2019

Knot Optimization For Univariate Cubic L^1 Spline Fits with Application in Change Point Detection

Manfei Xie
z1834367@students.niu.edu

ABSTRACT

KNOT OPTIMIZATION FOR UNIVARIATE CUBIC L^1 SPLINE FITS WITH APPLICATION IN CHANGE POINT DETECTION

Manfei Xie, MS
Department of Industrial and Systems Engineering
Northern Illinois University, 2019
Dr. Ziteng Wang, Co-Director
Dr. Purushothaman Damodaran, Co-Director

Cubic L^1 spline fits, as a type of L^1 approximating splines, have shown superior performance in shape preservation of geometric data with great changes. To better construct a cubic L^1 spline fit, the number and the location of the spline knots should be optimized rather than predetermined. This research investigates knot optimization methods for univariate cubic L^1 spline fits. When the number of knots is given, we design an optimization-based method to determine the best location of the knots. When the number and the location are unknown, we propose a heuristic method to find proper knot number and location. Numerical experiments show that cubic L^1 spline fits with optimized knots can better approximate data and preserve shapes. The cubic L^1 spline fits with optimized knots show good potential when applied in change point detection.

NORTHERN ILLINOIS UNIVERSITY
DE KALB, ILLINOIS

MAY 2019

KNOT OPTIMIZATION FOR UNIVARIATE CUBIC L^1 SPLINE FITS WITH APPLICATION IN CHANGE POINT DETECTION

BY

MANFEI XIE

A THESIS SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE

MASTER OF SCIENCE

DEPARTMENT OF INDUSTRIAL AND SYSTEMS ENGINEERING

Thesis Co-Director:
Dr. Ziteng Wang
Thesis Co-Director:
Dr. Purushothaman Damodaran

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor Dr. Ziteng Wang for his guidance, great ideas, encouragement, and support. I feel fortunate to work as a research assistant with Dr. Wang. In weekly meetings with Dr. Wang over the past two years, we discussed every aspect of conducting this research and presenting our work. Dr. Wang demonstrates himself a responsible and rigorous researcher that has influenced me a lot. Also, Dr. Wang provided recommendation letters for my application of a PhD program.

It is also my great honor to have Dr. Purushothaman Damodaran, Dr. Christine Nguyen and Dr. Reinaldo Moraga in my Thesis Committee. Their constructive criticism and comments inspired me to improve a great deal of my research and thesis. I would also appreciate Dr. Ehsan Asoudegi, Dr. Damodaran, and Dr. Nguyen for their thoughtful and invaluable suggestions on my decision of pursuing a higher degree.

A big thank you must be sent to Northern Illinois University for the graduate assistantship that supported my research and education. My special thanks go to the wonderful staff of the Department of Industrial and Systems Engineering, Ms. Tracy Mereness and Ms. Christel Ackland for the kind help they gave me during my master study.

I am grateful for my parents and parents-in-law. My parents allowed me lots of independence in creating my ideas and making my own decisions since I was young. The confidence I had to tackle various challenges on the MSc journey originated from the trust they put in me. My parents-in-law supported and understood me as the most they could, which enabled me to concentrate on my research while fulfilling my family obligations.

I am so very blessed with my husband and my baby boy accompanying me throughout the MSc experience. I couldn't ask for a better husband and a better son. My husband,

lover, and soulmate, Long, loves and supports me wholeheartedly, which is the power drives me to go beyond myself and pursue my dreams. I am so touched by and proud of every incredible growth my son, Eiffel, made since he was born, especially, I could not play with him as much as I should have for the most of the time. Every moment with Eiffel is full of giggling and brings the sweetest memories in my life.

DEDICATION

To my beloved Long and Eiffel.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

Originally, splines were used by engineers to draw curves for building boats. Over centuries of development, there have been a vast variety of different types of splines. Generally speaking, a spline is a piecewise polynomial function, of which the pieces are connected at a certain number of knots (or referred as "nodes"). Splines are primarily used to approximate and interpolate data. Figure 1.1 shows an example of splines that approximates the given data. In the figure, the given data points are plotted as "+". The solid curves are the spline. The spline consists of five pieces that are connected by the red knots. There are totally six knots in the spline.

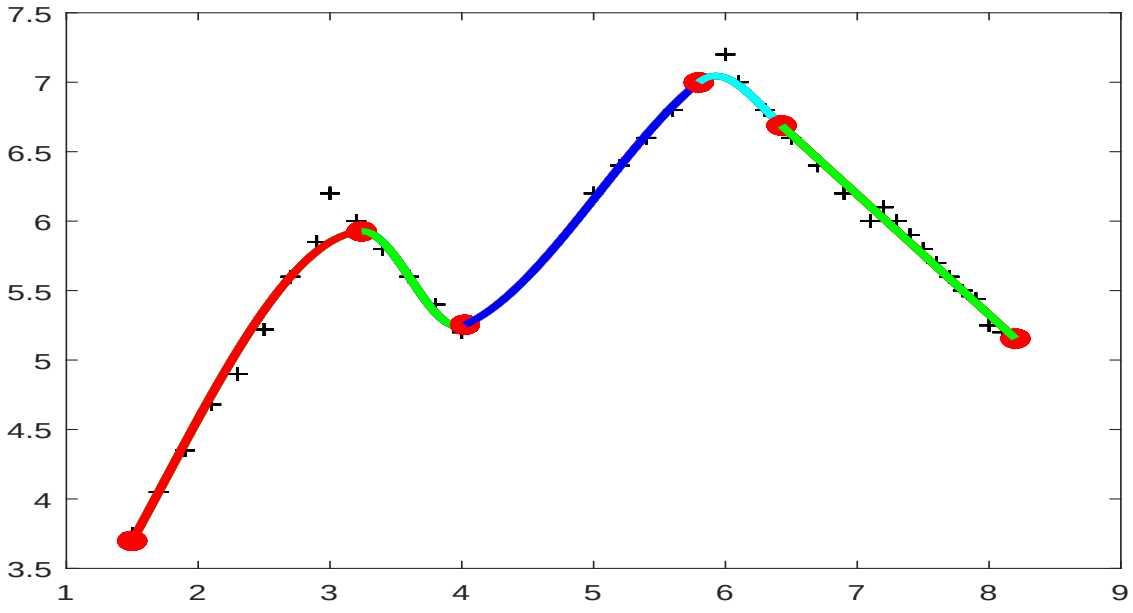

Figure 1.1: A cubic L^1 spline fit.

Nowadays, splines are broadly applied in science and engineering applications including robot robust control [12], gene expression [24], medical image segmentation [1], structure analysis of finance data [6], geometric designs of aircraft [10], geopotential reconstruction in Geodesy [17], and outline of terrain description in military [31].

In the past two decades, since univariate cubic L^1 splines were introduced by Lavery [18], there have been continuous development of L^1 splines, including development of univariate [19] and bivariate [35] cubic L^1 spline fits, analytical solution of univariate cubic L^1 spline functional in 5-point window [13, 34], global [19] and local [33] claculations for univariate cubic L^1 spline fits, and theoretical analysis on the shape-preserving capability of univariate cubic L^1 spline fits [32, 26].

Cubic L^1 splines minimize L^1 norm of spline functionals. L^1 splines were proved by Lavery [18, 19] as outperforming conventionally used L^2 splines in shape-preserving. L^1 splines can be categorized into two classes: interpolating and approximating splines. The approximating splines are more practically important than the interpolating splines [33]. As a type of approximating splines, cubic L^1 spline fits do not require the user to choose a balance parameter and are of interest of this research.

To construct a spline function, knots need to be determined. A better choice of knots could help a spline to better preserve the shapes of data by approximation or interpolation. Knot determination includes determination of number and location of knots. Existing research on L^1 splines often takes knots as given. Existing methods for knot optimization are done for other types of splines, which can not be directly applied for cubic L^1 spline fits. Therefore, this research investigates the knot optimization for univariate cubic L^1 spline fits.

To further demonstrate the use of univariate cubic L^1 spline fits and the benefits of knot optimization, we investigate the possibility of applying univariate cubic L^1 spline fits with knot optimization in change point detection (CPD). CPD is the problem to find abrupt variations that occur between different states in data [2]. CPD has been studied for a broad

range of purposes such as medical condition monitoring [25], climate change detection [9], speech recognition [27], and human activity analysis [7]. Many works have been done using non-spline-based methods either statistically or geometrically detect change points based on probability density models, autoregressive models, state-space models, averages, covariances, autocorrelation, or spectrums [15]. Only a few of them used spline-based methods. Among this type of method, a linear combination of B-spline basis functions was used for detecting change in monotonicity and concavity and convexity [21]. Nevertheless, there is no work yet to apply cubic L^1 spline fits for CPD. The optimized knot location in cubic L^1 spline fits can be good candidates of change points. For example, the different shapes of a spline fit before and after the knots may indicate change of magnitude in data. In this research, we apply univariate cubic L^1 spline fits with knot optimization in CPD.

The remaining of this thesis is organized as follows. The remaining of Chapter 1 states the research problems, the objectives, scope, assumption, benefits and deliverables of the thesis. Chapter 2 reviews existing literature which is related to the cubic L^1 spline fits, knot determination, and CPD. Chapter 3 proposes the methodology for knot location optimization, knot number and location determination, and CPD. Chapter 4 presents results of numerical experiments on both artificial and real data. Chapter 5 concludes the thesis and discusses future research.

1.1 Problem Description

In this research, a univariate cubic spline is a C^1-Smooth piecewise cubic function $z(x)$ with strictly monotonically increasing but otherwise arbitrary knots $x_i, i = 1, 2, \ldots, I$. In each interval $[x_i, x_{i+1}], i = 1, 2, \ldots, I-1$, $z(x)$ takes the following form:

$$z(x) = z_i + b_i(x - x_i) + \frac{1}{l_i}[-(2b_i + b_{i+1}) + 3\triangle z_i](x - x_i)^2 + \frac{1}{{l_i}^2}[b_i + b_{i+1} - 2\triangle z_i](x - x_i)^3, \quad (1.1)$$

where z_i, z_{i+1} are the function values of $z(x)$ at x_i and x_{i+1}, b_i, b_{i+1} are the first-order derivatives of $z(x)$ at x_i and x_{i+1}, $l_i = x_{i+1} - x_i$, $\triangle z_i = \frac{z_{i+1} - z_i}{l_i}$ [18]. Denote $\boldsymbol{x} = (x_1, x_2, \ldots, x_I)$, $\boldsymbol{z} = (z_1, z_2, \ldots, z_I)$, and $\boldsymbol{b} = (b_1, b_2, \ldots, b_I)$.

If $\boldsymbol{x}$ and $\boldsymbol{z}$ are given, a cubic L^1 (interpolating) spline is a univariate cubic spline $z(x)$ that minimizes the spline functional

$$\int_{x_1}^{x_I} |\frac{d^2 z}{dx^2}| dx, \quad (1.2)$$

by choosing $\boldsymbol{b}$.

Consider fitting a given series of data $(\hat{x}_m, \hat{z}_m)$ where $m = 1, 2, ...M$ and $\hat{x}_1 \leq \hat{x}_2 \leq \ldots \leq \hat{x}_M$. A cubic L^1 spline fit is a cubic L^1 spline that minimizes the L^1-norm of the data approximation error

$$\sum_{m=1}^{M} |\hat{z}_m - z(\hat{x}_m)|, \quad (1.3)$$

by choosing parameters including $I, \boldsymbol{x}, \boldsymbol{z}$ and $\boldsymbol{b}$. Without loss of generality, we may assume $I \geq 1, x_1 = \hat{x}_1$, and $x_I = \hat{x}_M$.

If the number of knots I and location of the knots $\boldsymbol{x}$ are predetermined, Lavery [19] and Wang et al. [33] have developed global and local calculation methods for optimizing $\boldsymbol{z}$ and $\boldsymbol{b}$, respectively. However, the predetermined knot location is not necessarily optimal for data fitting. In this research, we investigate knot optimization methods when I and $\boldsymbol{x}$ are not given and explore potential applications. Specially, this thesis is focused on the following three research problems:

(1) If the number of knots I is predetermined but their location $\boldsymbol{x}$ is free, how can we determine $\boldsymbol{x}, \boldsymbol{z}$ and $\boldsymbol{b}$ of the cubic L^1 spline fits?

(2) If both I and $\boldsymbol{x}$ are free, how can we determine $I, \boldsymbol{x}, \boldsymbol{z}$ and $\boldsymbol{b}$ of the cubic L^1 spline fits?

(3) Can the new methods developed in (1) and (2) be applied to CPD?

1.1.1 Change Point Detection

In general, CPD is the problem to find abrupt variations that occur between different states in data [2]. Such variations may include change in average, variance, monotonicity, convexity, continuity, etc. Figure 1.2 shows a sample of time series that contains several states separated by change points.

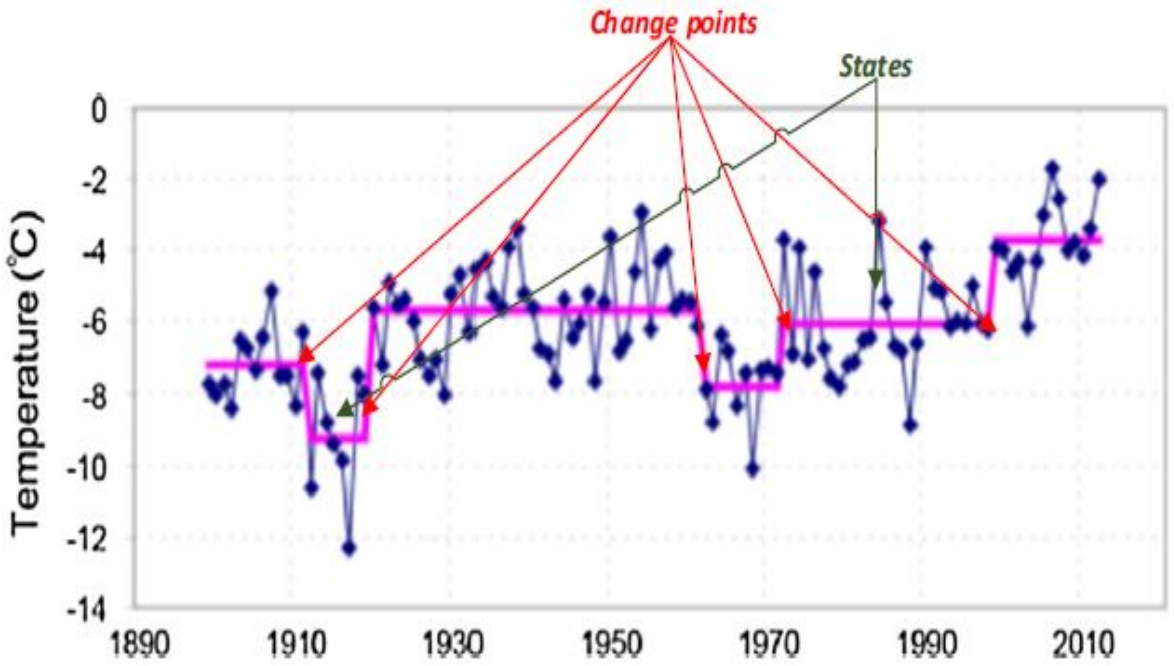

Figure 1.2: Sample time series and change points (horizontal lines indicate separate states) [2]

Given time series data $(\hat{x}_m, \hat{z}_m)$ $(m = 1, 2, ...M)$, we study the applicability of using cubic L^1 spline fits to identify change points. The optimized knot location can be good candidates of change points. For example, in two consecutive intervals divided by a knot, the different shapes of the cubic L^1 spline fits may indicate change of magnitude in data.

1.2 Thesis Objectives, Scope, and Assumption

This research aims to expand the capability of the cubic L^1 spline fits for application purpose. Specially, this thesis will achieve the following objectives:

- Develop the mathematical models and algorithms to optimize number and location of knots for cubic L^1 spline fits.
- Implement the algorithms in MATLAB and test the numerical experiments.

