist. In addition, the success of this algorithm depends critically on the choice of initial values for the change points. This situation might be mitigated by making visual inspections of the shape of the relationship and running the algorithm with different starting points to examine the sensitivity of results.

Chapter 3

FAST GRID SEARCH ALGORITHMS FOR PIECEWISE LINEAR MULTI-PHASE REGRESSION

3.1 Multi-phase model

In the generalized linear regression framework, the mean function of multi-phase models with two thresholds can be parameterized as follows:

$$\eta(x, z) = \alpha_1 + \alpha_2^T z + \beta_e(x - e)_- + \beta_f(x - f)_- + \gamma x, \tag{3.1}$$

where e and f are the threshold parameters ($e < f$), x is the predictor with threshold effect, and z denotes additional predictors. $(x - e)_- = x - e$ if $x < e$ and 0 otherwise, and $(x - f)_- = x - f$ if $x < f$ and 0 otherwise. The parameters are given by $\theta \equiv (\alpha_1, \alpha_2, \beta_e, \beta_f, e, f, \gamma)$.

According to this parameterization, $(\beta_e + \beta_f + \gamma)$ is the slope of x when $x < e$, $(\beta_f + \gamma)$ is the slope of x when $e \leq x < f$, and γ is the slope of x when $x \geq f$.

3.2 Algorithm development

Our algorithm extends the fast grid search algorithm for upper hinge regression models proposed by Elder and Fong (2019). Their method substantially reduces the computational complexity of the search procedure through the use of a matrix inversion formula when a column is added (Khan, 2008). In this section, we develop the fast grid search algorithm for multi-phase linear regression in the context of three-phase (i.e. two-threshold) models.

Let Y denote the outcome vector and $X_{e,f} \equiv [\mathbb{1}, Z, x, v_e, v_f]$ denote the design

matrix, where v_e and v_f are the vectors $(x - \mathbb{1}e)_-$ and $(x - \mathbb{1}f)_-$, respectively. For a given pair of threshold values (e, f), the log likelihood function under linear regression is inversely proportional to the residual sum of squares (RSS), which can be written as

$$Y^T(I - H_{e,f})Y = Y^TY - Y^TH_{e,f}Y,$$

where I is the identity matrix and $H_{e,f} = X_{e,f}(X_{e,f}^TX_{e,f})^{-1}X_{e,f}^T$ is the hat matrix. Since Y^TY does not change with thresholds (e, f), we can ignore it for this calculation, and it is sufficient to choose the (e, f) that maximizes $Y^TH_{e,f}Y$. Thus, the grid search algorithm for finding the MLE of θ proceeds by computing $Y^TH_{e,f}Y$ on a discrete set of support for (e, f). Here, the candidate threshold values are chosen to be the observed x's with the most extreme values (say, top 5% and bottom 5%) trimmed off for more stable finite sample performance.

Computation of $H_{e,f}$ can become intensive when doing so for every (e, f) in a grid within a bootstrap procedure. To accelerate this computation, we first decompose the design matrix into two parts:

$$X_{e,f} = [X, V_{e,f}],$$

where $X \equiv [\mathbb{1}, Z, x]$ and $V_{e,f} \equiv [v_e, v_f]$. Note that only $V_{e,f}$ involves the thresholds (e, f), whereas X is independent of any threshold effects. After the decomposition, we can apply the matrix inversion formula on $(X_{e,f}^TX_{e,f})^{-1}$ to obtain:

$$\begin{aligned}
(X_{e,f}^TX_{e,f})^{-1} &= \{\begin{bmatrix} X^T \\ V_{e,f}^T \end{bmatrix} \begin{bmatrix} X & V_{e,f} \end{bmatrix}\}^{-1} \\
&= \begin{bmatrix} X^TX & X^TV_{e,f} \\ V_{e,f}^TX & V_{e,f}^TV_{e,f} \end{bmatrix}^{-1} \\
&= c_{e,f}^{-1} \begin{bmatrix} Ac_{e,f} + AX^TV_{e,f}V_{e,f}^TXA^T & -AX^TV_{e,f} \\ -V_{e,f}^TXA^T & 1 \end{bmatrix},
\end{aligned}$$

where $A \equiv (X^TX)^{-1}$, $H \equiv XAX^T$, and $c_{e,f} \equiv V_{e,f}^TV_{e,f} - V_{e,f}^THV_{e,f}$. Then the hat matrix can be written as follows:

$$
\begin{aligned}
H_{e,f} &= X_{e,f}(X_{e,f}^TX_{e,f})^{-1}X_{e,f}^T \\
&= \begin{bmatrix} X & V_{e,f} \end{bmatrix} c_{e,f}^{-1} \begin{bmatrix} Ac_{e,f} + AX^TV_{e,f}V_{e,f}^TXA^T & -AX^TV_{e,f} \\ -V_{e,f}^TXA^T & 1 \end{bmatrix} \begin{bmatrix} X^T \\ V_{e,f}^T \end{bmatrix} \\
&= XAX^T + c_{e,f}^{-1}[XAX^TV_{e,f}V_{e,f}^TXA^TX^T - V_{e,f}V_{e,f}^TXA^TX^T - XAX^TV_{e,f}V_{e,f}^T + V_{e,f}V_{e,f}^T] \\
&= H + c_{e,f}^{-1}[[HV_{e,f}][HV_{e,f}]^T - V_{e,f}V_{e,f}^TH - HV_{e,f}V_{e,f}^T + V_{e,f}V_{e,f}^T] \\
&= H + c_{e,f}^{-1}[HV_{e,f} - V_{e,f}][HV_{e,f} - V_{e,f}]^T.
\end{aligned}
$$

Let $r \equiv [H - I]Y$ and $B_{n\times(p-2)} \equiv X(X^TX)^{-1/2}$. Now we are ready to solve for the criterion function through matrix algebra.

$$
\begin{aligned}
Y^TH_{e,f}Y &= Y^T\{H + c_{e,f}^{-1}[HV_{e,f} - V_{e,f}][HV_{e,f} - V_{e,f}]^T\}Y \\
&= Y^THY + c_{e,f}^{-1}Y^T\{[H - I]V_{e,f}\}\{[H - I]V_{e,f}\}^TY \\
&= Y^THY + c_{e,f}^{-1}r^TV_{e,f}V_{e,f}^Tr \\
&= Y^THY + r^TV_{e,f}(V_{e,f}^TV_{e,f} - V_{e,f}^THV_{e,f})^{-1}V_{e,f}^Tr \\
&= Y^THY + r^TV_{e,f}(V_{e,f}^TV_{e,f} - V_{e,f}^TBB^TV_{e,f})^{-1}V_{e,f}^Tr \qquad (3.2)
\end{aligned}
$$

Comparing with Elder and Fong (2019), we see that equation (3.2) generalizes their results such that $V_{e,f}$ has more than one column.

