

Article

Change Point Detection Using Penalized Multidegree Splines

Eun-Ji Lee [†]  and Jae-Hwan Jhong ^{*,†} 

Department of Information Statistics, Chungbuk National University, Cheongju 28644, Chungbuk, Korea; ejkrsu97@gmail.com

* Correspondence: jjh25@cbnu.ac.kr

† These authors contributed equally to this work.

Abstract: We consider a function estimation method with change point detection using truncated power spline basis and elastic-net-type L^1 -norm penalty. The L^1 -norm penalty controls the jump detection and smoothness depending on the value of the parameter. In terms of the proposed estimators, we introduce two computational algorithms for the Lagrangian dual problem (coordinate descent algorithm) and constrained convex optimization problem (an algorithm based on quadratic programming). Subsequently, we investigate the relationship between the two algorithms and compare them. Using both simulation and real data analysis, numerical studies are conducted to validate the performance of the proposed method.

Keywords: change point detection; coordinate descent algorithm; elastic net; spline; quadratic programming



Citation: Lee, E.-J.; Jhong, J.-H. Change Point Detection Using Penalized Multidegree Splines. *Axioms* **2021**, *10*, 331. <https://doi.org/10.3390/axioms10040331>

Academic Editor: Lucia Romani

Received: 13 October 2021

Accepted: 29 November 2021

Published: 1 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

A nonparametric function estimation is useful for estimating the actual relationship of data exhibiting nonlinear relationships. It is a statistical method for estimating the relationship between variables based on observed data, assuming that the actual function representing the relationships between input variables and response variables belongs to an infinite-dimensional parametric space. Representative examples, including kernel density estimation (chapter 1 of [1]), local polynomials [2], splines (pp. 118–180 of [3]) and curve estimation (chapter 2 of [4]), have been investigated previously.

A representative method used for function estimation is the basis function method. Among the basis function methods, splines are the most frequently used. In function estimation, an infinite number of parameters cannot be estimated using a finite amount of data. Therefore, function estimation is performed by introducing a function space and a basis function. If the function space and basis function are defined, then the shape of the function to be estimated using the basis function as a predictor variable can be expressed as a linear combination of the basis functions spanning the appropriate function space. Therefore, the function estimation problem can be regarded as estimating the regression coefficients.

Among the many basis functions used to interpolate data or fit smooth curves, the spline basis is the most typically used. The spline basis function is defined as a piecewise polynomial that is differentiable for every knot spacing. Its main underlying technology is truncated power splines (see chapter 3 of [5]). Additionally, it offers the advantages of simple construction and easy interpretation of the model parameters. However, in model with many overlapping intervals, basis functions are highly correlated. Therefore, several computational problems may occur when the number of knots increases significantly because the predictor matrix for the truncated power spline basis becomes dense. Consequently, incorrect fitting may occur (chapter 7 of [6]).

In the basis function methodology, the objective function to be optimized is a convex function. The coordinate descent algorithm (CDA) [7] is simple, efficient and useful for optimizing these objective functions. The concept of the algorithm is to optimize the solution

of a problem involving the minimization (maximization) of a convex (concave) function for a multidimensional vector. Coefficient updates keep the remaining coefficients constant. The objective function is regarded as a one-dimensional function. Tseng [8] proved that the coordinate descent converges when a convex penalty term is non-differentiable but separable in each coordinate. This result implies that the coordinate-wise algorithms for the least absolute shrinkage and selection operator (lasso) [9], group lasso [10] and elastic net [11] converge to their optimal solutions. Another simpler method to obtain the optimal solution is to use quadratic programming (QP) [12]. QP is a type of nonlinear programming that obtains a mathematical optimization problem involving quadratic functions.

The selection of knots and the detection of change points in regression splines significantly affect the performance of the model. Knot selection and change point detection methods have been studied extensively. Osbone et al. [13] proposed an algorithm that allows an efficient computation of the lasso estimator for knot selection to be performed. Leitenstorfer and Tutz [14] considered the boosting technique to select variables in knot selection. Garton [15] proposed a method for selecting the number and position of knots using Gaussian and non-Gaussian data. Aminikhanhahi and Cook [16] proposed a change point detection method for time-series data. Meanwhile, Tibshirani et al. [17] proposed sparsity and smoothness using a fused lasso.

The main contributions of this study are as follows.