- Provide a proof of concept of applying knot optimization for cubic L^1 spline fits in CPD.

The scope of the thesis is limited to second-order derivative based univariate cubic L^1 spline fits. Bivariate and multivariate L^1 splines, L^2 splines, L^p splines $(0 < p < 1)$, first-order derivative based L^1 splines, function-value based L^1 splines are out of the scope.

It is assumed in this thesis that the given data $\{(\hat{x}_m, \hat{z}_m)\}$ includes no uncertain or missing values.

1.3 Thesis Benefits and Deliverables

This thesis provides the following expected benefits:

- With knots optimized, cubic L^1 splines fits can approximate data with higher accuracy.
- The knot optimization methodologies will lay the foundation for research of knot optimization for bivariate and multivariate L^1 splines.
- The cubic L^1 spline fits, with knot optimization techniques, will provide an alternative for change point detection.
- The success of using cubic L^1 spline fits applied in change point detection will inspire more applications of L^1 splines in solving real-world problems.

Deliverables of this thesis include the following:

- A thorough procedure to utilize augmented Lagrangian method to solve the knot location optimization problem.
- A heuristic method to decide the optimal number of knots.
- A method to apply cubic L^1 spline fits for change point detection.
- A MATLAB program that implements the algorithm.
- A detailed report that documents this research, its findings, and experimental results.

CHAPTER 2

LITERATURE REVIEW

In this chapter, we review literature that is most related to this research. The collected literature is organized into the following three categories:

- Cubic L^1 Spline Fits
- Knot Determination
- Change Point Detection

Table 2.1 lists the reviewed literature and indicates their relevance to the three categories.

Table 2.1: Summary of past literature reviewed

Sources	**Cubic L^1 Spline Fits**	**Knot Determination**	**Change Point Detection**
Arslan [30]		√	
Beliakov [3]		√	
De Boor [8]		√	
Han et al. [11]			√
Jin et al. [13]	√		
Lavery [19]	√		
Li et al. [20]		√	
Liao and Meyer [21]			√
Lyche and Mørken [23]		√	
Kang et al. [14]		√	
Kawahara et al. [16]			√
Kawahara and Sugiyama [15]			√
Tjahjowidodo [28]		√	
Tjahjowidodo et al. [29]		√	
Wang et al. [32]	√		
Wang et al. [33]	√		
Yu et al. [34]	√		
Zaman [35]	√		
This Thesis	√	√	√

2.1 Cubic L^1 Spline Fits

Cubic L^1 spline fits were first proposed by Lavery [19] which also showed its superior shape-preserving capability over L^2 spline fits and L^2 smoothing splines. With knot number and location predetermined, Lavery [19] formulated the calculation of the univariate cubic L^1 spline fits as a bi-level optimization problem. The approximation error function and the cubic L^1 spline functional were discretized. Consequently, the bi-level problem was approximated by a linear programming problem, and then solved by a Lagrange-multiplier-based primal affine algorithm. Wang et al. [33] developed a local calculation method to solve the univariate cubic L^1 spline fit problem by utilizing the analytic solution of the cubic L^1 spline functional in a 5-point local algorithm proposed in [13, 34]. Therefore, the bi-level problem was converted to an unconstrained nonlinear programming problem which was solved by a steepest decent algorithm. Computational experiments in [33] showed that the locally calculated L^1 spline fits can preserve shapes better than globally calculated L^1 spline fits and were more computationally efficient.

Wang et al. [32] took the first step to theoretically analyze the shape-preserving capability of univariate cubic L^1 spline fits for a a basic shape of two parallel line segments. The observed strong shape-preserving capability of spline fits was supported by the calculation results of the defined shape-preserving metric in both 3-point and 5-point windows. Given a rectangular grid generated by the tensor product of nodes, Zaman [35] formulated the bivariate cubic (bicubic) L^1 spline fits problem as a bi-level optimization problem. He calculated the bicubic interpolating splines by using the 5-point window algorithm (previously was used for the calculation of univariate interpolating splines in [34]). Then the steepest descent method was used to solve the unconstrained optimization problem converted from the bi-level optimization problem. The experiment results showed that both the construct-

ed bicubic interpolation and the approximation splines can preserve the shapes of 3D data effectively.

2.2 Knot Determination

Existing literature has investigated methods for determining knots of L^2 splines. The number and location of knots can be predetermined. For example, knots can be evenly spaced or can take the Chebyshev points [8], or can depended on the change of radius of curvature [20]. However, overshooting could be an issue when knots are evenly spaced in the case of "smooth inhomogeneous or discontinuous data" [28]. Optimization of knots is needed in this case.

When the number of knots is predetermined but knot location is free, optimization tools can be used for knot determination and are found to be superior than other knot determination methods [28]. These optimization approaches can be roughly categorized into two classes: deterministic approaches and stochastic approaches. As an example of deterministic approaches, Beliakov [3] applied a cutting angle method to obtain knot location which was then modified by local discrete gradient method to guarantee the global optimum. For the class of stochastic approaches, Ulker and Arslan [30] employed an artificial immune system to solve a discrete combinatorial optimization problem to calculate the knot location of B-splines.

When both number and location of knots are free, the knots are often determined by modifying predetermined initial knots. In the modification process, location of knots can be shifted and the number of knots can be increased or decreased. There are roughly two kinds of strategies to achieve this: knot insertion strategy and knot removal strategy. In the first strategy, a bisecting method is usually employed. New knots are added to a small number

of given knots until some specific criteria are satisfied. For instance, Tjahjowidodo et al. [29] calculated the candidate knots for a cubic spline by bisecting the second derivative of the data to obtain the best fitted linear piecewise functions. Then an optimization tool was used to modify the location of the candidate knots. In the second strategy, knots that are less essential are eliminated from the initially given knots to obtain the final knots. Lyche and Mørken [23] used all sample data points as the initial knots for B-spline. Kang et al. [14] solved a convex sparse optimization model and chose points where "a certain order derivatives of the fitting spline is discontinuous" as initial knots.

2.3 Change Point Detection

Most commonly used CPD methods are not splines-based approaches. A typical statistical approach is based on likelihood ratio methods. When probability distributions of data before and after a candidate change point are significantly different, the candidate is identified as a change point [15]. This approach monitors the logarithm of the likelihood ratio between consecutive intervals of sequence data to detect change points. Another type of approach, geometrically-based subspace model methods, detects change points by analyzing subspaces in which time series sequences are constrained. This approach is strongly connected with a system identification method, subspace identification.

The above-mentioned approaches rely on "pre-specified parametric models such as probability density models, autoregressive models, and state-space models, or some specific quantities such as averages, covariances, autocorrelation and spectrums" [15]. However, Kawahara and Sugiyama [15] estimated the density ratio directly (non-parametric) by providing an online version of the Kullback-Leibler Importance Estimation Procedure (KLIEP) algorithm for CPD. In addition, Kawahara et al. [16] proposed nonparametric algorithms for CPD

based on subspace identification. In their work, the distance evaluated as the gap between subspaces was employed as the measure of the change in time sequences.

Some spline-based methods have been used to identify change points. These methods use various types of conventional (L^2) splines. Han et al. [11] used the partial spline model for CPD. To estimate the change point, sum (in terms of a smoothing parameter) of a model selection criterion was minimized. Differently, Liao and Meyer [21] used a linear combination of B-spline basis functions with constraints to estimate a regression function. There are two types of the constraints. If the regression function is increasing-decreasing, its change point will be the mode. In this scenario, the constraints will enforce the linear combination to increase before the change point occurs and decrease after the change point. If the regression function is convex-concave, its change point will be the inflection point. In this scenario, the constraints will enforce the linear combination to be convex (concave) before (after) the change point occurs. In both cases, the change point is detected by searching through the domain of the regression function to find the point that minimizes the sum of squared residuals (fitting error of the linear combination to estimate the regression function).

CHAPTER 3

METHODOLOGY

3.1 Optimize Knot Location

Consider fitting given data $(\hat{x}_m, \hat{z}_m)$ $(m = 1, 2, ...M)$, when the number of knots I is predetermined but $\boldsymbol{x}, \boldsymbol{z}$, and $\boldsymbol{b}$ are unknown. Without loss of generality, we fix $x_1 = \hat{x}_1$ and $x_I = \hat{x}_M$. To simplify notation without causing ambiguity, in this section, we continue to use $\boldsymbol{x}$ in the optimization problems while the decision variables are $x_2, x_3, \ldots, x_{I-1}$. Denote the data approximation error by

$$f(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{b}) = \sum_{m=1}^{M} |\hat{z}_m - z(\hat{x}_m)|.$$

The cubic L^1 spline fit problem can be formulated as the following bi-level optimization problem

$$\begin{aligned} \min \quad & f(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{b}) \\ \text{s.t.,} \quad & x_{i+1} - x_i \geq \delta, \ i = 1, ..., I-1 \\ & \boldsymbol{b} \min \int_{x_1}^{x_I} |\frac{d^2 z}{dx^2}| dx, \end{aligned} \tag{3.1}$$

where the upper level minimizes the data approximation error and the lower level determines $\boldsymbol{b}$ by minimizing the L^1 spline functional. $\delta = \min\{\hat{x}_{m+1} - \hat{x}_m\}$ $(m = 1, 2, ...M-1)$. The linear constraint $x_{i+1} - x_i \geq \delta$ ensures monotonic increase of knots $x_i, i = 1, 2, \ldots, I$. The

minimal distance between two consecutive knots is greater than or equal to the minimal distance between two consecutive data points.

Jin et al. [13] and Yu et al. [34] have developed a 5-point moving window algorithm which is a local calculation method to analytically solve the lower level problem. This algorithm provides the closed-form solution $\boldsymbol{b} = \boldsymbol{b}(\boldsymbol{x}, \boldsymbol{z})$ for any given $\boldsymbol{x}$ and $\boldsymbol{z}$. Therefore, substituting $\boldsymbol{b} = \boldsymbol{b}(\boldsymbol{x}, \boldsymbol{z})$ in $f(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{b})$, we define

$$F(\boldsymbol{x}, \boldsymbol{z}) = f(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{b}(\boldsymbol{x}, \boldsymbol{z}))$$

and the cubic L^1 spline fit problem (3.1) is converted to the following linearly constrained nonlinear optimization problem

$$\begin{aligned} &\min \quad F(\boldsymbol{x}, \boldsymbol{z}) \\ &\text{s.t.,} \quad x_{i+1} - x_i \geq \delta, \ i = 1, ..., I-1. \end{aligned} \tag{3.2}$$

We use the augmented Lagrangian method to solve problem (3.2). Augmented Lagrangian method is "one of the most effective general classes of nonlinear programming methods" [22]. This method combines the penalty method and the Lagrangian duality method. Applied to problem (3.2), the augmented Lagrangian method is explained in details below.

To define the augmented Lagrangian function, we convert problem (3.2) into an equality constrained problem by adding slack variables $s_i, i = 1, 2, ..., I-1$, as follows.

$$\begin{aligned} &\min \quad F(\boldsymbol{x}, \boldsymbol{z}) \\ &\text{s.t.,} \quad x_{i+1} - x_i - \delta - s_i^2 = 0, \ i = 1, ..., I-1. \end{aligned} \tag{3.3}$$

The augmented Lagrangian function is defined as

$$\phi(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{s}; \boldsymbol{w}, \sigma) = F(\boldsymbol{x}, \boldsymbol{z}) - \sum_{i=1}^{I-1} w_i(h_i - {s_i}^2) + \frac{\sigma}{2} \sum_{i=1}^{I-1} (h_i - {s_i}^2)^2, \tag{3.4}$$

where $\boldsymbol{w} = (w_1, w_2, \ldots, w_{I-1})$ denotes multipliers, σ is a penalty factor, and $h_i = x_{i+1} - x_i - \delta$. We also denote the equality constrains $g_i = x_{i+1} - x_i - \delta - {s_i}^2$. g_i will be used later in this subsection to determine the stop of the augmented Lagrangian method algorithm and the update of σ and $\boldsymbol{w}$.

For any fixed σ , define the dual function in $\boldsymbol{w}$ as

$$\varphi(\boldsymbol{w}; \sigma) = \min_{\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{s}} \phi(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{s}; \boldsymbol{w}, \sigma). \tag{3.5}$$

Note that it is easy to find the closed-form

$$s_i^2 = \frac{1}{\sigma} \max(0, \sigma h_i - w_i) \tag{3.6}$$

that minimizes $\phi(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{s}; \boldsymbol{w}, \sigma)$ for given $\boldsymbol{x}$ and $\boldsymbol{z}$ [5]. Therefore, substituting s_i^2 in $\phi(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{s}; \boldsymbol{w}, \sigma)$, we have

$$\Phi(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{w}, \sigma) = F(\boldsymbol{x}, \boldsymbol{z}) + \frac{1}{2\sigma} \sum_{i=1}^{I-1} \{[\max(0, w_i - \sigma h_i)]^2 - (w_i)^2\} \tag{3.7}$$

and

$$\varphi(\boldsymbol{w}; \sigma) = \min_{\boldsymbol{x}, \boldsymbol{z}} \Phi(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{w}, \sigma). \tag{3.8}$$

According to Luenberger and Ye [22], when σ is sufficiently large, if $\boldsymbol{w}^*$ maximizes $\varphi(\boldsymbol{w}^*; \sigma)$, and $(\boldsymbol{x}^*(\boldsymbol{w}^*), \boldsymbol{z}^*(\boldsymbol{w}^*))$ minimizes $\Phi(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{w}^*, \sigma)$, then $(\boldsymbol{x}^*(\boldsymbol{w}^*), \boldsymbol{z}^*(\boldsymbol{w}^*))$ is the optimal solution to problem (3.2).

To minimize $\Phi(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{w}, \sigma)$ for any fixed $\boldsymbol{w}$ and σ, we use the subgradient method because $\Phi(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{w}, \sigma)$ is a nonsmooth function. $(\boldsymbol{x}, \boldsymbol{z})$ is updated as

$$\begin{pmatrix} \boldsymbol{x}^{(t+1)}(\boldsymbol{w}) \\ \boldsymbol{z}^{(t+1)}(\boldsymbol{w}) \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}^{(t)}(\boldsymbol{w}) \\ \boldsymbol{z}^{(t)}(\boldsymbol{w}) \end{pmatrix} - \lambda^{(t)} \begin{pmatrix} \frac{\partial \Phi}{\partial \boldsymbol{x}^{(t)}(\boldsymbol{w})} \\ \frac{\partial \Phi}{\partial \boldsymbol{z}^{(t)}(\boldsymbol{w})} \end{pmatrix}$$

in the t^{th} iteration of the subgradient method. The subgradients are calculated as below.

$$\frac{\partial \Phi}{\partial x_i} = \begin{cases} \frac{\partial f(\boldsymbol{x},\boldsymbol{z},\boldsymbol{b})}{\partial x_i} + \frac{\partial f(\boldsymbol{x},\boldsymbol{z},\boldsymbol{b})}{\partial \boldsymbol{b}} \frac{\partial \boldsymbol{b}}{\partial x_i} + [w_i - \sigma h_i] - [w_{i-1} - \sigma h_{i-1}] & \text{if } h_i \leq \frac{w_i}{\sigma} \text{ and } h_{i-1} \leq \frac{w_{i-1}}{\sigma} \\ \frac{\partial f(\boldsymbol{x},\boldsymbol{z},\boldsymbol{b})}{\partial x_i} + \frac{\partial f(\boldsymbol{x},\boldsymbol{z},\boldsymbol{b})}{\partial \boldsymbol{b}} \frac{\partial \boldsymbol{b}}{\partial x_i} + [w_i - \sigma h_i] & \text{if } h_i \leq \frac{w_i}{\sigma} \text{ and } h_{i-1} > \frac{w_{i-1}}{\sigma} \\ \frac{\partial f(\boldsymbol{x},\boldsymbol{z},\boldsymbol{b})}{\partial x_i} + \frac{\partial f(\boldsymbol{x},\boldsymbol{z},\boldsymbol{b})}{\partial \boldsymbol{b}} \frac{\partial \boldsymbol{b}}{\partial x_i} - [w_{i-1} - \sigma h_{i-1}] & \text{if } h_i > \frac{w_i}{\sigma} \text{ and } h_{i-1} \leq \frac{w_{i-1}}{\sigma} \\ \frac{\partial f(\boldsymbol{x},\boldsymbol{z},\boldsymbol{b})}{\partial x_i} + \frac{\partial f(\boldsymbol{x},\boldsymbol{z},\boldsymbol{b})}{\partial \boldsymbol{b}} \frac{\partial \boldsymbol{b}}{\partial x_i} & \text{otherwise} \end{cases}$$

where $i = 2, \ldots, I-1$.

$$\frac{\partial \Phi}{\partial z_i} = \frac{\partial f(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{b})}{\partial z_i} + \sum_{u=1}^{I} \frac{\partial f(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{b})}{\partial b_u} \frac{\partial b_u}{\partial z_i}, \ i = 2, \ldots, I-1.$$

The stepsize $\lambda^{(t)}$ is chosen by the constant step length rule and thus "the subgradient algorithm is guaranteed to converge to within some range of the optimal value"[4]:

$$\lambda = d / ||(\frac{\partial \Phi}{\partial \boldsymbol{x}^{(t)}(\boldsymbol{w})}, \frac{\partial \Phi}{\partial \boldsymbol{z}^{(t)}(\boldsymbol{w})})||_2, \tag{3.9}$$

where d is a constant.