Note that, when computing $Y^TH_{e,f}Y$, only the second term on the right hand side of equation (3.2) depends on thresholds (e, f), and thus needs to be computed repeatedly for a new candidate threshold value. More specifically, this term depends on three intermediate terms, $V_{e,f}^TV_{e,f}$, $V_{e,f}^Tr$, and $V_{e,f}^TB$. Consider two successive values of the smaller threshold e, e_t and e_{t+1}, which correspond to the kth and the $k+1$th ordered values of x, respectively. In addition, consider two successive values of the larger threshold f, and suppose f_s and f_{s+1} correspond to the jth and the $j+1$th

ordered values of x, where $k \leq j-1$ and $j \leq n-1$. Assume that the rows of the design matrix $X_{e,f}$ are ordered according to the ascending order of x, then we find that

$$V_{e_{t+1},f_s} - V_{e_t,f_s} = [d_t\delta_t, 0],$$

where $d_t \equiv e_{t+1} - e_t$, and δ_t is a vector of size n with the first k entries equal to -1 and the remaining entries equal to 0. Likewise,

$$V_{e_t,f_{s+1}} - V_{e_t,f_s} = [0, d_s\delta_s],$$

where $d_s \equiv f_{s+1} - f_s$, and δ_s is a vector of size n with the first j entries equal to -1 and the remaining entries equal to 0. It follows that

$$\begin{aligned}
V^T_{e_{t+1},f_s}V_{e_{t+1},f_s} &= \begin{bmatrix} v^T_{e_t} + d_t\delta^T_t \\ v^T_{f_s} \end{bmatrix} \begin{bmatrix} v_{e_t} + d_t\delta_t & v_{f_s} \end{bmatrix} \\
&= \begin{bmatrix} v^T_{e_t}v_{e_t} + 2d_tv^T_{e_t}\delta_t + d^2_t\delta^T_t\delta_t & v^T_{e_t}v_{f_s} + d_t\delta^T_t v_{f_s} \\ v^T_{f_s}v_{e_t} + d_tv^T_{f_s}\delta_t & v^T_{f_s}v_{f_s} \end{bmatrix} \\
&= \begin{bmatrix} v^T_{e_t}v_{e_t} - 2d_t\{(\sum_{i=1}^k x_i) - ke_t\} + kd^2_t & v^T_{e_t}v_{f_s} - d_t\{(\sum_{i=1}^k x_i) - kf_s\} \\ v^T_{f_s}v_{e_t} - d_t\{(\sum_{i=1}^k x_i) - kf_s\} & v^T_{f_s}v_{f_s} \end{bmatrix}
\end{aligned} \tag{3.3}$$

and

$$\begin{aligned}
V^T_{e_t,f_{s+1}}V_{e_t,f_{s+1}} &= \begin{bmatrix} v^T_{e_t} \\ v^T_{f_s} + d_s\delta^T_s \end{bmatrix} \begin{bmatrix} v_{e_t} & v_{f_s} + d_s\delta_s \end{bmatrix} \\
&= \begin{bmatrix} v^T_{e_t}v_{e_t} & v^T_{e_t}v_{f_s} + d_sv^T_{e_t}\delta_s \\ v^T_{f_s}v_{e_t} + d_s\delta^T_s v_{e_t} & v^T_{f_s}v_{f_s} + 2d_sv^T_{f_s}\delta_s + d^2_s\delta^T_s\delta_s \end{bmatrix} \\
&= \begin{bmatrix} v^T_{e_t}v_{e_t} & v^T_{e_t}v_{f_s} - d_s\{(\sum_{i=1}^k x_i) - ke_t\} \\ v^T_{f_s}v_{e_t} - d_s\{(\sum_{i=1}^k x_i) - ke_t\} & v^T_{f_s}v_{f_s} - 2d_s\{(\sum_{i=1}^j x_i) - jf_s\} + jd^2_s \end{bmatrix}
\end{aligned} \tag{3.4}$$

Next, in a similar way, we develop formulae for computing the other two intermediate terms, which, together with the one above, allow us to update the second term on the right hand side of equation (3.2) for a new e, f. We find that

$$V^T_{e_{t+1},f_s} r = \begin{bmatrix} v^T_{e_t} r + d_t \delta^T_t r \\ v^T_{f_s} r \end{bmatrix} = \begin{bmatrix} v^T_{e_t} r - d_t(\sum_{i=1}^{k} r_i) \\ v^T_{f_s} r \end{bmatrix}, \tag{3.5}$$

and

$$V^T_{e_t,f_{s+1}} r = \begin{bmatrix} v^T_{e_t} r \\ v^T_{f_s} r + d_s \delta^T_s r \end{bmatrix} = \begin{bmatrix} v^T_{e_t} r \\ v^T_{f_s} r - d_s(\sum_{i=1}^{j} r_i) \end{bmatrix}. \tag{3.6}$$

Lastly,

$$V^T_{e_{t+1},f_s} B = \begin{bmatrix} v^T_{e_t} B + d_t \delta^T_t B \\ v^T_{f_s} B \end{bmatrix} = \begin{bmatrix} v^T_{e_t} B - d_t(\sum_{i=1}^{k} B_{i,1} \ \ldots \ \sum_{i=1}^{k} B_{i,p-2}) \\ v^T_{f_s} B \end{bmatrix}, \tag{3.7}$$

and

$$V^T_{e_t,f_{s+1}} B = \begin{bmatrix} v^T_{e_t} B \\ v^T_{f_s} B + d_s \delta^T_s B \end{bmatrix} = \begin{bmatrix} v^T_{e_t} B \\ v^T_{f_s} B - d_s(\sum_{i=1}^{j} B_{i,1} \ \ldots \ \sum_{i=1}^{j} B_{i,p-2}) \end{bmatrix}, \tag{3.8}$$

where $B_{i,j}$ is the (i,j)th element of the matrix B. Now that the formulae have been derived, we can use these relations to update the likelihood function of the submodel conditional on e_{t+1} and f_{s+1} by reusing the intermediate terms $V^T_{e,f} V_{e,f}$, $V^T_{e,f} r$ and $V^T_{e,f} B$ saved from previous submodels. Taking this shortcut will simplify the computation considerably as it allows sequential updating of $Y^T H_{e,f} Y$, without computing it for every e, f from scratch.

Putting all steps together, the proposed fast grid search algorithm can be expressed as follows:

1. Sort the samples by the ascending order of x_i. Take the re-ordered x's with the most extreme values trimmed off to be the set of candidate threshold values. Let M denote the size of this set.