First, a new type of function estimator is established for change points in the nonparametric regression function estimation model. The proposed function estimator is defined by a linear combination of multidegree-based splines to simultaneously provide a change point detection and smoothing.

Second, two computational algorithms are introduced to address the constrained convex optimization problem (algorithm based on QP) and Lagrangian dual problem (CDA); moreover, the relationship between them is investigated. Numerical studies using both simulated and real datasets are provided to demonstrate the performance of the proposed method.

Herein, we present a new statistical learning methodology for modeling and analyzing data to address the change point detection problem. The proposed method involves two tuning parameters that control the penalty terms. We express the estimator as a linear combination of a polynomial and a truncated power-spline basis for zero and positive integer degrees, respectively. The coefficients are estimated using a CDA to minimize the residual sum of squares and QP. We consider three sets of simulation data to measure the performance of the proposed method and conduct an analysis on real data with changes or jump point(s).

This paper is organized as follows: In Section 2, we present a nonparametric regression model containing observed data and the penalized regression spline estimator. The process of updating the coefficients based on the CDA and the estimated coefficients obtained using QP is presented in Section 3. Additionally, the estimators obtained using the two methods are compared and their characteristics are introduced briefly. In Section 4, using the simulation and real data, we validate the performance of our proposed model. In Section 5, we summarize the conclusions of this study. All codes implemented using the software program R for numerical analysis are available as Supplementary Materials.

2. Model and Estimator

Consider a nonparametric regression model

$$y_i = f(x_i) + \varepsilon_i \quad \text{for } i = 1, \dots, n, \quad (1)$$

where $\{x_i\}$'s $\in \mathcal{I} \subset \mathbb{R}$ are the predictors and $\{y_i\}$'s $\in \mathbb{R}$ are responses. The model is nonparametric because it does not contain any parametric assumptions regarding f . Model (1) is expressed similarly as the conditional expectation $E(Y|X = x)$. In this case, $\varepsilon_1, \dots, \varepsilon_n$ are only required to satisfy the property that the expected value is zero with

positive variance σ^2 and no distribution assumption is required. Without loss of generality, we write $\mathcal{I} = [0, 1]$ hereinafter for notational simplicity.

Because it is impossible to estimate an infinite number of parameters using a finite amount of data, a specific function space for estimation should be considered. In nonparametric estimation, it is typically assumed that f belongs to a massive class of functions [1]. In this study, because the goal is to achieve a smooth function estimation that can detect change point(s), the function space to which f belongs can be specified as follows.

Let $\mathcal{F}_K$ be the space of spline functions of mixed degrees 0 and positive integer $d > 0$ with K interior knots $0 = t_0 < t_1 < \dots < t_K < t_{K+1} = 1$. Therefore, any function $f \in \mathcal{F}_K$ can be expressed as

$$f(x; \theta) = \alpha_0 + \sum_{j=1}^d \alpha_j x^j + \sum_{k=1}^K \{ \beta_k P_{0,k}(x) + \gamma_k P_{d,k}(x) \},$$

where $\theta = (\alpha, \beta, \gamma)$ with $\alpha = (\alpha_0, \alpha_1, \dots, \alpha_d)$, $\beta = (\beta_1, \dots, \beta_K)$, $\gamma = (\gamma_1, \dots, \gamma_K)$,

$$P_{0,k}(x) = \begin{cases} 1 & \text{if } x > t_k \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad P_{d,k}(x) = (x - t_k)_+^d \quad \text{for } k = 1, \dots, K.$$

It is noteworthy that $(a)_+$ is a function that assumes a real value of a if $a > 0$ and 0 otherwise. We first select the number of the initial knots K for the splines and then place them at the same interval within $(0, 1)$. Generally, one may select a sufficiently large value of K — n , or $n/2$, for example—to select data-adaptive knots.

The empirical risk of θ is defined as

$$R(\theta) = \frac{1}{2n} \sum_{i=1}^n (y_i - f(x_i; \theta))^2.$$

The penalized objective function to be minimized is expressed as

$$R^{\lambda, \tau}(\theta) = R(\theta) + \lambda(\tau \|\beta\|_1 + (1 - \tau) \|\gamma\|_1) \quad (2)$$

for $\lambda > 0$, $\tau \in [0, 1]$ and $\|\cdot\|_1$, the L^1 -norm. The penalty term is the concept of elastic-net penalty [11]. Similar to the lasso, λ reduces only β and γ . Furthermore, similar to an elastic net, τ is a tuning parameter that controls the relative penalty. If τ is of a higher value, then a greater weight is assigned to the L^1 -norm of β , where the jump detection penalty is. By contrast, if τ is of a lower value, then a greater weight is assigned to the L^1 -norm of γ , which is a penalty indicating the function smoothness.