"Since the subgradient method is not a descent method, it is common to keep track of the best point found so far"[4]. The final solution of the subgradient method, $(\boldsymbol{x}^*(\boldsymbol{w}), \boldsymbol{z}^*(\boldsymbol{w}))$ is kept tracked as the one returns the lowest value of $\Phi(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{w}, \sigma)$ so far.

Note that the optimal $\boldsymbol{w}^*$ is not known in advance. Therefore, in the augmented Lagrangian method, we update $\boldsymbol{w}$ as

$$w_i^{(k+1)} = \max\left(0, w_i^{(k)} - \sigma h_i(\boldsymbol{x}^{(k)})\right),\ i = 1, \ldots, I-1, \tag{3.10}$$

in the k^{th} iteration. When $\boldsymbol{w}^{(k)}$ converges to $\boldsymbol{w}^*$, $(\boldsymbol{x}^*(\boldsymbol{w}^{(k)}), \boldsymbol{z}^*(\boldsymbol{w}^{(k)}))$ converges to $(\boldsymbol{x}^*(\boldsymbol{w}^*), \boldsymbol{z}^*(\boldsymbol{w}^*))$.

We increase the value of σ when $\boldsymbol{w}^{(k)}$ does not converge or converges too slowly ($\frac{||g(\boldsymbol{x}^{(k)})||}{||g(\boldsymbol{x}^{(k-1)})||}$ greater than a threshold, β). In such case, the parameter σ can be increased by a constant rate α, $\sigma := \alpha\sigma$.

The knot location optimization algorithm is as follows:

Knot Location Optimization Algorithm

Step 1. Initialization. Choose a starting feasible solution of knot location $\boldsymbol{x}^{(0)}$ and function value $\boldsymbol{z}^{(0)}$. Set initial penalty factor σ, initial multipliers $\boldsymbol{w}^{(1)}$, increasing rate of the penalty factor $\alpha(\alpha > 1)$, and converge rate threshold β. Set $k = 1$.

Step 2. New Solution. Use $(\boldsymbol{x}^{(k-1)}, \boldsymbol{z}^{(k-1)})$ as initial point and obtain new solution $(\boldsymbol{x}^{(k)}, \boldsymbol{z}^{(k)})$ by using subgradient method to solve the unconstrained nonlinear problem:

$$\min_{\boldsymbol{x},\boldsymbol{z}} \Phi(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{w}^{(k)}, \sigma).$$

Step 3. Local Optimum. If $||g(\boldsymbol{x}^{(k)})|| < \epsilon$, stop the algorithm, $(\boldsymbol{x}^{(k)}, \boldsymbol{z}^{(k)})$ is a local optimum; otherwise go to step 4.

Step 4. Update $\boldsymbol{\sigma}$. If $\frac{||g(\boldsymbol{x}^{(k)}||}{||g(\boldsymbol{x}^{(k-1)})||} \geq \beta, \sigma := \alpha\sigma$, and go to step 5; otherwise, go to step 5.

Step 5. Update $\boldsymbol{w}$. $w_i^{(k+1)} = \max\left(0, w_i^{(k)} - \sigma g_i(\boldsymbol{x}^{(k)})\right), i = 1, ..., I-1$. Set $k := k+1$ and go to step 2.

The pseudo-code is as follows:

Knot Location Optimization Algorithm

Start
$k \leftarrow 1$
$\sigma \leftarrow$ *Initial Penalty Factor*
$\boldsymbol{w}^{(1)} \leftarrow$ *Initial Multipliers*
$\alpha \leftarrow$ *Increasing Rate of the Penalty factor*
$\beta \leftarrow$ *Converge Rate Threshold*
$\mathbf{x}^{(0)} \leftarrow$ *Initial Knot Location*
Calculate:
  $\mathbf{z}^{(0)} \leftarrow$ *Initial Function Value at Knots*
loop:
  Obtain:
    $(\boldsymbol{x}^{(k)}, \boldsymbol{z}^{(k)}) \leftarrow$ *Solution of* $\min_{\boldsymbol{x},\boldsymbol{z}} \Phi(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{w}^{(k)}, \sigma)$
  Check:
    If $||g(\boldsymbol{x}^{(k)})|| > \epsilon$ **then**
      If $\frac{||g(\boldsymbol{x}^{(k)}||}{||g(\boldsymbol{x}^{(k-1)})||} \geq \beta$ **then**
        Assign:
          $\sigma \leftarrow \alpha\sigma$
        Calculate:
          $w_i^{(k+1)} \leftarrow \max(0, w_i^{(k)} - \sigma g_i(\boldsymbol{x}^{(k)})), i = 1, ..., I-1$
        Assign:
          $k \leftarrow k+1$
        go to *loop*
      Else
        Calculate:
          $w_i^{(k+1)} \leftarrow \max(0, w_i^{(k)} - \sigma g_i(\boldsymbol{x}^{(k)})), i = 1, ..., I-1$
        Assign:
          $k \leftarrow k+1$
        go to *loop*
    Else
      Record:
        $(\boldsymbol{x}^{(k)}, \boldsymbol{z}^{(k)}) \leftarrow$ *Local Optimum*
End loop
Stop

3.2 Determine Knot Number and Location

Consider fitting given data $(\hat{x}_m, \hat{z}_m)$ $(m = 1, 2, ...M)$, when $I, \boldsymbol{x}, \boldsymbol{z}$, and $\boldsymbol{b}$ are unknown. Without loss of generality, we fix $x_1 = \hat{x}_1$ and $x_I = \hat{x}_M$.

We determine the knot number and location by proposing a heuristic method. The method starts with a small number of evenly spaced knots. Then, the knots are optimized by the knot location optimization algorithm. The location that has the biggest change of knot will be inserted in one knot. These procedures are repeated until a stopping criterion is met.

The specific procedures of the heuristic method are:

Choose a starting knots: $\{v_1^{(0)}, v_2^{(0)}, \ldots, v_{I_0}^{(0)}\}$.

In the t^{th} iteration of the heuristic method, the initial knots $\boldsymbol{x}^{(t)}$ consist of two types of knots: Type 1 knots is the set of knots added in the process of the heuristic method, denoted as $u_i^{(t)}, i = 1, 2, \ldots, K_t$. Type 2 knots is the set of preexisting knots, with location adjusted at the beginning of each iteration, denoted as $v_j^{(t)}, j = 1, 2, \ldots, I_0$.

In the t^{th} iteration, we denote: initial knots: $\boldsymbol{x}^{(t)} = \{u_i^{(t)}\} \bigcup \{v_j^{(t)}\}$, knots after location optimized: $\boldsymbol{x}^{(*,t)}$.

Changes of knots are calculated: $\Delta_i^{(t)} = |x_i^{(*,t)} - x_i^{(t)}|$. Let $i_{\max}^{(t)} = \arg\max \Delta_i^{(t)}$.

If $x_{i_{\max}^{(t)}}^{(*,t)}$ is a type 1 knot, we do not add any new knot. Thus, type 1 knots in $(t+1)^{th}$ iteration are $\{u_i^{(*,t)}\}$.

Otherwise, $x_{i_{\max}^{(t)}}^{(*,t)}$ is a type 2 knot, we add a new knot at $x_{i_{\max}^{(t)}}^{(*,t)}$ to the type 1 knots. Thus, type 1 knots in $(t+1)^{th}$ iteration are $\{u_i^{(*,t)}\} \bigcup \{x_{i_{\max}^{(t)}}^{(*,t)}\}$.

Denote type 1 knots in $(t+1)^{th}$ iteration as $_iu^{(t+1)}, i = 1, 2, \ldots, K_{t+1}$. In each interval $[u_i^{(t+1)}, u_{i+1}^{(t+1)})$, count the number $_ip^{(t+1)}$ of initial location of preexisting knots, $\{v_1^{(0)}, v_2^{(0)}, \ldots, _Iv^{(0)}\}$.

Then update type 2 knots by setting $_iv_q^{(t+1)} = u_i^{(t+1)} + \frac{q}{p_i^{(t+1)}+1}(u_{i+1}^{(t+1)} - u_i^{(t+1)})$ for $i = 1, 2, \ldots, K_{t+1}$ and $q = 1, 2, \ldots, _ip^{(t+1)}$.

The initial location of $\boldsymbol{x}$ at the $(t+1)^{th}$ iteration is update as

$$\boldsymbol{x}^{(t+1)} = \{u_i^{(t+1)}\} \bigcup \{v_j^{(t+1)}\}, \tag{3.11}$$

where $i = 1, 2, \ldots, K_{t+1}$ and $j = 1, 2, \ldots, I_0$.

The iterations of the heuristic method stop when the fitting error ($F(\boldsymbol{x}, \boldsymbol{z})$) of the cubic L^1 spline fit converges. $F(\boldsymbol{x}, \boldsymbol{z})$ can sometimes increase and sometimes decrease with iterations of heuristic method. Thus, we intuitively decide that the convergence occurs when there is sufficiently small difference between the two consecutive average of objective function values of equation (3.2):

$$|\frac{E^{(t)} - E^{(t-1)}}{E^{(t-1)}}| < \eta,$$

where η is a small number and

$$E^{(t)} = \frac{F^{(t-9)} + F^{(t-8)} + ... + F^{(t)}}{p}.$$

p is a positive integer. $F(\boldsymbol{x}, \boldsymbol{z})$ represents the objective function value (fitting error of the cubic L^1 spline fit) in equation (3.2).

3.3 Determine Change Point(s)

Consider finding points of abrupt change of magnitude in given data $(\hat{x}_m, \hat{z}_m)$ $(m = 1, 2, ...M)$.

Our change point detection method firstly runs the heuristic method to determine a spline fit to approximate the data. Obtained knot location and obtained function value at knots are denoted as $\{x_1^*, x_2^*, \ldots, x_{I^*}^*\}$ and $\{z_1^*, z_2^*, \ldots, z_{I^*}^*\}$, respectively.

Then, the absolute difference between the function value of every two consecutive knots is calculated: $\Delta_i = |z_{i+1}^* - z_i^*|$. The differences are ranked. Let $i_{\max} = \arg\max \Delta_i$. The largest difference indicates the highest chance of occurrence of change point, e.g. the most possible change point is located in the interval $[x_{i_{\max}}, x_{i_{\max}+1}]$.

CHAPTER 4

EXPERIMENTS AND RESULTS

We conduct numerical experiments to test the effectiveness of proposed methods in Chapter 3. This chapter presents all the experimental results of this research. All experiments are conducted in Northern Illinois University "AnywhereApp" computing environment with MATLAB 2018b. In subsection 4.1, cubic L^1 spline fits with free-location knots are compared with cubic L^1 spline fits with fixed-location knots [33]. In subsection 4.2, the spline fits with free number and location of knots are compared with the spline fits with free location of knots and given number of knots. In subsection 4.3, application of cubic L^1 spline fits in change point detection is experimented in an artificial dataset and a real one.

4.1 Knot Location Optimization

This section presents five experiments. Five datasets with different shapes are used to see how good approximation can be by using the cubic L^1 spline fits with knots optimized. In experiments 1, 2, and 3, we use datasets 1, 2, and 3 in [33], respectively. In experiments 4 and 5, we use artificial datasets 4 and 5, respectively. In the five experiments, we use our knot location optimization algorithm and compare the experimental results including approximation error and shape preservation with the results by the method in [33]. In [33], knot number and location are predetermined and fixed.

In the first three experiments, the starting knot location is chosen as the knot location used by [33]. In the last two experiments, we use two settings of initial knot location to study

the algorithm's sensitivity to the initial solution. In all experiments, we choose the initial function value from the linear spline [33]. We set initial penalty factor σ, initial multipliers $w_i, i = 1, 2, \ldots, I-1$, increasing rate of the penalty factor α, converge rate threshold, β, stopping creation threshold of augmented Lagrangian method ϵ, and constant step length of subgradient method d as: $\sigma = 2, w_i = 1,\ i = 1, 2, \ldots, I-1, \alpha = 1.5, \beta = 0.5, \epsilon = 10^{-5}, d = 0.005$.

Experiment 1

In [33], for dataset 1,

$$\begin{aligned} \boldsymbol{x} &= (0.00, 1.00, 2.00, 3.00, 4.00, 5.00, 5.20, 5.40, 5.60, 5.80, 6.00, 6.20, 6.40, 6.60, 6.80, 7.00, 7.20) \\ I &= 17 \\ M &= 81 \end{aligned}$$

The dataset $\{(\hat{x}_m, \hat{z}_m)\}$ is in the Appendix.

Knots after location optimization are:

$$\boldsymbol{x}^{(*)} = (0.00, 1.00, 2.00, 3.65, 4.09, 4.76, 5.13, 5.40, 5.44, 5.73, 5.98, 6.20, 6.40, 6.60, 6.80, 7.00, 7.20).$$

Approximation errors of the cubic L^1 spline fits obtained by the method in [33] and the knot location optimization method are 3.81 and 0.42, respectively.

In Figure 4.1, the given data points are plotted as "+". The cubic L^1 spline fits obtained by the method in [33] and the knot location optimization method are plotted as solid curves in blue and red, respectively. Knots are represented by circles.