2. Compute and store the initial values $V_{e_1,f_2}^T V_{e_1,f_2}$, $V_{e_1,f_2}^T r$, and $V_{e_1,f_2}^T B$, where e_1 is the smallest observation among candidate thresholds and f_2 is the second smallest one.

3. Compute and store $Y^T H_{e_1,f_2} Y - Y^T H Y$, which is equivalent to the second term on the right hand side of equation (3.2).

4. For s in 2 to $M-1$:

   (a) update $V_{e_t,f_{s+1}}^T V_{e_t,f_{s+1}}$, $V_{e_t,f_{s+1}}^T r$, and $V_{e_t,f_{s+1}}^T B$ from step (2) and according to the corresponding formulae.

   (b) compute and store $Y^T H_{e_t,f_{s+1}} Y - Y^T H Y$ according to equation (3.2).

   (c) for t in 1 to $s-1$:

      i. update $V_{e_{t+1},f_s}^T V_{e_{t+1},f_s}$, $V_{e_{t+1},f_s}^T r$, and $V_{e_{t+1},f_s}^T B$ from step 4(a) and according to the corresponding formulae.

      ii. compute and store $Y^T H_{e_{t+1},f_s} Y - Y^T H Y$ according to equation (3.2).

This search procedure returns the threshold estimate $(\hat{e}, \hat{f})$ that maximizes $Y^T H_{e,f} Y - Y^T H Y$. Then the corresponding slope estimates can be computed by plugging $(\hat{e}, \hat{f})$ in the design matrix $X_{e,f}$.

Chapter 4
SIMULATION STUDIES

In this chapter we conduct Monte Carlo (MC) experiments to evaluate the computational and statistical performance of the proposed fast grid search algorithm.

4.1 Assessment of computational performance

To study the computational performance of our fast grid search for fitting three-phase linear regression models, we compare it against a brute force grid search method at different sample sizes. In the brute force procedure, each point on the criterion function $Y^T H_{e,f} Y$ is independently computed, rather than updated sequentially as the fast grid search does. We simulate data from the three-phase model defined by equation (3.1), with true parameters $\beta_e = 5$, $\beta_f = 2$, $\gamma = 1$, $\alpha_1 = 0$, $\alpha_2 = 1$, $e = 3$, and $f = 7$. The covariate vector z is set to be one-dimensional and is uniformly distributed over $[-4, 4]$; x is taken to be uniformly distributed over $[-1, 10]$. We investigate sample sizes of 50, 100, 250, 500, and 1000. Table 4.1 summarizes the average run time of the fast grid search versus brute force grid search for three-phase model from 20 Monte Carlo replicates. For each Monte Carlo dataset, 10^3 bootstrap replicates are performed to make inference. Note that when the sample size reaches 500 or more, we failed to get results for brute force grid search as it takes too long.

Table 4.1: Average run time (second) for three-phase model estimation. Standard errors estimated from 20 Monte Carlo replicates are shown in parentheses. 10^3 bootstrap replicates are performed for each Monte Carlo dataset.

n	Grid search	Fast grid search	Ratio
50	1536.13 (407.03)	2.07 (0.10)	742
100	6379.55 (1661.94)	8.22 (0.13)	776
250	43031.93 (5620.24)	51.21 (0.90)	840
500		129.22 (30.80)	
1000		258.90 (167.43)	

The results in Table 4.1 show that the fast grid search algorithm achieves 10^2 folds improvement in computation time over the brute force grid search at sample sizes 50, 100, and 250. Specifically, the fast grid search algorithm is able to fit datasets of size 100 or less and make inference reliably within 10 seconds. Even when the sample sizes are relatively larger, e.g. 250 to 1000, the estimation and inference procedures can be completed within minutes. The computational efficiency of the proposed algorithm makes it feasible for researchers to use three-phase linear models on a routine basis.

4.2 Assessment of statistical performance

We present two series of experiments to investigate the performance of parameter estimates for three-phase models under correct model specification. In particular, we are interested in exploring the finite sample behaviors of the parameter estimates when the distance between the two thresholds goes to zero. In the first series, the data are generated from 4 different three-phase models whose limit is a two-phase segmented model. In the second series, the data are generated from 4 different three-

phase models whose limit is a stegmented model (i.e. the fusion of a step model and a segmented model).

4.2.1 Scenario I

In the first series of Monte Carlo experiments, we simulate data from four three-phase models $Y = \eta + \epsilon$, where the error term $\epsilon \sim N(0, \sigma = 0.3)$ and the mean functions η are given by:

$$\eta = z + (-6)(x-3)_- + 5(x-9)_- + 2x \tag{1a}$$

$$\eta = -10 + z + (-6)(x-3)_- + 5(x-7)_- + 2x \tag{1b}$$

$$\eta = -20 + z + (-6)(x-3)_- + 5(x-5)_- + 2x \tag{1c}$$

$$\eta = -27.5 + z + (-6)(x-3)_- + 5(x-3.5)_- + 2x \tag{1d}$$

The covariate z is uniformly distributed over [-4,4], and x is uniformly distributed over [-1,15]. The shapes of η as functions of variable x are displayed in Figure 4.1. For each model, four sample sizes are considered: 50, 200, 800, and 1600. All simulations are carried out using 10^3 Monte Carlo replicates.

Numerical summaries of the simulation results when the true models are (1a)-(1d) are collected in Table 4.2. Specifically, we include MC mean, bias, and standard deviation of the threshold estimates and slope estimates for each three-phase model. The bias estimate is defined as the MC mean minus the true value of the parameter. In addition, we obtain the rate of convergence estimated from simulations for studying convergence behaviors. To compute the rate of convergence of the parameter estimates, we use the following relationship:

$$\sigma_n = c \times n^r.$$

Here σ_n is the MC standard deviation, n is the sample size, and r is the rate of

convergence. If we do a log transformation of both sides of this equation, we get:

$$log(\sigma_n) = log(c) + rlog(n).$$

Thus, if we have multiple data points (n_i, σ_{ni}) for the parameter of a three-phase model, we can fit a linear regression model using log-transformed values of these data points and retrieve the regression coefficient as the estimated rate of convergence.

Under our simulation settings, for each parameter of interest we have four sample sizes, 50, 200, 800 and, 1600. We choose to use the two largest sample sizes to estimate the rate of convergence because there appears to be a nonlinear trend over the full range of sample sizes (Figure 4.2). The estimated rates of convergence for all parameters in model (1a)-(1d) are collected in Table 4.3.