Let

$$\hat{\theta}^{\lambda, \tau} = \underset{\theta}{\operatorname{argmin}} R^{\lambda, \tau}(\theta)$$

and define the *multidegree spline estimator* (MDSE) as

$$\hat{f} = f(\cdot; \hat{\theta}).$$

3. Implementation

Specifically in mathematical statistics fields, penalized regression problems can seek to optimize a multivariate quadratic function subject to linear constraints on the regression coefficients. To realize the proposed estimator, two computational methods are considered, i.e., the CDA and the algorithm for QP, using the quadprog package [18] in the R program. We describe the abovementioned two methods comprehensively and present their advantages and disadvantages. Both methods are implemented using the software program R.

3.1. CDA

The CDA optimizes the objective function with respect to a single coefficient at a time, iteratively cycling through all coefficients until convergence is reached. We adopt the CDA introduced by Jhong et al. (2017) to perform univariate smoothing for each coefficient. Because the objective function (2) is convex for θ , one can adopt the CDA to obtain the MDSE. This algorithm is inspired by the studies of Friedman et al. [19] and Jhong et al. [20]. However, it is a new attempt, in that the proposed algorithm is applied to the elastic net penalty considering two hyper parameters, λ and τ .

For $j = 0, 1, 2, \dots, d$ and $k = 1, 2, \dots, K$, let $\tilde{\alpha}_j$, $\tilde{\beta}_k$ and $\tilde{\gamma}_k$ denote the initial (or updated) values. Specifically,

$$\tilde{\alpha} = (\tilde{\alpha}_0, \tilde{\alpha}_1, \dots, \tilde{\alpha}_d) \in \mathbb{R}^{1+d}, \quad \tilde{\beta} = (\tilde{\beta}_1, \dots, \tilde{\beta}_K) \in \mathbb{R}^K,$$

and $\tilde{\gamma} = (\tilde{\gamma}_1, \dots, \tilde{\gamma}_K) \in \mathbb{R}^K$. For notational convenience, we use

$$\tilde{\theta} = (\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}) = (\tilde{\theta}_1, \dots, \tilde{\theta}_J) \quad \text{for } J = 1 + d + 2K.$$

Therefore, the coordinate descent update is written in the following forms:

$$\tilde{\theta}_j \leftarrow \underset{\theta_j \in \mathbb{R}}{\operatorname{argmin}} R^{\lambda, \tau}(\tilde{\theta}_1, \dots, \tilde{\theta}_{j-1}, \theta_j, \tilde{\theta}_{j+1}, \dots, \tilde{\theta}_J) \quad \text{for } j = 1, \dots, J.$$

3.1.1. Updating α

The optimization problem for α_j , $j = 0, 1, \dots, d$ has a quadratic form. Hence, we can solve this problem by transforming the objective function for α_j into a perfect square form. It is noteworthy that $x_i^0 = 1$.

$$\begin{aligned} & R^{\lambda, \tau}(\tilde{\alpha}_0, \dots, \tilde{\alpha}_{j-1}, \alpha_j, \tilde{\alpha}_{j+1}, \dots, \tilde{\alpha}_d, \tilde{\beta}, \tilde{\gamma}) \\ &= \frac{1}{2n} \sum_{i=1}^n \left\{ y_i - \sum_{\ell \neq j} \tilde{\alpha}_\ell x_i^\ell - \alpha_j x_i^j - \sum_{k=1}^K (\tilde{\beta}_k P_{0,k}(x_i) + \tilde{\gamma}_k P_{d,k}(x_i)) \right\}^2 \\ & \quad + \lambda \tau \sum_{k=1}^K |\tilde{\beta}_k| + \lambda(1 - \tau) \sum_{k=1}^K |\tilde{\gamma}_k| \\ &= \frac{1}{2n} \sum_{i=1}^n \left(r_{i,\alpha}^{(j)} - \alpha_j x_i^j \right)^2 + (\text{terms independent for } \alpha_j) \\ &= \frac{\sum_{i=1}^n (x_i^j)^2}{2n} \left\{ \alpha_j - \frac{\sum_{i=1}^n r_{i,\alpha}^{(j)} x_i^j}{\sum_{i=1}^n (x_i^j)^2} \right\}^2 + (\text{terms independent for } \alpha_j), \end{aligned}$$