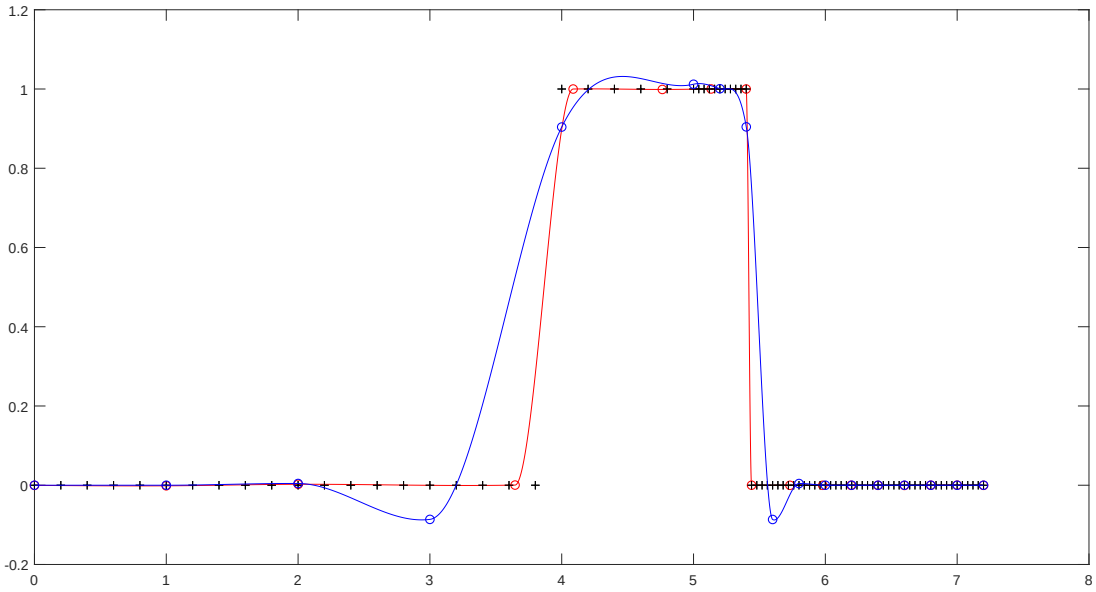

Figure 4.1: Cubic L^1 spline fits for dataset 1 obtained by the method in [33] (in blue) vs the knot location optimization (in red).

After the knot location is optimized, knot x_4 moves from 3.00 to 3.65 while knot x_5 moves from 4.00 to 4.09 and knot x_9 moves from 5.60 to 5.44. These changes enable the spline fits to better preserve the shapes in the corresponding areas. The approximation error of the spline fit decreases from 3.81 to 0.42.

Experiment 2

In [33], in dataset 2

$$\boldsymbol{x} = (0.00, 1.00, 2.00, 3.00, 3.50, 4.00, 4.50, 5.00, 5.50, 6.00, 6.50, 7.00, 8.00, 9.00, 10.00, 11.00, 12.00)$$

$$I = 17$$

$$M = 117$$

The dataset $\{(\hat{x}_m, \hat{z}_m)\}$ is in the Appendix.

Knots after location optimization are:

$$\boldsymbol{x}^{(*)} = (0.00, 1.00, 2.06, 2.87, 3.95, 4.00, 4.50, 4.94, 5.45, 6.03, 6.44, 6.94, 7.88, 8.75, 10.00, 11.00, 12.00).$$

Approximation errors of the cubic L^1 spline fits obtained by the method in [33] and the knot location optimization method are 13.88 and 5.15, respectively.

In Figure 4.2, the given data points are plotted as "+". The cubic L^1 spline fits obtained by the method in [33] and the knot location optimization method are plotted as solid curves in blue and red, respectively. Knots are represented by circles.

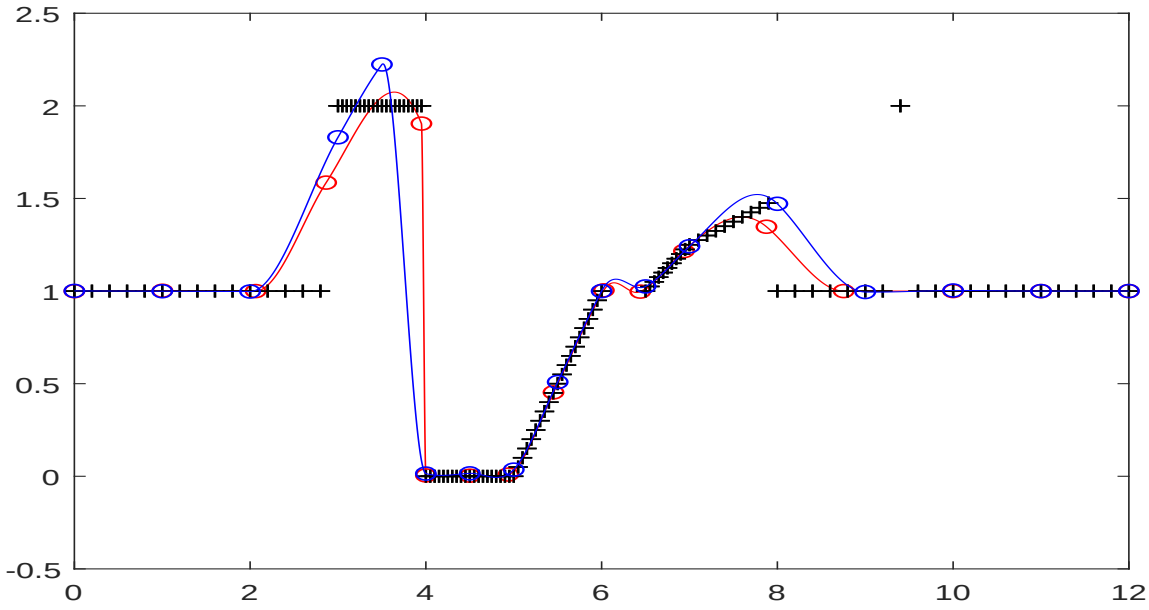

Figure 4.2: Cubic L^1 spline fits for dataset 2 obtained by the method in [33] (in blue) vs the knot location optimization (in red).

After the knot location is optimized, knot x_5 moves from 3.50 to 3.95 whereas knot x_{14} moves from 9.00 to 8.75. These changes enable the spline fit to better preserve the shapes in the corresponding areas. The approximation error of the spline fit decreases from 13.88 to 5.15.

Experiment 3

In [33], in dataset 3,

$$\begin{aligned} \boldsymbol{x} &= (0.00, 10.00, 20.00, 30.00, 35.00, 40.00, 45.00, 50.00, 55.00, 60.00, 65.00, 70.00, 75.00, 80.00) \\ I &= 14 \\ M &= 81 \end{aligned}$$

The dataset $\{(\hat{x}_m, \hat{z}_m)\}$ is in the Appendix.

Knots after location optimization are:

$$\boldsymbol{x}^{(*)} = (0.00, 10.06, 19.93, 28.81, 32.48, 40.21, 44.92, 50.18, 54.98, 60.03, 65.16, 69.98, 75.00, 80.00).$$

Approximation errors of the cubic L^1 spline fits obtained by the method in [33] and the knot location optimization method are 63.42 and 40.64, respectively.

In Figure 4.3, the given data points are plotted as "+". The cubic L^1 spline fits obtained by the method in [33] and the knot location optimization method are plotted as solid curves in blue and red, respectively. Knots are represented by circles.

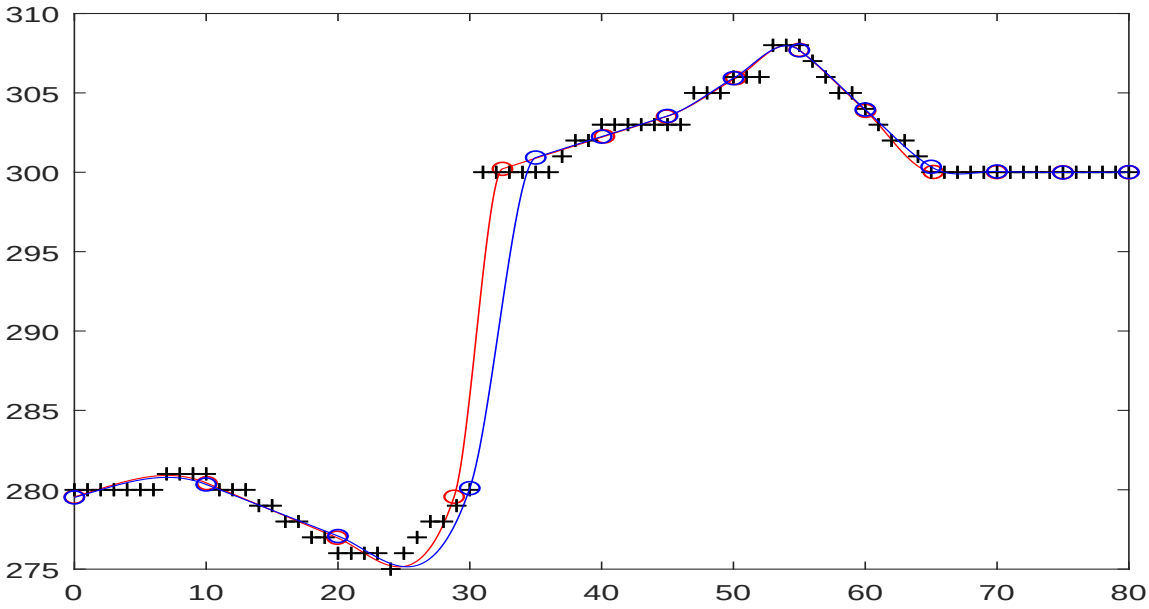

Figure 4.3: Cubic L^1 spline fits for dataset 3 obtained by the method in [33] (in blue) vs the knot location optimization (in red).

After the knot location is optimized, knot x_4 moves from 30.00 to 28.81 whereas knot x_5 moves from 35.00 to 32.48. These changes enable the spline fit to better preserve the shapes in the corresponding areas. The approximation error of the spline fit decreases from 63.42 and 40.64.

Experiment 4

We use an artificial data set, dataset 4, for this experiment. The data are in two straight line segments of which the first is increasing and the second decreasing. The data is:

$$\begin{aligned}
\hat{x}_m &= (0.00, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 1.00, 1.10, 1.20, 1.30, 1.40, 1.50, \\
&\quad 1.60, 1.70, 1.80, 1.90, 2.00, 2.10, 2.20, 2.30, 2.40, 2.50, 2.60, 2.70, 2.80, 2.90, 3.00, 3.10, \\
&\quad 3.20, 3.30, 3.40, 3.50, 3.60, 3.70, 3.80, 3.90, 4.00) \\
\hat{z}_m &= (0.500, 0.575, 0.650, 0.725, 0.800, 0.875, 0.950, 1.025, 1.100, 1.1750, 1.250, 1.325, 1.400, \\
&\quad 1.475, 1.550, 1.625, 1.700, 1.775, 1.850, 1.925, 2.000, 1.925, 1.850, 1.775, 1.700, 1.625, \\
&\quad 1.550, 1.475, 1.400, 1.325, 1.250, 1.175, 1.100, 1.025, 0.950, 0.875, 0.800, 0.725, 0.650, \\
&\quad 0.575, 0.50) \\
M &= 41
\end{aligned}$$

This experiment aims to study shape-preserving capability of the knot location optimization algorithm and its sensitivity to initial knot location. For this purpose, we use two settings of the initial location of knots.

In the first setting, we arbitrarily choose one initial knot located in the second line segment and five initial knots located in the first line segment. The purpose is to see how the asymmetrically spaced knots can be optimized to construct a spline fit to approximate the symmetrically spaced data. The first initial location setting for knots:

$$\begin{aligned}
\boldsymbol{x} &= (0.00, 0.40, 0.60, 1.00, 1.60, 4.00) \\
I &= 6
\end{aligned}$$

Knots after location optimization are:

$$\boldsymbol{x}^{(*)} = (0.00, 0.47, 0.63, 0.84, 1.58, 4.00).$$

Approximation errors of the cubic L^1 spline fits obtained by the method in [33] and the knot location optimization method are 5.69 and 8.39, respectively.

In Figure 4.4, the given data points are plotted as "+". The cubic L^1 spline fits obtained by the method in [33] and the knot location optimization method are plotted as solid curves in blue and red, respectively. Knots are represented by circles.

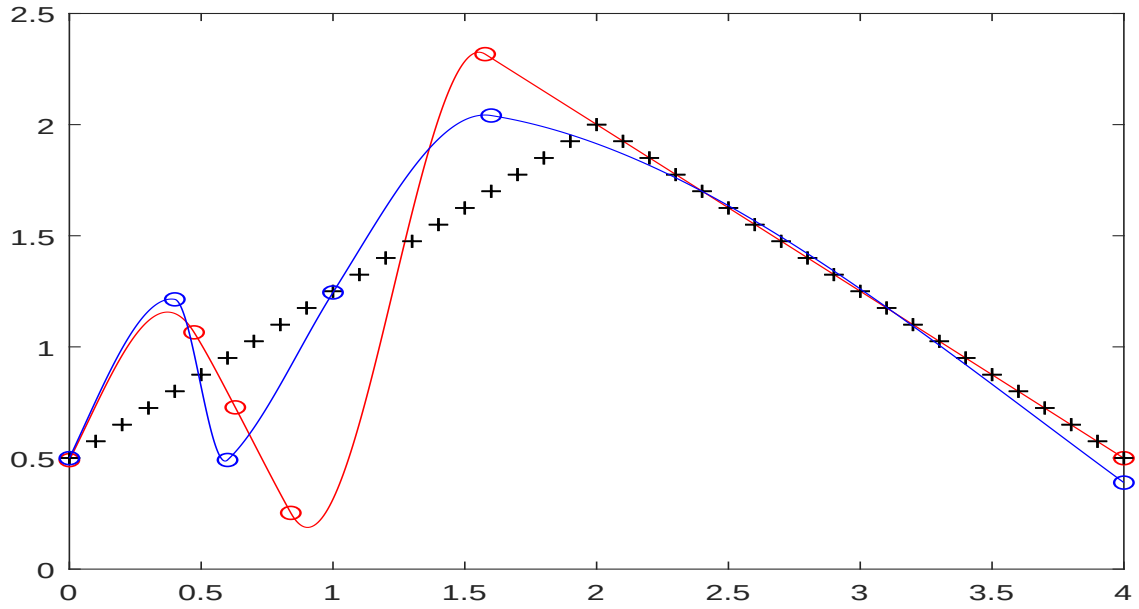

Figure 4.4: Cubic L^1 spline fits for dataset 4 obtained by the method in [33] (in blue) vs the knot location optimization (in red) starts with the first initial knot location setting.

Even though the approximation error of the spline fits increases from 5.69 to 8.39 after knot location is optimized, the shape is better reserved in the second line segment.

The second initial location setting for knots:

$$\begin{aligned} \boldsymbol{x} &= (0.00, 0.80, 1.60, 2.40, 3.20, 4.00) \\ I &= 6 \end{aligned}$$

The knots are evenly spaced in the second initial location setting while keeping the number of knots same as in the first initial location setting.

Knots after location optimization are:

$$\boldsymbol{x}^{(*)} = (0.00, 0.80, 1.77, 2.23, 3.20, 4.00).$$

Approximation errors of the cubic L^1 spline fits obtained by the method in [33] and the knot location optimization method are 0.41 and 0.19, respectively.

In Figure 4.5, the given data points are plotted as "+". The cubic L^1 spline fits obtained by the method in [33] and the knot location optimization method are plotted as solid curves in blue and red, respectively. Knots are represented by circles.