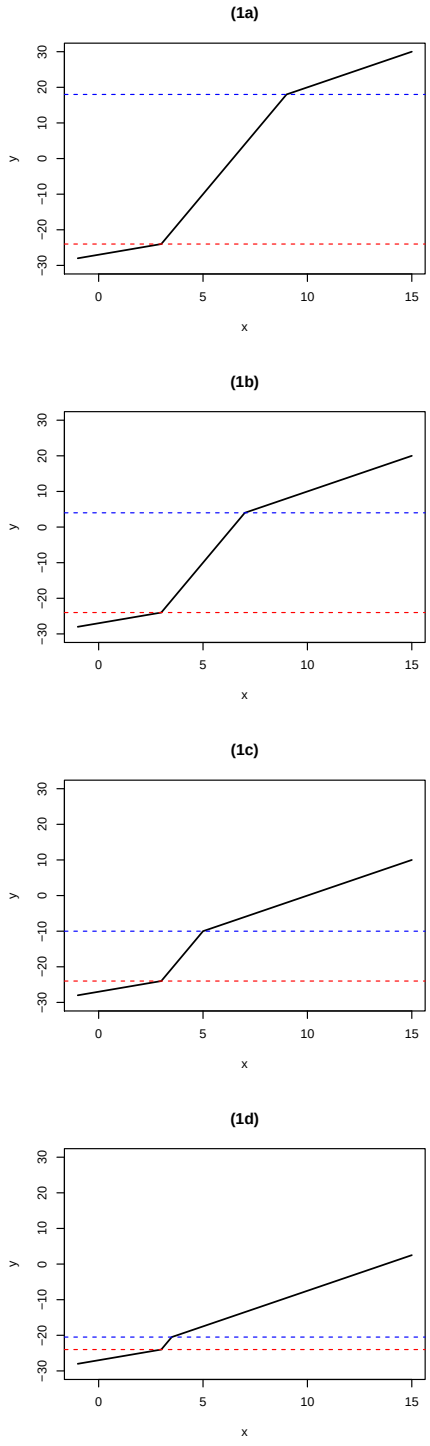

Figure 4.1: Model (1a)-(1d), a series of three-phase models whose limit is a two-phase segmented model.

Table 4.2: Summary of simulation results from 1000 Monte Carlo runs when the data is generated from model (1a)-(1d). Mean estimate, bias, and Monte Carlo standard deviation are shown. Model (1a): $\alpha_2 = 1$, $\gamma = 2$, $\beta_e = -6$, $\beta_f = 5$, $e = 3$ and $f = 9$; model (1b): $\alpha_2 = 1$, $\gamma = 2$, $\beta_e = -6$, $\beta_f = 5$, $e = 3$ and $f = 7$; model (1c): $\alpha_2 = 1$, $\gamma = 2$, $\beta_e = -6$, $\beta_f = 5$, $e = 3$ and $f = 5$; model (1d): $\alpha_2 = 1$, $\gamma = 2$, $\beta_e = -6$, $\beta_f = 5$, $e = 3$ and $f = 3.5$

	n	**Model (1a)**		**Model (1b)**		**Model (1c)**		**Model (1d)**	
		Est(bias)	MC SD	Est(bias)	MC SD	Est(bias)	MC SD	Est(bias)	MC SD
α_2									
	50	1.00 (0.00)	0.029	1.00 (0.00)	0.030	1.00 (0.00)	0.036	1.00 (0.00)	0.022
	200	1.00 (0.00)	0.010	1.00 (0.00)	0.010	1.00 (0.00)	0.010	1.00 (0.00)	0.009
	800	1.00 (0.00)	0.005	1.00 (0.00)	0.005	1.00 (0.00)	0.005	1.00 (0.00)	0.005
	1600	1.00 (0.00)	0.003	1.00 (0.00)	0.003	1.00 (0.00)	0.003	1.00 (0.00)	0.003
γ									
	50	2.02 (0.02)	0.166	1.99 (-0.01)	0.093	1.97 (-0.03)	0.068	1.99 (-0.01)	0.034
	200	2.00 (0.00)	0.042	2.00 (0.00)	0.022	2.00 (0.00)	0.012	2.00 (0.00)	0.008
	800	2.00 (0.00)	0.013	2.00 (0.00)	0.008	2.00 (0.00)	0.005	2.00 (0.00)	0.004
	1600	2.00 (0.00)	0.008	2.00 (0.00)	0.005	2.00 (0.00)	0.003	2.00 (0.00)	0.003
β_e									
	50	-5.89 (0.11)	0.300	-5.76 (0.24)	0.427	-4.98 (1.02)	1.071	-3.56 (2.44)	2.701
	200	-5.99 (0.01)	0.075	-5.99 (0.01)	0.092	-5.98 (0.02)	0.255	-5.84 (0.16)	1.669
	800	-6.00 (0.00)	0.025	-6.00 (0.00)	0.031	-6.00 (0.00)	0.079	-6.03 (-0.03)	0.563
	1600	-6.00 (0.00)	0.017	-6.00 (0.00)	0.022	-6.00 (0.00)	0.049	-6.02 (-0.02)	0.354
β_f									
	50	4.93 (-0.07)	0.167	4.84 (-0.16)	0.373	4.06 (-0.94)	1.071	2.59 (-2.41)	2.701
	200	5.00 (0.00)	0.046	4.99 (-0.01)	0.094	4.98 (-0.02)	0.269	4.84 (-0.16)	1.675
	800	5.00 (0.00)	0.017	5.00 (0.00)	0.029	5.00 (0.00)	0.082	5.03 (0.03)	0.564
	1600	5.00 (0.00)	0.011	5.00 (0.00)	0.017	5.00 (0.00)	0.048	5.02 (0.02)	0.354
e									
	50	3.00 (0.00)	0.240	2.97 (-0.03)	0.259	2.81 (-0.19)	0.335	2.71 (-0.29)	0.459
	200	3.00 (0.00)	0.062	3.00 (0.00)	0.064	3.00 (0.00)	0.068	2.97 (-0.03)	0.088
	800	3.00 (0.00)	0.018	3.00 (0.00)	0.018	3.00 (0.00)	0.020	3.00 (0.00)	0.028
	1600	3.00 (0.00)	0.010	3.00 (0.00)	0.010	3.00 (0.00)	0.012	3.00 (0.00)	0.018
f									
	50	9.02 (0.02)	0.257	7.12 (0.12)	0.336	5.48 (0.48)	0.665	4.19 (0.69)	0.935
	200	9.00 (0.00)	0.061	7.00 (0.00)	0.062	5.00 (0.00)	0.070	3.53 (0.03)	0.098
	800	9.00 (0.00)	0.017	7.00 (0.00)	0.018	5.00 (0.00)	0.021	3.50 (0.00)	0.032
	1600	9.00 (0.00)	0.010	7.00 (0.00)	0.011	5.00 (0.00)	0.012	3.50 (0.00)	0.020

In Figure 4.1, as values of e and f get closer from models (1a) to (1d), the y values at these two thresholds also approach each other such that the shape of the true model gradually approximates that of a two-phase segmented model.