where

$$r_{i,\alpha}^{(j)} = y_i - \sum_{\ell \neq j} \tilde{\alpha}_\ell x_i^\ell - \sum_{k=1}^K \tilde{\beta}_k P_{0,k}(x_i) - \sum_{k=1}^K \tilde{\gamma}_k P_{d,k}(x_i)$$

is a partial residual. Because α_j is not subject to penalization, one can update each coefficient systematically using the partial residual. Therefore, we update

$$\tilde{\alpha}_j \leftarrow \frac{\sum_{i=1}^n r_{i,\alpha}^{(j)} x_i^j}{\sum_{i=1}^n (x_i^j)^2} \quad \text{for } j = 0, 1, \dots, d. \quad (3)$$

3.1.2. Updating β and γ

Unlike updating α , one cannot use only the partial residual to update β and γ , because the objective function contains the penalty terms. Hence, we used the soft-thresholding operator [19] to solve the lasso solutions. The soft-thresholding operator is defined as

$$S(y, \lambda) = \begin{cases} y - \lambda & \text{if } y > \lambda \\ y + \lambda & \text{if } y < -\lambda \\ 0 & \text{otherwise} \end{cases} \quad \text{for } y \in \mathbb{R} \quad \text{and } \lambda > 0.$$

We select $k \in \{1, \dots, K\}$ and transform the objective function for β_k into a quadratic form. Subsequently,

$$\begin{aligned} & R^{\lambda, \tau}(\tilde{\alpha}, \tilde{\beta}_1, \dots, \tilde{\beta}_{k-1}, \beta_k, \tilde{\beta}_{k+1}, \dots, \tilde{\beta}_K, \tilde{\gamma}) \\ &= \frac{1}{2n} \sum_{i=1}^n \left\{ y_i - \sum_{j=0}^d \tilde{\alpha}_j x_i^j - \sum_{\ell \neq k} \tilde{\beta}_\ell P_{0,\ell}(x_i) - \beta_k P_{0,k}(x_i) - \sum_{\ell=1}^K \tilde{\gamma}_\ell P_{d,\ell}(x_i) \right\}^2 \\ &+ \lambda \tau \sum_{\ell \neq k} |\tilde{\beta}_\ell| + \lambda \tau |\beta_k| + \lambda(1-\tau) \sum_{\ell=1}^K |\tilde{\gamma}_\ell| \\ &= \frac{1}{2n} \sum_{i=1}^n \left(r_{i,\beta}^{(k)} - \beta_k P_{0,k}(x_i) \right)^2 + \lambda \tau |\beta_k| + (\text{terms independent for } \beta_k), \end{aligned}$$

where

$$r_{i,\beta}^{(k)} = y_i - \sum_{j=0}^d \tilde{\alpha}_j x_i^j - \sum_{\ell \neq k} \tilde{\beta}_\ell P_{0,\ell}(x_i) - \sum_{\ell=1}^K \tilde{\gamma}_\ell P_{d,\ell}(x_i).$$

Finally, using the soft-thresholding operator, we can update $\tilde{\beta}_k$ as follows:

$$\tilde{\beta}_k \leftarrow \frac{S\left(\frac{1}{n} \sum_{i=1}^n r_{i,\beta}^{(k)} P_{0,k}(x_i), \lambda \tau\right)}{\frac{1}{n} \sum_{i=1}^n [P_{0,k}(x_i)]^2}. \quad (4)$$

Similarly, $\tilde{\gamma}_k$ is expressed as

$$\tilde{\gamma}_k \leftarrow \frac{S\left(\frac{1}{n} \sum_{i=1}^n r_{i,\gamma}^{(k)} P_{d,k}(x_i), \lambda(1-\tau)\right)}{\frac{1}{n} \sum_{i=1}^n [P_{d,k}(x_i)]^2} \quad \text{for } k = 1, \dots, K. \quad (5)$$

3.1.3. Algorithm Details

Algorithm 1 shows the process of computing the proposed estimator using the CDA. We implement the code using the software program R. For the observed data, we create a basis matrix B to fit to our model.