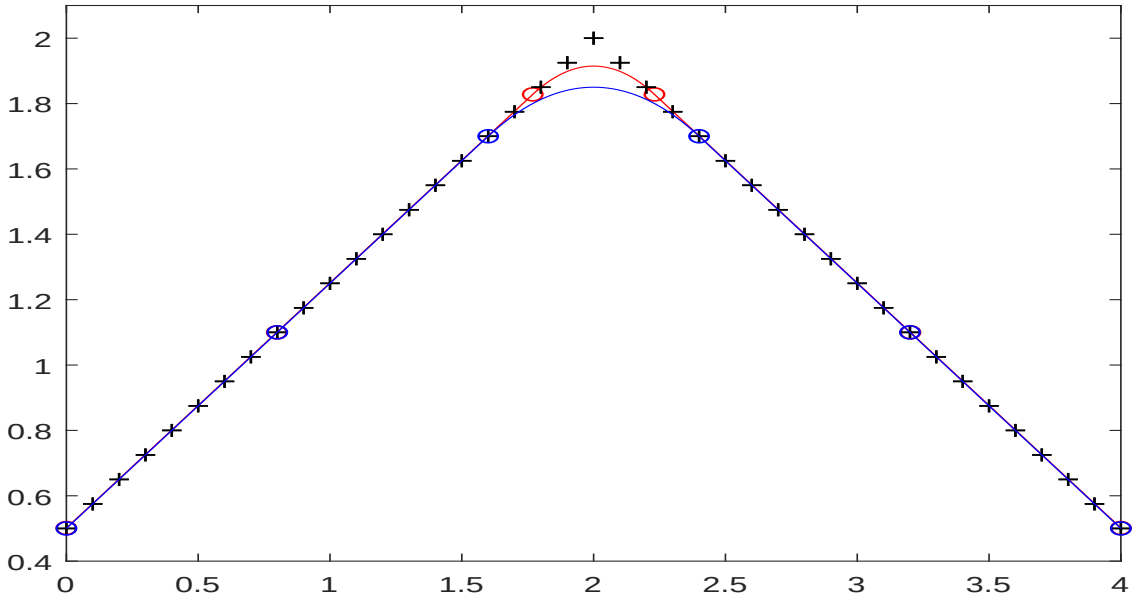

Figure 4.5: Cubic L^1 spline fits for dataset 4 obtained by the method in [33] (in blue) vs the knot location optimization (in red) starts with the second knot initial location setting.

After knot location is optimized, knot x_3 moves from 1.60 to 1.77 while knot x_4 moves from 2.40 to 2.23. The change enables the spline fits to approximate the data with better shape-preservation. The approximation error of the spline fit decreases from 0.41 and 0.19.

In Experiment 4, the comparison between spline fits with knots optimized from the two initial settings shows that a better choice of the initial knot location setting can result in a better local optimum of the spline fits.

Experiment 5

We use an artificial data set, dataset 5, for this experiment. The data are in two parallel straight line segments. The data is:

$$\begin{aligned}
\hat{x}_m &= (0.00, 0.20, 0.40, 0.60, 0.80, 1.00, 1.20, 1.40, 1.60, 1.80, 2.00, 2.20, 2.40, 2.60, 3.20, 3.40, \\
&\quad 3.60, 3.80, 4.00, 4.20, 4.40, 4.60, 4.80, 5.00, 5.20, 5.40, 5.60, 5.80, 6.00, 6.20, 6.40) \\
\hat{z}_m &= (1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 2.00, 2.00, \\
&\quad 2.00, 2.00, 2.00, 2.00, 2.00, 2.00, 2.00, 2.00, 2.00, 2.00, 2.00, 2.00, 2.00, 2.00, 2.00) \\
M &= 31
\end{aligned}$$

This experiment aims to study shape-preserving capability of knot location optimization algorithm and its sensitivity to initial knot location.

In the first setting, we arbitrarily choose one initial knot located in the second line segment and six initial knots located in the first line segment. The purpose is to see how the asymmetrically spaced knots can be optimized to construct a spline fit to approximate the data. The first initial location setting for knots:

$$\begin{aligned}
\boldsymbol{x} &= (0.00, 0.50, 1.00, 1.50, 2.00, 2.50, 6.40) \\
I &= 7
\end{aligned}$$

Knots after location optimization are:

$$\boldsymbol{x}^{(*)} = (0.00, 0.64, 1.32, 1.66, 1.98, 3.09, 6.40).$$

Approximation errors of the cubic L^1 spline fits obtained by the method in [33] and the knot location optimization method are 4.37 and 1.44, respectively.

In Figure 4.6, the given data points are plotted as "+". The cubic L^1 spline fits obtained by the method in [33] and the knot location optimization method are plotted as solid curves in blue and red, respectively. Knots are represented by circles.

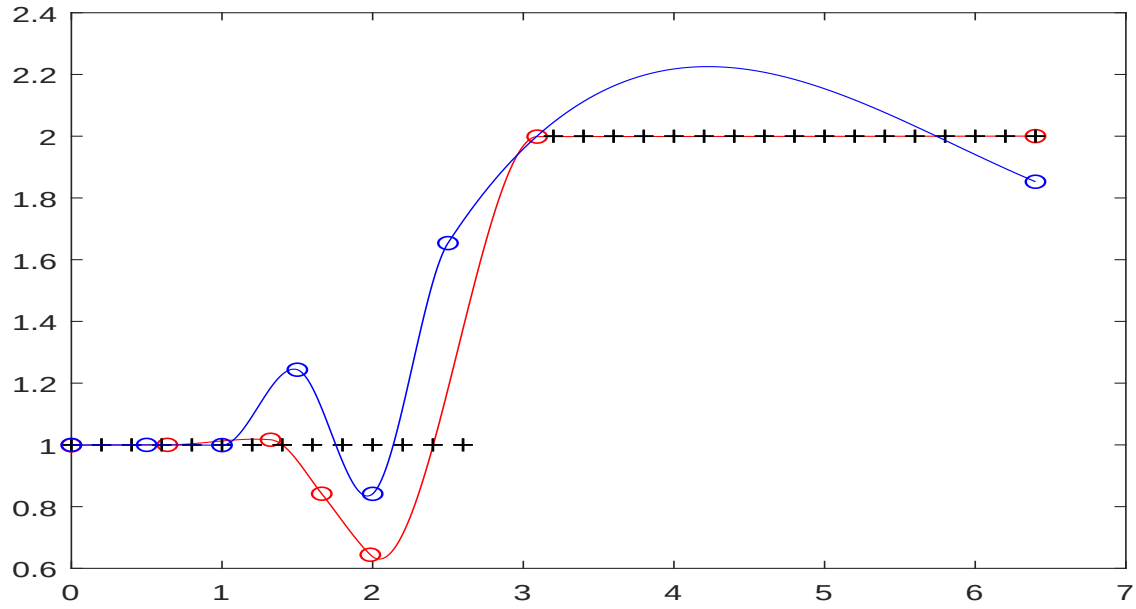

Figure 4.6: Cubic L^1 spline fits for dataset 5 obtained by the method in [33] (in blue) vs the knot location optimization (in red) starts with the first initial knot location setting.

After knot location is optimized, knot x_5 moves from 2.50 to 3.09. The change enables the spline fit to better preserve the shape of the second line segment. The approximation error of the spline fit decreases from 4.37 to 1.44.

The second initial location setting for knots::

$$\begin{aligned} \boldsymbol{x} &= (0.00, 1.07, 2.13, 3.20, 4.27, 5.33, 6.40) \\ I &= 7 \end{aligned}$$

The knots are evenly spaced in the second initial location setting while keeping the number of knots same as in the first initial location setting.

Knots after location optimization are:

$$\boldsymbol{x}^{(*)} = (0.00, 1.07, 2.58, 3.21, 4.27, 5.33, 6.40).$$

Approximation errors of the cubic L^1 spline fits obtained by the method in [33] and the knot location optimization method are 0.57 and 0.02, respectively.

In Figure 4.7, the given data points are plotted as "+". The cubic L^1 spline fits obtained by the method in [33] and the knot location optimization method are plotted as solid curves in blue and red, respectively. Knots are represented by circles.

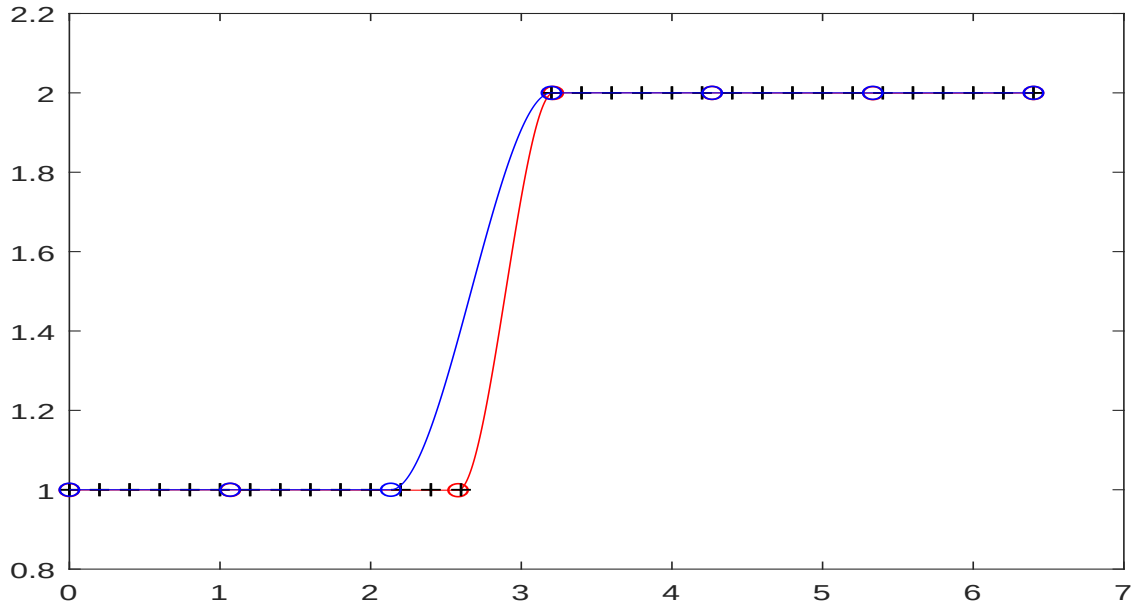

Figure 4.7: Cubic L^1 spline fits for dataset 5 obtained by the method in [33] (in blue) vs the knot location optimization (in red) starts with the second initial knot location setting.

After knot location is optimized, knot x_3 moves from 2.13 to 2.58. The change enables the spline fit to better preserve the shapes of the data. The approximation error of the spline fit decreases from 0.57 to 0.02.

In Experiment 5, comparison between the spline fits with knots optimized from the two initial settings shows that a better choice of the initial knot location setting can result in a better local optimum of the spline fits.

In conclusion, the experimental results in this subsection show that the knots of the cubic L^1 spline fits can be optimized to the shape change areas. The knot location optimization method can help the cubic L^1 spline fits to better preserve shapes of data with smaller fitting error. Additionally, performance of knot location optimization algorithm could depend on the choice of the initial location of knots. Due to non-convexity of the objective function in the knot location optimization model, the local optimum depends on the choice of the initial location of knots.

4.2 Knot Number and Location Determination

This section presents five experiments using the same five datasets as in the subsection 4.1. Datasets 1 to 5 are used in experiments 6 to 10, respectively. We use the heuristic method and compare the experimental results including approximation error and shape preservation with the results by the knot location optimization algorithm from subsection 4.1. In experiments 6 to 8, we compare splines calculated by the knot number and location determination heuristic method starting from a small number of knots with splines calculated by the knot location optimization algorithm starting from predetermined I and $\boldsymbol{x}$ in [33]. The latter are showed in subsection 4.1. In experiments 9 to 10, we compare splines calculated by the heuristic method starting from a small number of knots with splines calculated by the knot location optimization algorithm starting from the second initial knot location setting in subsection 4.1. The latter are showed in subsection 4.1.

To apply the heuristic method, in Experiment 6, we use six evenly spaced knot location as the initial knot location for cubic L^1 spline fits to approximate dataset 1. In Experiment 7 to 10, we use five evenly spaced knot location as the initial knot location for the cubic L^1 spline fits to approximate datasets 2 to 5, respectively. In all experiments, we choose the initial function value at knots from the linear spline [33]. The stoping criterion η and number of compared objective function value p are set as: $\eta = 10^{-4}, p = 10$. The rest of parameters are set the same as experiments in subsection 4.1.

Experiment 6

Experiment 6 uses dataset 1. The initial knots are:

$$\begin{aligned} \boldsymbol{x} &= (0.00, 1.44, 2.88, 4.32, 5.76, 7.20) \\ I &= 6 \end{aligned}$$

Knots after the knot number and location determination method are:

$$\begin{aligned} \boldsymbol{x}^{(*)} &= (0, 1.06, 2.12, 3.18, 3.79, 4.01, 4.44, 4.89, 5.40, 5.44, 6.32, 7.20) \\ I^{(*)} &= 12. \end{aligned}$$

Recall the knots after location optimization in the subsection 4.1 are:

$$\boldsymbol{x}^{(*)} = (0.00, 1.00, 2.00, 3.65, 4.09, 4.76, 5.13, 5.40, 5.44, 5.73, 5.98, 6.20, 6.40, 6.60, 6.80, 7.00, 7.20),$$

where the number of knots is predetermined as 17:

$$I = 17.$$

Approximation errors of the cubic L^1 spline fits obtained by the knot location optimization method and the knot number and location determination method are 0.42 and 0.02, respectively.

In Figure 4.8, the given data points are plotted as "+". The cubic L^1 spline fits obtained by the knot location optimization method and the knot number and location determination method are plotted as solid curves in blue and red, respectively. Knots are represented by circles.

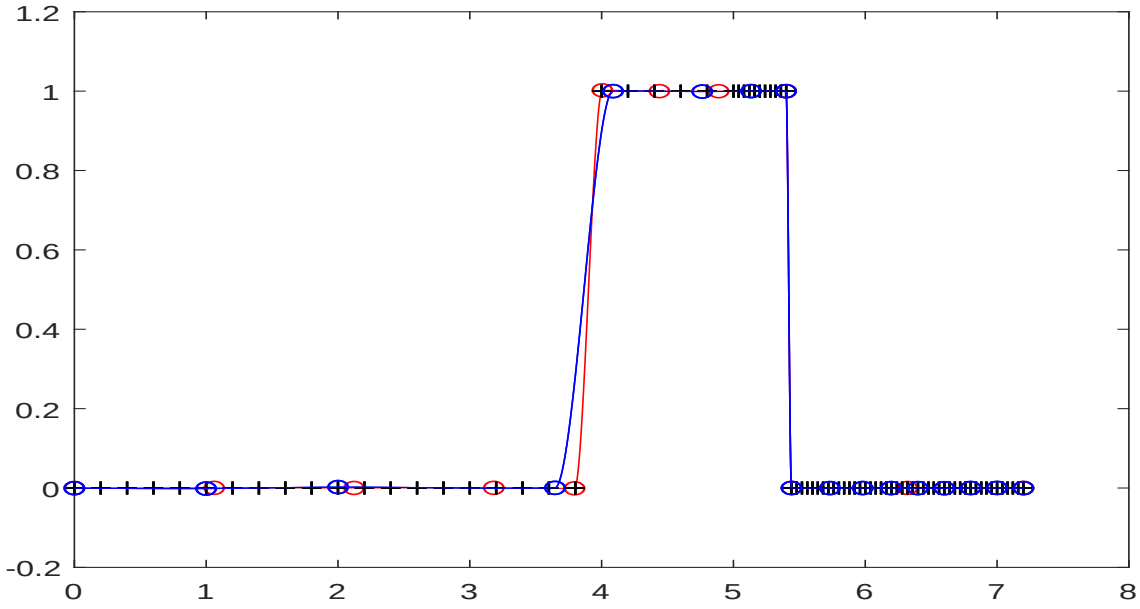

Figure 4.8: Cubic L^1 spline fits for dataset 1 obtained by the knot location optimization (in blue) vs the knot number and location determination (in red).

The approximation error of the spline fit decreases from 0.42 to 0.02 after the knot number is optimized. The spline fit also better preserves the shapes of the data with the number of the knots optimized. Using the knot location and number determination method, we can use fewer knots than [33] in the spline fit to approximate the dataset with smaller fitting error and better shape-preservation.