Based on the results in Table 4.2, the bias decreases as the sample size increases in all four models examined under this scenario. In particular, the bias becomes minimal when the sample size is 800 or above for most parameter estimates, with the exception of $\hat{\beta}_e$ and $\hat{\beta}_f$ in model (1d). The finite sample bias of the slopes associated with the thresholds in model (1d) seems to be relatively substantial, and the bias persists even when the sample size reaches 1600. Intuitively, this may be attributed to the fact that the true values of the thresholds in model (1d) are much closer to each other compared to the other 3 models, and thus there are less samples between the two thresholds to estimate the slope in-between well. There is also a decreasing trend in the precision of $\hat{\beta}_e$, $\hat{\beta}_f$, $\hat{e}$, and $\hat{f}$ as the distance between the two thresholds gradually diminishes.

Furthermore, by comparing the magnitude of biases across different parameters in the same model, we observe that the estimate of α_2, i.e. regression coefficient of the covariate not subject to threshold effect, is least biased among the six parameters. For example, under model (1c), the bias of $\hat{\beta}_e$, $\hat{\beta}_f$, $\hat{f}$, $\hat{e}$, $\hat{\gamma}$, and $\hat{\alpha}_2$ at $n = 50$ is 1.02, -0.94, 0.48, -0.19, -0.03, and 0.00, respectively. This matches our expectations as the estimation of parameters associated with the thresholds is generally more complicated due to the non-smoothness of the likelihood function with respect to the threshold parameters. On the other hand, the finite sample bias of $\hat{\gamma}$ is also relatively small when compared with $\hat{\beta}_e$ and $\hat{\beta}_f$. This contrast is likely related to the fact that β's are the differences in slopes across three phases, while γ is the slope that does not change across thresholds. As a result, any changes in the data points around the thresholds can exert a more substantial effect on the estimation of β_e and β_f over that of γ.

Lastly, we look at the convergence behaviors of the parameter estimates. Conceptually, we would expect the least squares estimators of the parameters to converge at a standard rate of $n^{-1/2}$. However, the presence of multiple thresholds could have an impact on the convergence behaviors of the estimators. Based on the results in Table 4.3, in model (1a)-(1d) the regression coefficient estimates have a convergence rate approximately between -1/2 and -2/3, whereas the threshold estimates tend to converge at a faster rate, ranging from -0.664 to -0.834. Moreover, in model (1d) the observed convergence rate is close to 1/2 for α_2 but not for the other parameters, which is arguably due to sample sizes not large enough. Overall, the convergence rates of the parameter estimates in all four models appear to be unstable at small to moderate sample sizes, and we would expect more stable convergence behaviors when using larger data sets to conduct the experiments.

Table 4.3: Estimated rates of convergence for data generating models in the first series of simulation experiments.

	Model (1a)	**Model (1b)**	**Model (1c)**	**Model (1d)**
α_2	-0.520	-0.522	-0.525	-0.528
γ	-0.693	-0.644	-0.578	-0.472
β_e	-0.608	-0.506	-0.690	-0.671
β_f	-0.680	-0.754	-0.758	-0.673
e	-0.821	-0.834	-0.778	-0.664
f	-0.793	-0.736	-0.772	-0.693

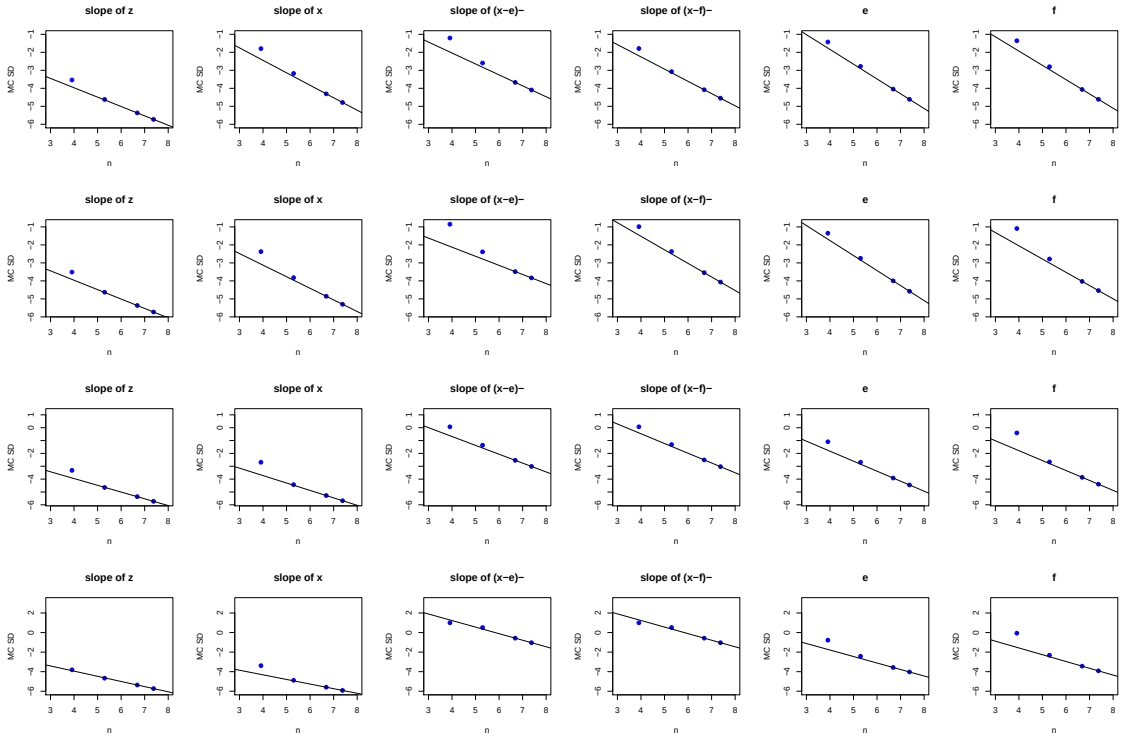

Figure 4.2: Fitted lines of log transformed Monte Carlo standard deviation of the parameter estimate against sample size n. The top to bottom rows correspond to models (1a), (1b), (1c), and (1d), respectively.