$$B = [\mathbf{1} \quad X^1 \quad \dots \quad X^d \quad P_0 \quad P_d] \in \mathbb{R}^{n \times (1+d+2K)},$$

where $\mathbf{1} = [1 \quad \dots \quad 1] \in \mathbb{R}^n$, $X^j = [x_1^j \quad \dots \quad x_n^j] \in \mathbb{R}^n$ for $j = 1, \dots, d$,

$P_0 = [P_{0,1} \quad \dots \quad P_{0,K}] \in \mathbb{R}^{n \times K}$, and $P_d = [P_{d,1} \quad \dots \quad P_{d,K}] \in \mathbb{R}^{n \times K}$. After B is created, we select the tuning parameters. We compute λ_{max} , which is the smallest value that causes all coefficients to be zero, as follows:

$$\lambda_{max} = \max_k \left(\frac{1}{n} \left| \langle P_{0,k}(x), y \rangle \right|, \frac{1}{n} \left| \langle P_{d,k}(x), y \rangle \right| \right),$$

where $\langle x, y \rangle = \sum_{i=1}^n x_i y_i$ for n -dimensional vectors x and y , respectively. Thereafter, we generate candidates for the two tuning parameters λ and τ , respectively.

For the case involving λ , the equally spaced candidates decrease from $\log(\lambda_{max})$ to $\log(\lambda_{max} \times \lambda_{min_max_ratio})$ for a maximum of n_λ times, which is the number of λ candidates. $\lambda_{min_max_ratio}$ is the ratio of the maximum and minimum values of λ , which is a sufficiently low positive value. Subsequently, it is reconverted to the exponential scale and the last candidate value is replaced with a zero. Next, in the case of τ , we generate n_τ values from 0 to 1 at equal intervals.

Subsequently, we compute the value of the initial objective function as $\sum_{i=1}^n y_i^2 / 2n$. Thereafter, we begin updating all the coefficients until that is achieved. Finally, we select the best model among $n_\lambda \times n_\tau$ models. In statistical data analysis, some representative information criteria exist, such as the Akaike information criterion (AIC; [21]), the Bayesian information criterion (BIC) and cross validation (CV). In general, the AIC and CV involve the selection of a model with a relatively large variance and small bias compared with the BIC. Because the purpose of this study is to accurately select significant change points, we use the BIC [22], which involves the selection of a model with a relatively small variance, defined as follows:

$$BIC = n \log(2 \times R^{\lambda, \tau}(\hat{\theta})) + p \log(n),$$

where p is the number of non-zero coefficients. We select the best model that minimized the BIC. For a detailed calculation of the degrees of freedom p in penalized regression model, see [9].

Algorithm 1: Coordinate descent algorithm (CDA).

Input: B : Basis matrix $\in \mathbb{R}^{n \times (1+d+2K)}$, y : response vector $\in \mathbb{R}^n$
 n_λ : the number of λ , n_τ : the number of τ
 d : non-zero degree, ϵ : stopping criterion
 K : number of interior knots, M : maximum iteration
Initialization: $\tilde{\alpha} \in \mathbb{R}^{1+d}$, $\tilde{\beta} \in \mathbb{R}^K$, $\tilde{\gamma} \in \mathbb{R}^K$
 $RSS_{old} = \sum_{i=1}^n y_i^2 / 2n$
for $\lambda_index = 1$ to n_λ **do**
 for $\tau_index = 1$ to n_τ **do**
 for $iteration = 1$ to M **do**
 for $j = 0$ to d **do**
 | Update $\tilde{\alpha}_j$ by (3)
 end
 for $k = 1$ to K **do**
 | Update $\tilde{\beta}_k$ by (4)
 end
 for $k = 1$ to K **do**
 | Update $\tilde{\gamma}_k$ by (5)
 end
 Compute $RSS_{new} = R^{\lambda, \tau}(\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma})$
 if $|RSS_{new} - RSS_{old}| \leq \epsilon$ **then**
 | **break**
 else
 | $RSS_{old} = RSS_{new}$
 end
 end
 end
end
Output: $\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}$

3.2. Quadratic Programming

The disadvantage of the CDA is that the computational cost increases with the higher number of iterations. Therefore, we use a new algorithm, i.e., QP, to compute the proposed estimator. In QP, certain mathematical optimization problems involving quadratic functions

are solved. In this regard, a multivariate quadratic function subject to linear constraints on the coefficients can be minimized. Hence, because the solution can be obtained without iterations, QP demonstrates computational advantage over the CDA.