Experiment 7

Experiment 7 uses dataset 2. The initial knots used are:

$$\begin{aligned} \boldsymbol{x} &= (0.00, 3.00, 6.00, 9.00, 12.00) \\ I &= 5 \end{aligned}$$

Knots after the knot number and location determination are:

$$\begin{aligned} \boldsymbol{x}^{(*)} &= (0.00, 1.42, 1.90, 2.29, 2.58, 2.80, 3.00, 3.48, 3.95, 4.00, 4.47, 4.93, 5.25, 5.65, 6.01, 6.37, \\ &\quad 6.7295, 7.08, 7.49, 7.90, 8.00, 8.81, 9.61, 12.00) \\ I^{(*)} &= 24 \end{aligned}$$

Recall the knots after location optimization in the subsection 4.1 are:

$$\boldsymbol{x}^{(*)} = (0.00, 1.00, 2.06, 2.87, 3.95, 4.00, 4.50, 4.94, 5.45, 6.03, 6.44, 6.94, 7.88, 8.75, 10.00, 11.00, 12.00),$$

where the number of knots is predetermined as 17:

$$I = 17.$$

Approximation errors of the cubic L^1 spline fits obtained by the knot location optimization method and the knot number and location determination method are 5.15 and 1.08, respectively.

In Figure 4.9, the given data points are plotted as "+". The cubic L^1 spline fits obtained by the knot location optimization method and the knot number and location determination are plotted as solid curves in blue and red, respectively. Knots are represented by circles.

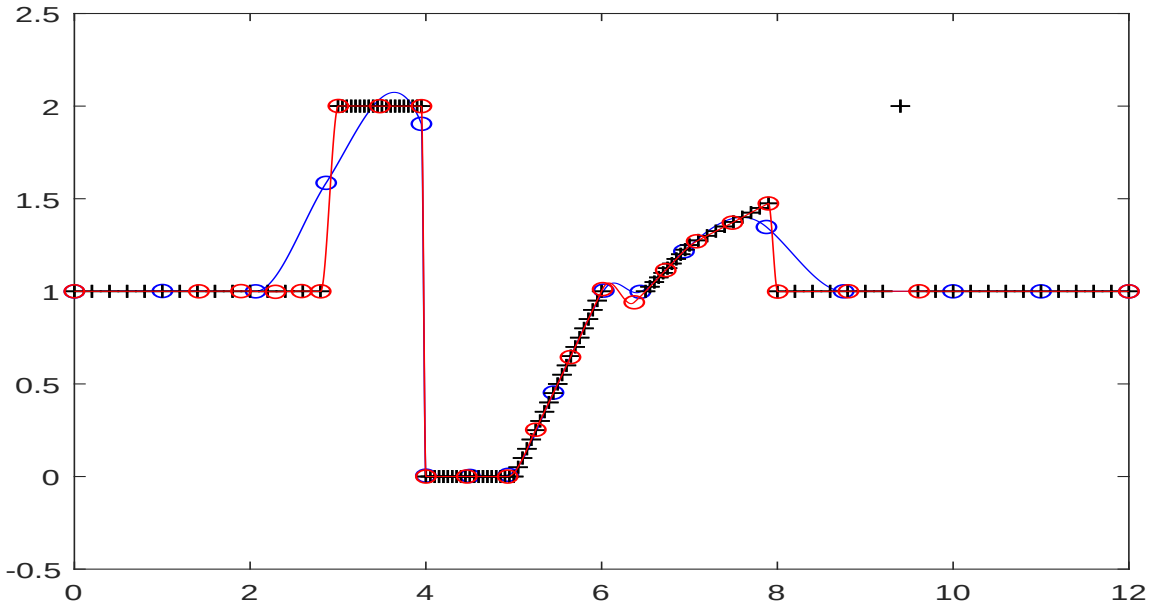

Figure 4.9: Cubic L^1 spline fits for dataset 2 obtained by the knot location optimization (in blue) vs the knot number and location determination (in red).

The approximation error of the spline fit decreases from 5.15 to 1.08 after the knot number is optimized. The spline fit also better preserves the shapes of the data with the number of the knots optimized.

Experiment 8

Experiment 8 uses dataset 3. The initial knots used are:

$$\begin{aligned} \boldsymbol{x} &= (0.00, 20.00, 40.00, 60.00, 80.00) \\ I &= 5 \end{aligned}$$

Knots after the knot number and location determination are:

$$\begin{aligned} \boldsymbol{x}^{(*)} &= (0.00, 12.14, 16.30, 18.64, 20.86, 22.34, 29.84, 31.00, 34.96, 39.20, 40.79, 42.43, 46.49, \\ & \quad 54.98, 57.70, 60.67, 67.11, 80.00) \\ I^{(*)} &= 18 \end{aligned}$$

Recall the knots after location optimization in the subsection 4.1 are:

$$\boldsymbol{x}^{(*)} = (0.00, 10.06, 19.93, 28.81, 32.48, 40.21, 44.92, 50.18, 54.98, 60.03, 65.16, 69.98, 75.00, 80.00),$$

where the number of knots is predetermined as 14:

$$I = 14.$$

Approximation errors of the cubic L^1 spline fits obtained by the knot location optimization method and the knot number and location determination method are 40.64 and 20.05, respectively.

In Figure 4.10, the given data points are plotted as "+". The cubic L^1 spline fits obtained by the knot location optimization method and the knot number and location determination are plotted as solid curves in blue and red, respectively. Knots are represented by circles.

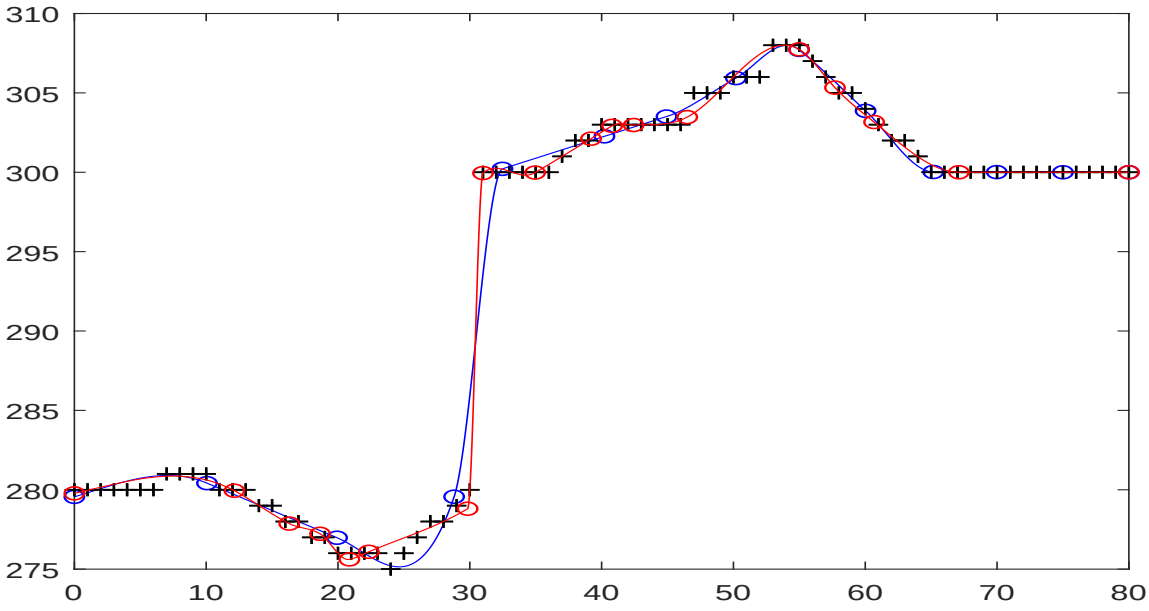

Figure 4.10: Cubic L^1 spline fits for dataset 3 obtained by the knot location optimization (in blue) vs the knot number and location determination (in red).

The approximation error of the spline fit decreases from 40.64 to 20.05 after the knot number is optimized. The spline fit also better preserves the shapes of the data with the number of the knots optimized.

Experiment 9

Experiment 9 uses dataset 4. The initial knots used are:

$$\begin{aligned} \boldsymbol{x} &= (0.00, 1.00, 2.00, 3.00, 4.00) \\ I &= 5 \end{aligned}$$

Knots after the knot number and location determination are:

$$\begin{aligned} \boldsymbol{x}^{(*)} &= (0.00, 0.95, 1.89, 2.35, 2.81, 3.41, 4.00) \\ I^{(*)} &= 7 \end{aligned}$$

Recall the knots after location optimization in the subsection 4.1 (starts with the second type of initial location) are:

$$\boldsymbol{x}^{(*)} = (0.00, 0.80, 1.77, 2.23, 3.20, 4.00),$$

where the number of knots is predetermined as 6:

$$I = 6.$$

Approximation errors of the cubic L^1 spline fits obtained by the knot location optimization method and the knot number and location determination method are 0.19 and 0.06, respectively.

In Figure 4.11, the given data points are plotted as "+". The cubic L^1 spline fits obtained by the knot location optimization method and the knot number and location determination are plotted as solid curves in blue and red, respectively. Knots are represented by circles.

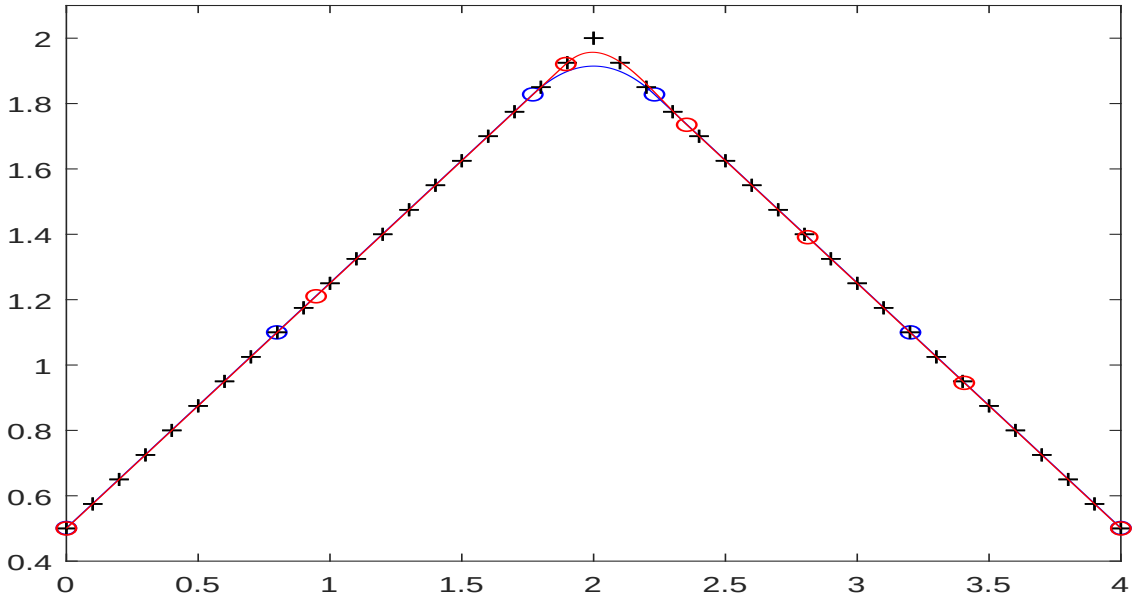

Figure 4.11: Cubic L^1 spline fits for dataset 4 obtained by the knot location optimization (starts with the second initial knot location setting)(in blue) vs the knot number and location determination (in red).

The approximation error of the spline fit decreases from 0.19 and 0.06 after the knot number is optimized. The spline fit also better preserves the shapes of the data with the number of the knots optimized. Using the knot location and number determination method, we can approximate the dataset with smaller fitting error and preserve the shapes better.

Experiment 10

Experiment 10 uses dataset 5. The initial knots used are:

$$\begin{aligned} \boldsymbol{x} &= (0.00, 1.60, 3.20, 4.80, 6.40) \\ I &= 5 \end{aligned}$$

Knots after the knot number and location determination are:

$$\begin{aligned} \boldsymbol{x}^{(*)} &= (0.00, 1.29, 2.58, 3.16, 3.74, 5.07, 6.40) \\ I^{(*)} &= 7 \end{aligned}$$

Recall the knots after location optimization in the subsection 4.1 (starts with the second type of initial location) are:

$$\boldsymbol{x}^{(*)} = (0.00, 1.07, 2.58, 3.21, 4.27, 5.33, 6.40),$$

where the number of knots is predetermined as 7:

$$I = 7.$$

Approximation errors of the cubic L^1 spline fits obtained by the knot location optimization method and the knot number and location determination method are 0.02 and 0.00, respectively.

In Figure 4.12, the given data points are plotted as "+". The cubic L^1 spline fits obtained by the knot location optimization method and knot number and location determination are plotted as solid curves in blue and red, respectively. Knots are represented by circles.

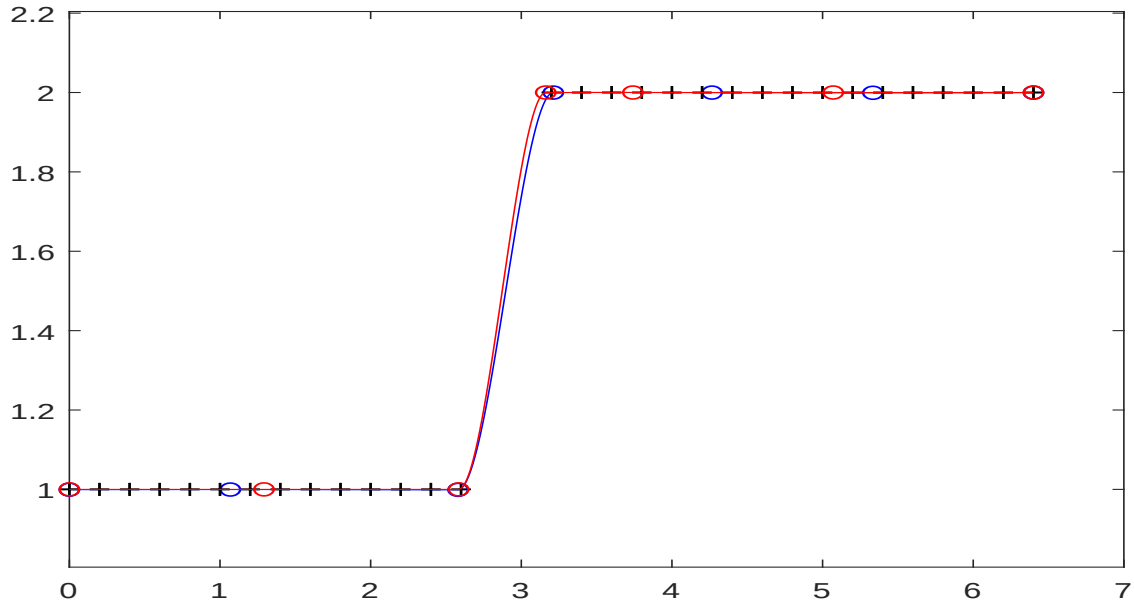

Figure 4.12: Cubic L^1 spline fits for dataset 5 obtained by the knot location optimization (starts from the second initial knot location setting)(in blue) vs the knot number and location determination (in red).

The approximation error of the spline fit decreases from 0.02 to 0.00 after the knot number is optimized. The spline fit almost perfectly preserves the shapes of the data with the number of the knots optimized.

The knot number and location determination method does not require the users to predetermine the initial knot location. The experimental results in this subsection show that the knot number and location determination method can start with a smaller number of evenly spaced knots and work effectively to construct the cubic L^1 spline fits that produce smaller approximation error and better shape-preservation.

4.3 Change Point Detection

This subsection presents two experiments using one artificial dataset and one real dataset. We use the change point detection method to find change point(s) in the two datasets.

We apply the heuristic method from subsection 3.2 to build cubic L^1 spline fits to approximate the datasets. We use six and five evenly spaced knot location as the initial knot location in experiments 11 and 12, respectively. We use the initial function value at knots from the linear spline [33]. All the parameters used in heuristic method of this subsection are set the same as experiments in subsection 4.1.

Experiment 11

We use an artificial dataset, dataset 6, for this experiment. The data are in five straight line segments with noise that follows a normal distribution $N(0, 0.01)$. The data are plotted as "+" in Figure 4.13.