4.2.2 Remarks

Table 4.4 compares the point estimates for model (1d) when levels of the noise standard deviations range from 0.3 to 0.9. We observe that the finite sample biases of $\hat{\beta}_e$ and $\hat{\beta}_f$ are abnormally large when the noise standard deviation = 0.9. The results also suggest that as the standard deviation of noise decreases, the estimate of β_e and β_f in model (1d) becomes closer to the true values. The underlying cause of this phenomenon seems unclear to us and may warrant further investigation.

	noise SD = 0.9	noise SD = 0.7	noise SD = 0.5	noise SD = 0.3
α_2	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
γ	2.00 (0.00)	2.00 (0.00)	2.00 (0.00)	2.00 (0.00)
β_e	**-27.46 (-21.46)**	-6.80 (-0.80)	-6.25 (-0.25)	-6.03 (-0.03)
β_f	**26.46 (21.46)**	5.80 (0.80)	5.25 (0.25)	5.03 (0.03)
e	3.01 (0.01)	3.00 (0.00)	3.00 (0.00)	3.00 (0.00)
f	3.49 (-0.01)	3.50 (0.00)	3.50 (0.00)	3.50 (0.00)

Table 4.4: Monte Carlo mean and bias (in parentheses) of the parameter estimates for model (1d) when $n = 800$, with different levels of noise standard deviations

4.2.3 Scenario II

In the following series of experiments, we look at the scenario when data are simulated from 4 three-phase models $Y = \eta + \epsilon$, with the error structure $\epsilon \sim N(0, \sigma = 0.3)$ and the mean functions η given by:

$$\eta = z + (-4)(x-3)_- + 3(x-9)_- + 2x \tag{2a}$$

$$\eta = 4 + z + (-6.5)(x-3)_- + 5.5(x-7)_- + 2x \tag{2b}$$

$$\eta = 8 + z + (-14)(x-3)_- + 13(x-5)_- + 2x \tag{2c}$$

$$\eta = 11 + z + (-59)(x-3)_- + 58(x-3.5)_- + 2x. \tag{2d}$$

Again, the covariate z is uniformly distributed over [-4,4], and x is uniformly distributed over [-1,15]. The shapes of η as functions of variable x are given in Figure 4.3. The slopes in the first and third phases are set to be constant across four models, whereas the slope in the middle gradually goes to infinity from models (2a) to (2d). All other simulation settings, including sample sizes to explore and the number of Monte Carlo replicates, are the same as those defined in the first series of experiments.

The simulation results when the true models are (2a)-(2d) are summarized in Table 4.5, where we include MC mean, bias and standard deviation of the threshold estimates and slope estimates for each three-phase model. To choose the proper number of data points for estimating the rate of convergence, we again look at the linear regression lines of log-transformed MC standard deviations on the four sample sizes available (Figure 4.4) and examine how well the line fits the data points for each parameter in model (2a)-(2d). Since nonlinear trend over the full range of sample sizes is detected as before, we use the data points from two largest sample sizes to obtain the estimated rate of convergence and collect the results for model (2a)-(2d) in Table 4.6.

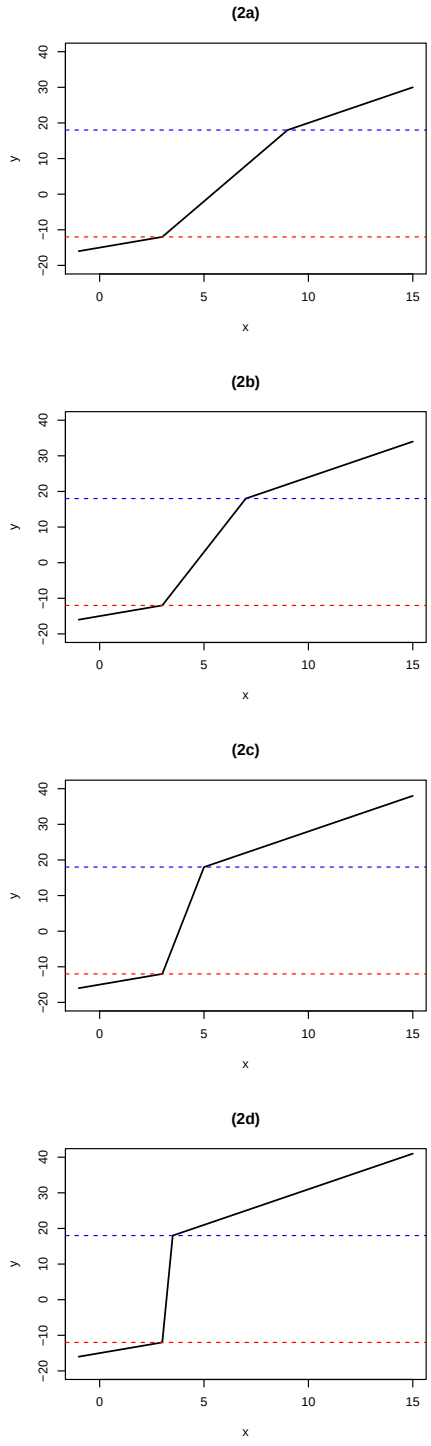

Figure 4.3: Model (2a)-(2d), a series of three-phase models whose limit is a stegmented model.

Table 4.5: Summary of simulation results from 1000 Monte Carlo runs when the data is generated from model (2a)-(2d). Mean estimate, bias, and Monte Carlo standard deviation are shown. Model (2a): $\alpha_2 = 1$, $\gamma = 2$, $\beta_e = -4$, $\beta_f = 3$, $e = 3$ and $f = 9$; model (2b): $\alpha_2 = 1$, $\gamma = 2$, $\beta_e = -6.5$, $\beta_f = 5.5$, $e = 3$ and $f = 7$; model (2c): $\alpha_2 = 1$, $\gamma = 2$, $\beta_e = -14$, $\beta_f = 13$, $e = 3$ and $f = 5$; model (2d): $\alpha_2 = 1$, $\gamma = 2$, $\beta_e = -59$, $\beta_f = 58$, $e = 3$ and $f = 3.5$