The Lagrangian dual problem (2) is converted into the constraint optimization problem, such that

$$\hat{\theta} = (\hat{\alpha}, \hat{\beta}, \hat{\gamma}) = \underset{\theta}{\operatorname{argmin}} R(\theta) \quad \text{subject to} \quad \tau \|\beta\|_1 + (1 - \tau) \|\gamma\|_1 \leq t,$$

where $t \geq 0$. For $u \in \mathbb{R}$, let

$$u_+ = \begin{cases} u & \text{if } u \geq 0 \\ 0 & \text{if } u < 0 \end{cases} \quad \text{and} \quad u_- = \begin{cases} -u & \text{if } u \leq 0 \\ 0 & \text{if } u > 0 \end{cases}.$$

Therefore, $u = u_+ - u_-$ and $|u| = u_+ + u_-$. By reparameterizing β_k and γ_k ,

$$\beta_k = (\beta_k)_+ - (\beta_k)_-, \quad |\beta_k| = (\beta_k)_+ + (\beta_k)_-,$$

$$\gamma_k = (\gamma_k)_+ - (\gamma_k)_-, \quad |\gamma_k| = (\gamma_k)_+ + (\gamma_k)_-$$

Therefore, the optimization problem is equivalent to minimizing

$$\frac{1}{2n} \sum_{i=1}^n \left(y_i - \sum_{j=0}^d \alpha_j x_i^j - \sum_{k=1}^K [\{(\beta_k)_+ - (\beta_k)_-\} P_{0,k}(x_i) + \{(\gamma_k)_+ - (\gamma_k)_-\} P_{d,k}(x_i)] \right)^2$$

subject to

$$\tau \left\{ \sum_{k=1}^K ((\beta_k)_+ + (\beta_k)_-) \right\} + (1 - \tau) \left\{ \sum_{k=1}^K ((\gamma_k)_+ + (\gamma_k)_-) \right\} \leq t,$$

in addition to $4 \times K$ non-negativity constraints

$$(\beta_k)_+ \geq 0, \quad (\beta_k)_- \geq 0, \quad (\gamma_k)_+ \geq 0, \quad (\gamma_k)_- \geq 0 \quad \text{for } k = 1, \dots, K.$$

Next, we define

$$Z = [1 \quad X^1 \quad \dots \quad X^d \quad P_0 \quad -P_0 \quad P_d \quad -P_d] \in \mathbb{R}^{n \times (1+d+4K)}$$

and

$$b = [\alpha \quad \beta_+ \quad \beta_- \quad \gamma_+ \quad \gamma_-] \in \mathbb{R}^{1+d+4K},$$

where $\beta_+ = \{(\beta_k)_+\}$, $\beta_- = \{(\beta_k)_-\}$, $\gamma_+ = \{(\gamma_k)_+\}$, and $\gamma_- = \{(\gamma_k)_-\}$. Subsequently,

$$R(\theta) = \frac{1}{2n} (y - Zb)^\top (y - Zb) = \frac{1}{2n} y^\top y - \frac{1}{n} y^\top Zb + \frac{1}{2n} b^\top (Z^\top Z)b.$$

Hence, the minimization problem is equivalent to

$$\min_b \left(-\frac{1}{n} y^\top Zb + \frac{1}{2n} b^\top (Z^\top Z)b \right)$$

subject to

$$\begin{bmatrix} \mathbf{0}_{(1+d) \times 1} & \mathbf{0}_{(1+d) \times K} & \mathbf{0}_{(1+d) \times K} & \mathbf{0}_{(1+d) \times K} & \mathbf{0}_{(1+d) \times K} \\ -\tau_{K \times 1} & I_{K \times K} & \mathbf{0}_{K \times K} & \mathbf{0}_{K \times K} & \mathbf{0}_{K \times K} \\ -\tau_{K \times 1} & \mathbf{0}_{K \times K} & I_{K \times K} & \mathbf{0}_{K \times K} & \mathbf{0}_{K \times K} \\ -(\mathbf{1} - \tau)_{K \times 1} & \mathbf{0}_{K \times K} & \mathbf{0}_{K \times K} & I_{K \times K} & \mathbf{0}_{K \times K} \\ -(\mathbf{1} - \tau)_{K \times 1} & \mathbf{0}_{K \times K} & \mathbf{0}_{K \times K} & \mathbf{0}_{K \times K} & I_{K \times K} \end{bmatrix}^\top \begin{bmatrix} \alpha \\ \beta_+ \\ \beta_- \\ \gamma_+ \\ \gamma_- \end{bmatrix} \geq \begin{bmatrix} -t \\ \mathbf{0}_{K \times 1} \\ \mathbf{0}_{K \times 1} \\ \mathbf{0}_{K \times 1} \\ \mathbf{0}_{K \times 1} \end{bmatrix},$$