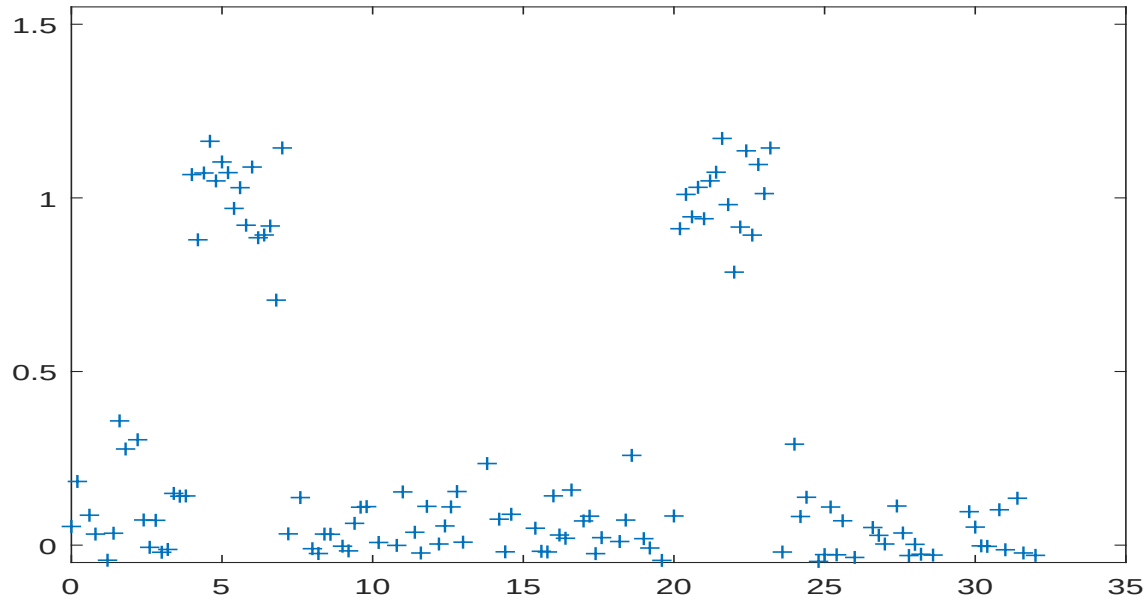

Figure 4.13: Dataset 6.

Dataset 6 can be found in the Appendix.

A cubic L^1 spline fit that approximates dataset 6 is constructed by using the heuristic method:

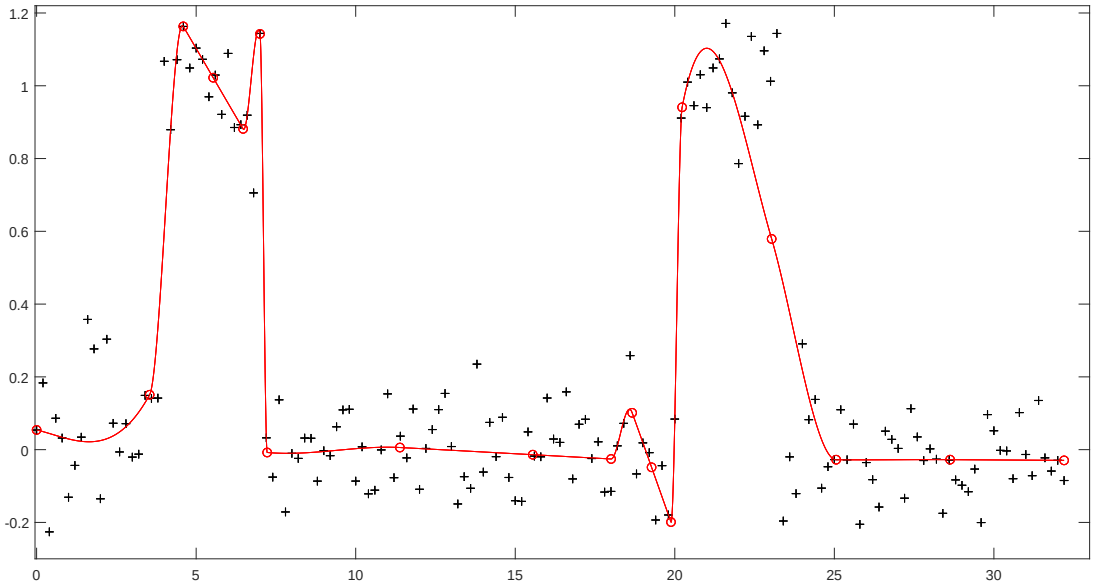

Figure 4.14: Cubic L^1 spline fit for dataset 6 obtained by the knot number and location determination.

In Figure 4.14, the given data points are plotted as "+". The cubic L^1 spline fit obtained by the knot number and location determination is plotted as solid curves in red. Knots are represented by circles.

The knots are:

$$\begin{aligned}
\boldsymbol{x}^{(*)} &= (0.00, 3.54, 4.60, 5.53, 6.47, 7.00, 7.23, 11.39, 15.55, 18.00, 18.66, 19.27, 19.88, 20.23, \\
&\quad 23.04, 25.06, 28.63, 32.20) \\
\boldsymbol{z}^{(*)} &= (0.05, 0.15, 1.16, 1.02, 0.88, 1.14, -0.01, 0.01, -0.01, -0.03, 0.10, -0.05, -0.20, 0.94, \\
&\quad 0.58, -0.03, -0.03, -0.03) \\
I^{(*)} &= 18
\end{aligned}$$

The absolute difference between the function value of every two consecutive knots is calculated:

$$\Delta = (0.10, 1.01, 0.14, 0.14, 0.26, 1.15, 0.01, 0.02, 0.01, 0.13, 0.15, 0.15, 1.14, 0.36, 0.61, 0.00, 0.00)$$

The four largest absolute differences are: $\Delta_6 = 1.15, \Delta_{13} = 1.14, \Delta_2 = 1.01, \Delta_{15} = 0.61$. Thus, the four most possible change points are found to be in the intervals $[x_6, x_7], [x_{13}, x_{14}], [x_2, x_3]$, $[x_{15}, x_{16}]$ that are $[7.00, 7.23], [19.88, 20.23], [3.54, 4.60], [23.04, 25.06]$.

Experiment 11 shows that change point detection method can detect change points even though there is noise in the data.

Experiment 12

We use a real data set, dataset 7, for this experiment. Dataset 7 is taken from Baltimore dataset [36] that depicts part of Baltimore City in Maryland State. Figure 4.15 shows the part of Baltimore dataset where horizontal indices and vertical indices are 470, 471,..., 539. The buildings with different heights are shown in the figure. Dataset 7 is sample data taken from Baltimore when horizontal indices are 470, 471, ..., 539, and vertical index is 490. Dataset 7 is highlighted in red in Figure 4.15.

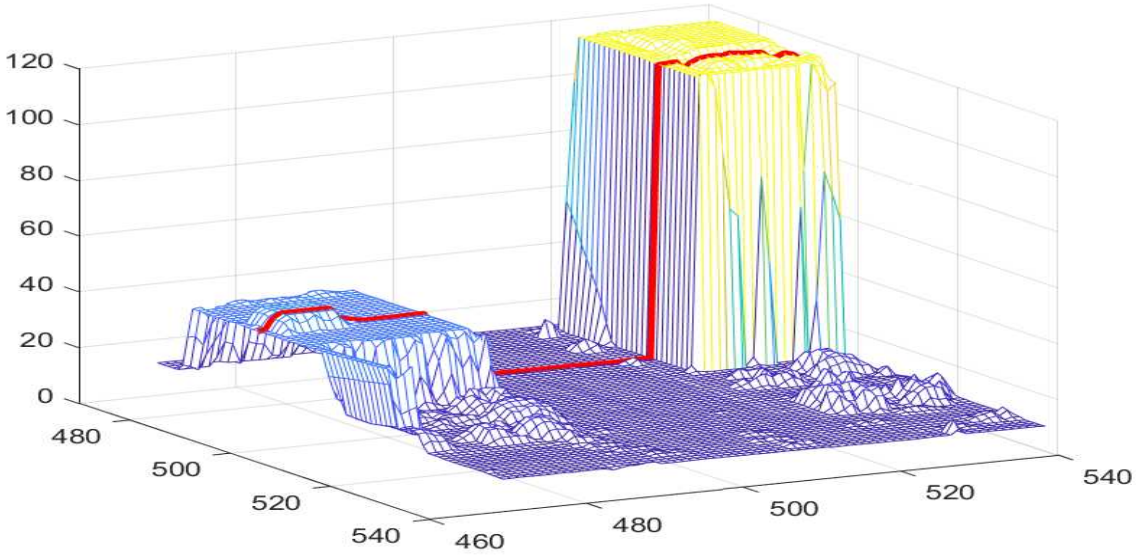

Figure 4.15: Sample data (Baltimore).

Dataset 7 can be found in the Appendix.

A cubic L^1 spline fit that approximates dataset 7 is constructed by using the heuristic method:

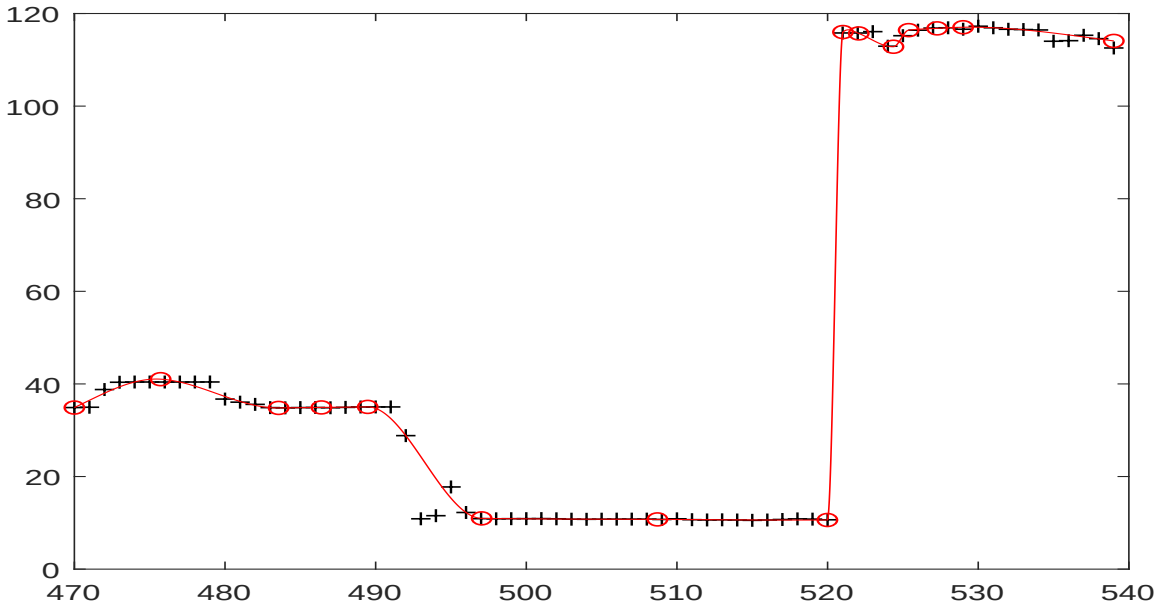

Figure 4.16: Cubic L^1 spline fit for dataset 7 obtained by the knot number and location determination.

In Figure 4.16, the given data points are plotted as "+". The cubic L^1 spline fit obtained by the knot number and location determination is plotted as solid curves in red. Knots are represented by circles.

The knots are:

$$\begin{aligned}
\boldsymbol{x}^{(*)} &= (470.00, 475.73, 483.56, 486.40, 489.47, 497.04, 508.71, 519.99, 521.02, 522.07, 524.37, \\
&\quad 525.38, 527.27, 529.00, 539.00,) \\
\boldsymbol{z}^{(*)} &= (34.90, 41.03, 34.82, 34.93, 35.04, 10.97, 10.74, 10.61, 116.00, 115.70, 112.82, 116.41, \\
&\quad 116.79, 117.03, 114.08,) \\
I^{(*)} &= 15
\end{aligned}$$

The absolute difference between the function value of every two consecutive knots is calculated:

$$\Delta = (6.13, 6.21, 0.10, 0.11, 24.07, 0.22, 0.13, 105.38, 0.30, 2.88, 3.59, 0.38, 0.24, 2.96)$$

The four largest absolute differences are: $\Delta_8 = 105.38$.

The most possible change point is found to be in the interval $[x_8, x_9]$ that is $[519.99, 521.02]$.

The experiments shown in this subsection show that we can apply the cubic L^1 spline fits with knot optimization for detecting the changes in magnitude.

CHAPTER 5

CONCLUSION AND FUTURE WORK

This research explores the methods to determine knots for univariate cubic L^1 spline fits. When the number of knots is given, we formulate the calculation of cubic L^1 spline fits as a bi-level nonsmooth nonlinear programming problem. The augmented Lagrangian method is applied to solve the constrained one-level nonlinear problem converted from the bi-level problem. When the number and the location are unknown, we propose a heuristic method to determine proper knot number and location. Numerical experiments on datasets with different shapes show that cubic L^1 spline fits with knot optimization can achieve higher accuracy in data approximation with smaller fitting error and better shape-preservation. The heuristic method can starts from a small number of evenly spaced knots to obtain the spline fits without requiring the users to determine the initial knot location. The cubic L^1 spline fits with optimized knots also show success in detecting the abrupt changes in magnitude. This thesis lays the foundation for research of knot optimization for other types splines such as bivariate and multivariate L^1 splines. The success of applying spline fits in magnitude change detection will inspire more applications of L^1 splines in solving real-world problem.

Due to the nonconvexity of the objective function in the minimization problem, the final solution is a local optimum rather than a global optimum. In experiments 4 and 5 when we use the knot location optimization method, we have witnessed that the choice of the initial knot location affects the final solution of knot location. In the future work, it can be interesting to investigate the relation between the initial and the final knots. A good location initialization method may be desired in application. It can also be promising to optimize the knots for other types of splines to achieve better performance in data approximation. The

application of cubic L^1 spline fits in CPD has not yet been fully investigated. Future work can focus on discovering how good the application is. Especially, the cubic L^1 spline fits applied in CPD can be compared with other CPD methods. Moreover, the cubic L^1 spline fits with knot optimization may also be used for other types of applications.

REFERENCES

[1] A. Alfiansyah. A unified energy approach for B-spline snake in medical image segmentation. *Telecommunication Computing Electronics and Control*, 8(2):175–186, 2010.

[2] S. Aminikhanghahi and D. Cook. A survey of methods for time series change point detection. *Knowledge and Information Systems*, 51(2):339–367, 2017.

[3] G. Beliakov. Least squares splines with free knots: global optimization approach. *Applied Mathematics and Computation*, 149(3):783–798, 2004.

[4] S. Boyd, L. Xiao, and A. Mutapcic. Subgradient methods. *Lecture Notes of EE392o, Stanford University, Autumn Quarter*, 2004:2004–2005, 2003.

[5] B. Chen. *Optimization Theory and Algorithms*. Beijing: Tsinghua University Publishing House, 2005.

[6] N. Chiu, S. Fang, J. E. Lavery, J. Lin, and Y. Wang. Approximating term structure of interest rates using cubic L^1 splines. *European Journal of Operational Research*, 184(3):990–1004, 2008.

[7] I. Cleland, M. Han, C. Nugent, H. Lee, S. McClean, S. Zhang, and S. Lee. Evaluation of prompted annotation of activity data recorded from a smart phone. *Sensors*, 14(9):15861–15879, 2014.

[8] C. De Boor. *A Practical Guide to Splines, revised Edition*. New York: Springer-Verlag, 2001.

[9] J. F. Ducré-Robitaille, L. A. Vincent, and G. Boulet. Comparison of techniques for detection of discontinuities in temperature series. *International Journal of Climatology: A Journal of the Royal Meteorological Society*, 23(9):1087–1101, 2003.