	n	**Model (2a)**		**Model (2b)**		**Model (2c)**		**Model (2d)**	
		Est(bias)	MC SD	Est(bias)	MC SD	Est(bias)	MC SD	Est(bias)	MC SD
α_2									
	50	1.00 (0.00)	0.024	1.00 (0.00)	0.032	1.00 (0.00)	0.053	1.00 (0.00)	0.090
	200	1.00 (0.00)	0.010	1.00 (0.00)	0.010	1.00 (0.00)	0.011	1.00 (0.00)	0.013
	800	1.00 (0.00)	0.005	1.00 (0.00)	0.005	1.00 (0.00)	0.005	1.00 (0.00)	0.005
	1600	1.00 (0.00)	0.003	1.00 (0.00)	0.003	1.00 (0.00)	0.003	1.00 (0.00)	0.003
γ									
	50	2.01 (0.01)	0.108	1.99 (-0.01)	0.110	1.98 (-0.02)	0.100	1.96 (-0.04)	0.098
	200	2.00 (0.00)	0.031	2.00 (0.00)	0.024	2.00 (0.00)	0.021	2.00 (0.00)	0.017
	800	2.00 (0.00)	0.012	2.00 (0.00)	0.008	2.00 (0.00)	0.007	2.00 (0.00)	0.006
	1600	2.00 (0.00)	0.007	2.00 (0.00)	0.005	2.00 (0.00)	0.004	2.00 (0.00)	0.004
β_e									
	50	-3.93 (0.07)	0.204	-6.24 (0.26)	0.467	-12.42 (1.58)	2.265	-32.30 (26.70)	22.295
	200	-3.99 (0.01)	0.058	-6.48 (0.02)	0.099	-13.97 (0.03)	0.572	-54.95 (4.05)	10.346
	800	-4.00 (0.00)	0.023	-6.50 (0.00)	0.032	-13.99 (0.01)	0.147	-58.60 (0.40)	3.195
	1600	-4.00 (0.00)	0.016	-6.50 (0.00)	0.022	-14.00 (0.00)	0.076	-58.98 (0.02)	1.610
β_f									
	50	2.96 (-0.04)	0.118	5.32 (-0.18)	0.407	11.55 (-1.45)	2.348	31.55 (-26.45)	22.623
	200	3.00 (0.00)	0.038	5.49 (-0.01)	0.100	12.97 (-0.03)	0.609	53.97 (-4.03)	10.399
	800	3.00 (0.00)	0.015	5.50 (0.00)	0.030	12.99 (-0.01)	0.155	57.61 (-0.39)	3.214
	1600	3.00 (0.00)	0.010	5.50 (0.00)	0.018	13.00 (0.00)	0.080	57.98 (-0.02)	1.619
e									
	50	3.00 (0.00)	0.241	2.97 (-0.03)	0.262	2.88 (-0.12)	0.279	2.72 (-0.28)	0.320
	200	3.00 (0.00)	0.066	3.00 (0.00)	0.063	3.00 (0.00)	0.063	2.97 (-0.03)	0.072
	800	3.00 (0.00)	0.020	3.00 (0.00)	0.018	3.00 (0.00)	0.016	3.00 (0.00)	0.016
	1600	3.00 (0.00)	0.012	3.00 (0.00)	0.010	3.00 (0.00)	0.009	3.00 (0.00)	0.008
f									
	50	9.02 (0.02)	0.265	7.12 (0.12)	0.356	5.24 (0.24)	0.407	3.95 (0.45)	0.473
	200	9.00 (0.00)	0.068	7.00 (0.00)	0.061	5.00 (0.00)	0.063	3.53 (0.03)	0.070
	800	9.00 (0.00)	0.022	7.00 (0.00)	0.017	5.00 (0.00)	0.016	3.50 (0.00)	0.016
	1600	9.00 (0.00)	0.014	7.00 (0.00)	0.010	5.00 (0.00)	0.008	3.50 (0.00)	0.008

As displayed in Figure 4.3, the values of e and f get closer from models (2a) to (2d) as before, but instead of letting the y values also approach each other, it is assumed that the y values at these two thresholds stay apart from each other so the model gradually shifts from a typical three-phase model to a stegmented model.

Similar to Scenario I, the accuracy of the parameter estimates increases with sample size in all four models considered. From Table 4.5 we observe that at $n = 800$, $\hat{\beta}_e$ and $\hat{\beta}_f$ in models (2c) and (2d) tend to have relatively considerable finite sample bias. This situation is improved when the sample size reaches 1600, while there is still noticeable bias for $\hat{\beta}_e$ and $\hat{\beta}_f$ in model (2d), which indicates that a greater sample size would be needed if we want to have more accurate estimates in model (2d) to reach a similar level as those in other models. Further, looking at the same sample size, the MC standard deviations of $\hat{\beta}_e$ and $\hat{\beta}_f$ increase drastically from models (2a) to (2d). For example, when $n = 200$, the MC standard deviations of $\hat{\beta}_f$ are 0.038, 0.100, 0.609, 10.399 in models (2a), (2b), (2c), and (2d), respectively. These patterns again can be explained by the much smaller distance between the two thresholds in model (2d). The fact that the true values of the thresholds are very close makes it more difficult to estimate the slope in-between as a result of less data points available between the two thresholds.

We then investigate the convergence behaviors of model (2a)-(2d) based on the results in Table 4.6. Under this setting, we see similar results with some noteworthy differences. First, the thresholds e and f tend to converge at a faster rate of roughly between $n^{-0.719}$ and $n^{-1.011}$. There is also an increasing trend observed for the convergence rate of these two thresholds when the shape of the model gradually approaches that of a stegmented model from (2a) to (2d). For example, the estimated convergence rates of threshold e are -0.752, -0.833, -0.930, and -1.011 for model (2a), (2b), (2c), and (2d), respectively. According to the theories established in Hansen (2000),

under correct model specification, the threshold estimate of a discontinuous (i.e. step or stegmented) threshold regression model converges at a faster rate of n^{-1}, while the regression coefficient estimates converge at the regular $n^{-1/2}$ rate. Our results can thus serve as an empirical evidence to support Hansen's theoretical insights regarding the convergence behaviors of the thresholds.

On the other hand, while the observed convergence rate of the regression coefficient estimates is slower in general compared to that of the thresholds, it is still faster than the usual rate of $n^{-1/2}$. Again, this is likely due to insufficient sample sizes and we would expect their convergence behaviors to further stabilize as the sample size increases.

Table 4.6: Estimated rates of convergence for data generating models in the second series of simulation experiments.

	Model (2a)	**Model (2b)**	**Model (2c)**	**Model (2d)**
α_2	-0.520	-0.525	-0.538	-0.610
γ	-0.637	-0.684	-0.791	-0.679
β_e	-0.549	-0.538	-0.948	-0.989
β_f	-0.629	-0.769	-0.960	-0.989
e	-0.752	-0.833	-0.930	-1.011
f	-0.719	-0.783	-1.010	-0.995