where $\tau_{K \times 1} = (\tau, \dots, \tau) \in \mathbb{R}^K$ and $(1 - \tau)_{K \times 1} = ((1 - \tau), \dots, (1 - \tau)) \in \mathbb{R}^K$. Hence, the problem can be computed via QP.

In R, the `solve.QP` function in the `quadprog` package [18] provides the solution for QP problems that involve minimizing $(-\text{dvec}^\top x + 1/2x^\top \text{Dmat } x)$ with respect to x with constraints $\text{Amat}^\top x \geq \text{bvec}$. `Dmat` is the matrix that appears in the quadratic function to be minimized. `dvec` is the vector that appears in the quadratic function to be minimized. `bvec` is the vector that has the values of the constraints. `Amat` is the matrix that defines the constraints for minimizing the quadratic function. Therefore, in our case, the matrices and vectors defined above correspond to

$$\text{Dmat} \leftarrow \frac{1}{n}(Z^\top Z), \quad \text{dvec} \leftarrow \frac{1}{n}y^\top Z, \quad \text{bvec} \leftarrow \begin{bmatrix} -t \\ \mathbf{0}_{4K \times 1} \end{bmatrix}$$

and

$$\text{Amat} \leftarrow \begin{bmatrix} \mathbf{0}_{(1+d) \times 1} & \mathbf{0}_{(1+d) \times K} & \mathbf{0}_{(1+d) \times K} & \mathbf{0}_{(1+d) \times K} & \mathbf{0}_{(1+d) \times K} \\ -\tau_{K \times 1} & I_{K \times K} & \mathbf{0}_{K \times K} & \mathbf{0}_{K \times K} & \mathbf{0}_{K \times K} \\ -\tau_{K \times 1} & \mathbf{0}_{K \times K} & I_{K \times K} & \mathbf{0}_{K \times K} & \mathbf{0}_{K \times K} \\ -(\mathbf{1} - \tau)_{K \times 1} & \mathbf{0}_{K \times K} & \mathbf{0}_{K \times K} & I_{K \times K} & \mathbf{0}_{K \times K} \\ -(\mathbf{1} - \tau)_{K \times 1} & \mathbf{0}_{K \times K} & \mathbf{0}_{K \times K} & \mathbf{0}_{K \times K} & I_{K \times K} \end{bmatrix}.$$

It is noteworthy that a small positive vector is added to the diagonal entries of $\frac{1}{n}(Z^\top Z)$ to guarantee a positive definite matrix and prevent numerical computational issues.

Finally, the solution of b obtained via QP is defined as

$$\hat{b} = [\hat{\alpha} \quad \hat{\beta}_+ \quad \hat{\beta}_- \quad \hat{\gamma}_+ \quad \hat{\gamma}_-],$$

and the MDSE is expressed as

$$\hat{\beta}_k = (\hat{\beta}_k)_+ - (\hat{\beta}_k)_-, \quad \text{and} \quad \hat{\gamma}_k = (\hat{\gamma}_k)_+ - (\hat{\gamma}_k)_- \quad \text{for } k = 1, \dots, K.$$

3.3. Comparison between CDA and QP

We introduce two computational algorithms to solve the constrained convex optimization problem (algorithm based on QP) and the Lagrangian dual problem (CDA); subsequently, we investigate the relationship between them.

First, we verify that the CDA estimator and the estimator obtained via QP are identical. Subsequently, a simulation is performed using the sample size $n = 300$ with one true knot of 0.5, as shown in Figure 1.

In this simulation, the settings used for the CDA yield zero for all the initial coefficients, with $n_\lambda = 50$ (the number of λ), $n_\tau = 10$ (the number of τ), $d = 1$ (the non-zero degree), $\epsilon = 1 \times 10^{-10}$ (stopping criterion) and $R = 5000$ (maximum iteration). Using λ and τ obtained from the best CDA model, the corresponding tuning parameter t required for QP is obtained as follows:

$$t = \lambda \tau \sum_{k=1}^K |\hat{\beta}_k| + \lambda (1 - \tau) \sum_{k=1}^K |\hat{\gamma}_k|.$$

The above relation can be derived using the Karush–Kuhn–Tucker (KKT) optimality condition [23] based on the sub-gradient [24], since the L^1 -norm penalty term cannot be differentiated with respect to the coefficients.