[10] T. A. Grandine. The extensive use of splines at boeing. *SIAM News*, 38(4):3–6, 2005.

[11] S. W. Han, H. Zhong, and M. Putt. An efficient operator for the change point estimation in partial spline model. *Communications in Statistics-Simulation and Computation*, 44(5):1171–1186, 2015.

[12] A. Hernoux-Villière, U. Lassi, T. Hu, A. Paquet, L. Rinaldi, G. Cravotto, S. Molina-Boisseau, M. Marais, and J. Lévêque. Simultaneous microwave/ultrasound-assisted hydrolysis of starch-based industrial waste into reducing sugars. *Sustainable Chemistry & Engineering*, 1(8):995–1002, 2013.

[13] Q. Jin, J. E. Lavery, and S. Fang. Univariate cubic l^1 interpolating splines: Analytical results for linearity, convexity and oscillation on 5-pointwindows. *Algorithms*, 3(3):276–293, 2010.

[14] H. Kang, F. Chen, Y. Li, J. Deng, and Z. Yang. Knot calculation for spline fitting via sparse optimization. *Computer Aided Design*, 58:179–188, 2015.

[15] Y. Kawahara and M. Sugiyama. Sequential change-point detection based on direct density-ratio estimation. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 5(2):114–127, 2012.

[16] Y. Kawahara, T. Yairi, and K. Machida. Change-point detection in time-series data based on subspace identification. *IEEE International Conference on Data Mining* 2012.

[17] M. J. Lai, C. K. Shum, V. Baramidze, and P. Wenston. Triangulated spherical splines for geopotential reconstruction. *Journal of Geodesy*, 83(8):695–708, 2009.

[18] J. E. Lavery. Univariate cubic L^p splines and shape-preserving, multiscale interpolation by univariate cubic L^1 splines. *Computer Aided Geometric Design*, 17(4):319–336, 2000.

[19] J. E. Lavery. Shape-preserving approximation of multiscale univariate data by cubic L^1 spline fits. *Computer Aided Geometric Design*, 21(1):43–64, 2004.

[20] W. Li, S. Xu, G. Zhao, and L. Goh. Adaptive knot placement in B-spline curve approximation. *Computer Aided Design*, 37(8):791–797, 2005.

[21] X. Liao and M. C. Meyer. Change-point estimation using shape-restricted regression splines. *Journal of Statistical Planning and Inference*, 188:8–21, 2017.

[22] D. G. Luenberger and Y. Ye. *Linear and Nonlinear Programming*. Vol. 2. MA: Addison-wesley, 1984.

[23] T. Lyche and K. Mørken. A data-reduction strategy for splines with applications to the approximation of functions and data. *IMA Journal of Numerical analysis*, 8(2):185–208, 1988.

[24] P. Ma, C. I. Castillo-Davis, W. Zhong and J. S. Liu. A data-driven clustering method for time course gene expression data. *Nucleic Acids Research*, 34(4):1261–1269, 2006.

[25] R. Malladi, G. P. Kalamangalam, and B. Aazhang. Online Bayesian change point detection algorithms for segmentation of epileptic activity. *Signals, Systems and Computers, 2013 Asilomar Conference on*, 1833–1837, 2013.

[26] T. Nie, Z. Wang, S. Fang and J. E. Lavery. Convex shape preservation of cubic L^1 spline fits. *Annals of Data Science*, 4(1):123–147, 2017.

[27] D. Rybach, C. Gollan, R. Schluter, and H. Ney. Audio segmentation for speech recognition using segment features. *Acoustics, Speech and Signal Processing*, 4197–4200, 2009.

[28] T. Tjahjowidodo. A direct method to solve optimal knots of B-spline curves: An application for non-uniform B-spline curves fitting. *PloS ONE*, 12(3):e0173857, 2017.

[29] T. Tjahjowidodo, V. Dung, and M. Han. A fast non-uniform knots placement method for B-spline fitting. *2015 IEEE International Conference on Advanced Intelligent Mechatronics*, 1490–1495, 2017.

[30] E. Ülker and A. Arslan. Automatic knot adjustment using an artificial immune system for B-spline curve approximation. *Information Sciences*, 179(10):1483–1494, 2009.

[31] Y. Wang. *Theory and Algorithms for Shape-preserving Bivariate Cubic L^1 Splines*. North Carolina State University, 2005.

[32] Z. Wang, S. Fang, and J. E. Lavery,. On shape-preserving capability of cubic L^1 spline fits. *Computer Aided Geometric Design*, 40:59–75, 2015.

[33] Z. Wang, J. Lavery, and S. Fang. Approximation of irregular geometric data by locally calculated univariate cubic L^1 spline fits. *Annals of Data Science*, 1(1):5–14, 2014.

[34] L. Yu, Q. Jin, J. E. Lavery, and S. Fang. Univariate cubic L^1 interpolating splines: Spline functional, window size and analysis-based algorithm. *Algorithms*, 3(3):311–328, 2010.

[35] M. A. U. Zaman. *Bicubic L^1 Spline Fits for 3D Data Approximation*. Northern Illinois University, 2018.

[36] W. Zhang. *Bivariate cubic L^1 Splines and applications*. North Carolina State University, 2007.

APPENDIX

DATASETS

Dataset 1

$$\hat{x}_m = (0, 0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8, 2, 2.2, 2.4, 2.6, 2.8, 3, 3.2,$$
$$3.4, 3.6, 3.8, 4, 4.2, 4.4, 4.6, 4.8, 5, 5.04, 5.08, 5.12, 5.16, 5.2, 5.24, 5.28,$$
$$5.32, 5.36, 5.4, 5.44, 5.48, 5.52, 5.56, 5.6, 5.64, 5.68, 5.72, 5.76, 5.8, 5.84, 5.88, 5.92,$$
$$5.96, 6, 6.04, 6.08, 6.12, 6.16, 6.2, 6.24, 6.28, 6.32, 6.36, 6.4, 6.44, 6.48, 6.52, 6.56,$$
$$6.6, 6.64, 6.68, 6.72, 6.76, 6.8, 6.84, 6.88, 6.92, 6.96, 7, 7.04, 7.08, 7.12, 7.16, 7.2)$$

$$\hat{z}_m = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,$$
$$0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,$$
$$1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,$$
$$0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,$$
$$0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$$

Dataset 2

$$\hat{x}_m = (0, 0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8, 2, 2.2, 2.4, 2.6, 2.8,$$
$$3, 3.05, 3.1, 3.15, 3.2, 3.25, 3.3, 3.35, 3.4, 3.45, 3.5, 3.55, 3.6, 3.65, 3.7,$$
$$3.75, 3.8, 3.85, 3.9, 3.95, 4, 4.05, 4.1, 4.15, 4.2, 4.25, 4.3, 4.35, 4.4, 4.45,$$
$$4.5, 4.55, 4.6, 4.65, 4.7, 4.75, 4.8, 4.85, 4.9, 4.95, 5, 5.05, 5.1, 5.15, 5.2,$$
$$5.25, 5.3, 5.35, 5.4, 5.45, 5.5, 5.55, 5.6, 5.65, 5.7, 5.75, 5.8, 5.85, 5.9, 5.95,$$
$$6, 6.5, 6.55, 6.6, 6.65, 6.7, 6.75, 6.8, 6.85, 6.9, 6.95, 7, 7.1, 7.2, 7.3,$$
$$7.4, 7.5, 7.6, 7.7, 7.8, 7.9, 8, 8.2, 8.4, 8.6, 8.8, 9, 9.2, 9.4, 9.6,$$
$$9.8, 10, 10.2, 10.4, 10.6, 10.8, 11, 11.2, 11.4, 11.6, 11.8, 12)$$

$$\hat{z}_m = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,$$
$$2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,$$
$$2, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,$$
$$0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.05, 0.1, 0.15, 0.2,$$
$$0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95,$$
$$1, 1, 1.025, 1.05, 1.075, 1.1, 1.125, 1.15, 1.175, 1.2, 1.225, 1.25, 1.275, 1.3, 1.325,$$
$$1.35, 1.375, 1.4, 1.425, 1.451.475, 1, 1, 1, 1, 1, 1, 1, 2, 1,$$
$$1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)$$

Dataset 3

$$\hat{x}_m = (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,$$
$$11, 12, 13, 14, 15, 16, 17, 18, 19, 20,$$
$$21, 22, 23, 24, 25, 26, 27, 28, 29, 30,$$
$$31, 32, 33, 34, 35, 36, 37, 38, 39, 40,$$
$$41, 42, 43, 44, 45, 46, 47, 48, 49, 50,$$
$$51, 52, 53, 54, 55, 56, 57, 58, 59, 60,$$
$$61, 62, 63, 64, 65, 66, 67, 68, 69, 70,$$
$$71, 72, 73, 74, 75, 76, 77, 78, 79, 80)$$
$$\hat{z}_m = (280, 280, 280, 280, 280, 280, 280, 281, 281, 281, 281,$$
$$280, 280, 280, 279, 279, 278, 278, 277, 277, 276,$$
$$276, 276, 276, 275, 276, 277, 278, 278, 279, 280,$$
$$300, 300, 300, 300, 300, 300, 301, 302, 302, 303,$$
$$303, 303, 303, 303, 303, 303, 305, 305, 305, 306,$$
$$306, 306, 308, 308, 308, 307, 306, 305, 305, 304,$$
$$303, 302, 302, 301, 300, 300, 300, 300, 300, 300,$$
$$300, 300, 300, 300, 300, 300, 300, 300, 300, 300)$$

Dataset 6

$$\hat{x}_m = (0, 0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6,$$
$$1.8, 2, 2.2, 2.4, 2.6, 2.8, 3, 3.2, 3.4,$$
$$3.6, 3.8, 4, 4.2, 4.4, 4.6, 4.8, 5, 5.2,$$
$$5.4, 5.6, 5.8, 6, 6.2, 6.4, 6.6, 6.8, 7,$$
$$7.2, 7.4, 7.6, 7.8, 8, 8.2, 8.4, 8.6, 8.8,$$
$$9, 9.2, 9.4, 9.6, 9.8, 10, 10.2, 10.4, 10.6,$$
$$10.8, 11, 11.2, 11.4, 11.6, 11.8, 12, 12.2, 12.4$$
$$12.6, 12.8, 13, 13.2, 13.4, 13.6, 13.8, 14, 14.2,$$
$$14.4, 14.6, 14.8, 15, 15.2, 15.4, 15.6, 15.8, 16,$$
$$16.2, 16.4, 16.6, 16.8, 17, 17.2, 17.4, 17.6, 17.8,$$
$$18, 18.2, 18.4, 18.6, 18.8, 19, 19.2, 19.4, 19.6,$$
$$19.8, 20, 20.2, 20.4, 20.6, 20.8, 21, 21.2, 21.4,$$
$$21.6, 21.8, 22, 22.2, 22.4, 22.6, 22.8, 23, 23.2$$
$$23.4, 23.6, 23.8, 24, 24.2, 24.4, 24.6, 24.8, 25,$$
$$25.2, 25.4, 25.6, 25.8, 26, 26.2, 26.4, 26.6, 26.8,$$
$$27, 27.2, 27.4, 27.6, 27.8, 28, 28.2, 28.4, 28.6,$$
$$28.8, 29, 29.2, 29.4, 29.6, 29.8, 30, 30.2, 30.4,$$
$$30.6, 30.8, 31, 31.2, 31.4, 31.6, 31.8, 32, 32.2)$$

$$\hat{z}_m = (0.0538, 0.1834, -0.2259, 0.0862, 0.0319, -0.1308, -0.0434, 0.0343, 0.3578,$$
$$0.2769, -0.1350, 0.3035, 0.0725, -0.0063, 0.0715, -0.0205, -0.0124, 0.1490,$$
$$0.1409, 0.1417, 1.0671, 0.8793, 1.0717, 1.1630, 1.0489, 1.1035, 1.0727,$$
$$0.9697, 1.0294, 0.9213, 1.0888, 0.8853, 0.8931, 0.9191, 0.7056, 1.1438,$$
$$0.0325, -0.0755, 0.1370, -0.1712, -0.0102, -0.0241, 0.0319, 0.0313, -0.0865,$$
$$-0.0030, -0.0165, 0.0628, 0.1093, 0.1109, -0.0864, 0.0077, -0.1214, -0.1114,$$
$$-0.0007, 0.1533, -0.0770, 0.0371, -0.0226, 0.1117, -0.1089, 0.0033, 0.0553,$$
$$0.1101, 0.1544, 0.0086, -0.1492, -0.0742, -0.1062, 0.2350, -0.0616, 0.0748,$$
$$-0.0192, 0.0889, -0.0765, -0.1402, -0.1422, 0.0488, -0.0177, -0.0196, 0.1419,$$
$$0.0292, 0.0198, 0.1588, -0.0804, 0.0697, 0.0835, -0.0244, 0.0216, -0.1166,$$
$$-0.1148, 0.0105, 0.0722, 0.2585, -0.0667, 0.0187, -0.0082, -0.1933, -0.0439,$$
$$-0.1795, 0.0840, 0.9112, 1.0100, 0.9455, 1.0304, 0.9400, 1.0490, 1.0739,$$
$$1.1712, 0.9806, 0.7862, 0.9160, 1.1355, 0.8928, 1.0961, 1.0124, 1.1437,$$
$$-0.1961, -0.0198, -0.1208, 0.2908, 0.0825, 0.1379, -0.1058, -0.0469, -0.0272,$$
$$0.1098, -0.0278, 0.0702, -0.2052, -0.0354, -0.0824, -0.1577, 0.0508, 0.0282,$$
$$0.0033, -0.1334, 0.1127, 0.0350, -0.0299, 0.0023, -0.0262, -0.1750, -0.0286,$$
$$-0.0831, -0.0979, -0.1156, -0.0534, -0.2003, 0.0964, 0.0520, -0.0020, -0.0035,$$
$$-0.0798, 0.1019, -0.0133, -0.0715, 0.1351, -0.0225, -0.0589, -0.0294, -0.0848)$$

Dataset 7

$$\hat{x}_m = (470, 471, 472, 473, 474, 475, 476,$$
$$477, 478, 479, 480, 481, 482, 483,$$
$$484, 485, 486, 487, 488, 489, 490,$$
$$491, 492, 493, 494, 495, 496, 497,$$
$$498, 499, 500, 501, 502, 503, 504,$$
$$505, 506, 507, 508, 509, 510.511,$$
$$512, 513, 514, 515, 516, 517, 518,$$
$$519, 520, 521, 522, 523, 524, 525,$$
$$526, 527, 528, 529, 530, 531, 532,$$
$$533, 534, 535, 536, 537, 538, 539)$$

$$\hat{z}_m = (34.8954, 34.9643, 38.7891, 40.3628, 40.4143, 40.4138, 40.4098,$$
$$40.4076, 40.4141, 40.4239, 36.7296, 36.0614, 35.5839, 34.8834,$$
$$34.8343, 34.8909, 34.9091, 34.8515, 34.9371, 35.0220, 35.0307,$$
$$35.0314, 28.8332, 10.8736, 11.5418, 17.7493, 12.2360, 10.8993,$$
$$10.8717, 10.8925, 10.9005, 10.9149, 10.8722, 10.7880, 10.7669,$$
$$10.7912, 10.8032, 10.8038, 10.8015, 10.7280, 10.8670, 10.6814,$$
$$10.6235, 10.6546, 10.6141, 10.5442, 10.6126, 10.7221, 10.8532,$$
$$10.8011, 10.6588, 115.7997, 115.7881, 116.0856, 112.9384, 115.2064,$$
$$116.3997, 116.8487, 116.9190, 116.6068, 117.2531, 116.9215, 116.5955,$$
$$116.5373, 116.4509, 114.0109, 114.1329, 115.2971, 114.5723, 112.5582)$$