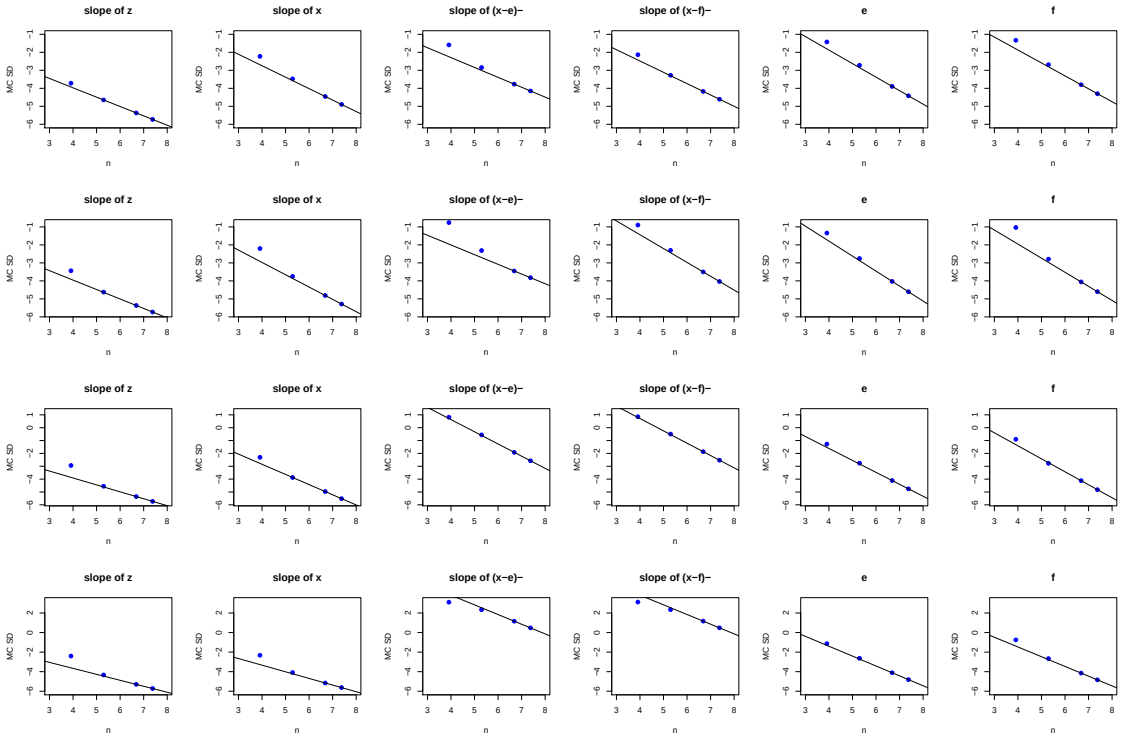

Figure 4.4: Fitted lines of log transformed Monte Carlo standard deviation of the parameter estimate against sample size n. The top to bottom rows correspond to models (2a), (2b), (2c), and (2d), respectively.

Chapter 5
DISCUSSION

In this thesis we presented a fast grid search algorithm for multi-phase regression models, with a primary focus on three-phase linear models, and demonstrated its computational and statistical performance through simulation studies. Compared to a two-dimensional brute force grid search procedure for three-phase model estimation, our algorithm does the job with greatly improved computational efficiency. Additionally, we studied the convergence behaviors of the parameter estimates when the distance between the two thresholds goes to zero. It would be an interesting extension to investigate how the results may differ if the the sample size and the distance between thresholds are changed concurrently. The proposed algorithm has been implemented in the R package *chngpt* (Fong et al., 2017a), readily available for practical use. We hope that this thesis could contribute to the existing body of literature on threshold regression models, and make it a more efficient tool for modeling nonlinear relationships.

While we have illustrated the algorithm in the context of models with two thresholds in a single covariate, the same idea can also be applied to the estimation of models with more than two thresholds, and/or across multiple covariates. However, the computational costs associated with using a grid search procedure to fit higher-dimensional models grow rapidly. To be more specific, letting K denote the number of threshold parameters, then the size of the set of candidate threshold values grows at the rate of n^K. For situations where more than two thresholds exist, it may be more feasible to pursue grid search methods that seek to sequentially determine multiple

thresholds, rather than searching over a very high-dimensional grid.

Furthermore, the proposed algorithm assumes known number of threshold parameters in the model, but it is likely that the true number of thresholds is unknown and needs to be estimated in practice. This involves additional investigation and is outside the scope of this thesis. A possible direction for future research is to incorporate threshold selection procedures into the current methodology to identify the appropriate number of thresholds and the optimal model.

BIBLIOGRAPHY

Bai, J. (1998), "Estimation of multiple-regime regressions with least absolutes deviation," *Journal of Statistical Planning and Inference*, 74, 103–134.

Bai, J. and Perron, P. (2003), "Computation and analysis of multiple structural change models," *Journal of Applied Econometrics*, 18, 1–22.

Barrodale, I. and Roberts, F.D.K. (1974), "Solution of an overdetermined system of equations in the l1 norm [F4]," *Commun. ACM*, 17, 319–320.

Bellman, R. and Roth, R. (1969), "Curve fitting by segmented straight lines," *Journal of the American Statistical Association*, 64, 1079–1084.

Bellman, R.E. and Dreyfus, S.E. (1962), *Applied Dynamic Programming*, Princeton University Press.

Breiman, L., Friedman, J., Stone, C. and Olshen, R. (1984), *Classification and Regression Trees*, The Wadsworth and Brooks-Cole statistics-probability series, Taylor & Francis.

Elder, A. and Fong, Y. (2019), "Estimation and inference for upper hinge regression models," *Environmental and Ecological Statistics*, 26, 287–302.

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.

Friedman, J.H. and Silverman, B.W. (1989), "Flexible parsimonious smoothing and additive modeling," *Technometrics*, 31, 3–21.

Hansen, B.E. (2000), "Sample splitting and threshold estimation," *Econometrica*, 68, 575–603.

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

Hawkins, D.M. (2001), "Fitting multiple change-point models to data," *Computational Statistics & Data Analysis*, 37, 323 – 341.

Khan, M.E. (2008), "Updating inverse of a matrix when a column is added/removed," Tech. rep., University of British Columbia, Vancouver, BC.

Muggeo, V.M.R. (2003), "Estimating regression models with unknown break-points," *Statistics in Medicine*, 22, 3055–3071.

Muggeo, V.M.R. (2008), "segmented: an R package to fit regression models with broken-line relationships," *R NEWS*, 8/1, 20–25.

Muggeo, V.M.R. and Adelfio, G. (2010), "Efficient change point detection for genomic sequences of continuous measurements," *Bioinformatics*, 27, 161–166.

Permar, S.R., Fong, Y., Vandergrift, N., Fouda, G.G., Gilbert, P., Parks, R. et al (2015), "Maternal HIV-1 envelope–specific antibody responses and reduced risk of perinatal transmission," *Journal of Clinical Investigation*, 125, 2702–2706.

Ulm, K. (1991), "A statistical method for assessing a threshold in epidemiological studies," *Statistics in Medicine*, 10, 341–349.

Zeileis, A., Kleiber, C., Krämer, W. and Hornik, K. (2003), "Testing and dating of structural changes in practice," *Computational Statistics & Data Analysis*, 44, 109–123.