The remaining options are set to the same value and then are simulated. As shown in Figure 1, we confirm that both algorithms yield the same results. However, when the number of knots (K) increases, the CDA does not converge appropriately to the minimum under a limited iteration number (M) and the stopping criterion (ϵ), unlike QP. Hence, we perform the next simulation using only QP. It is noteworthy that a higher number K of interior knots indicates a higher probability of overfitting.

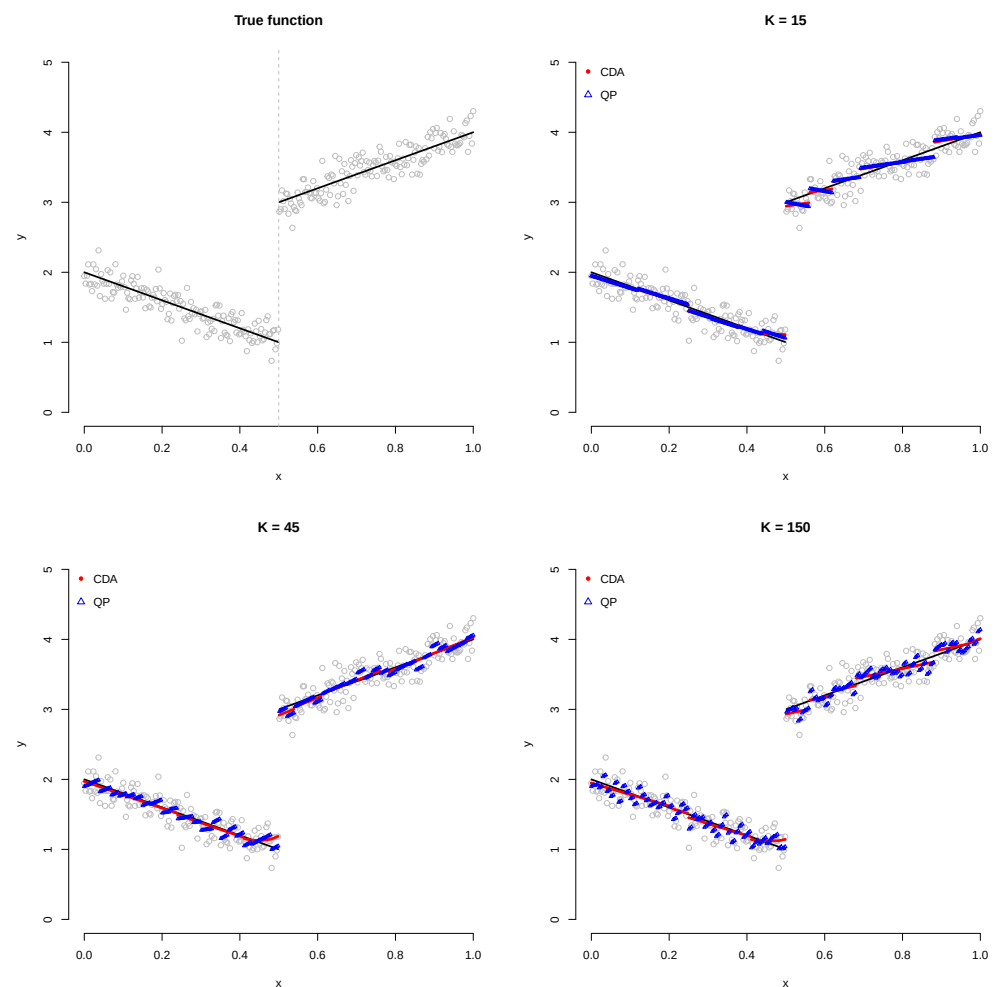


Figure 1. Plot showing comparison between CDA and QP. Red circles and blue triangles represent CDA and QP fitted values, respectively.

4. Numerical Analysis

4.1. Simulation

In this section, we analyze the performance of the proposed method based on simulated examples. From model (1), we generate the predictor as n sequences in the range $[0, 1]$. ε_i is generated from $N(0, \sigma^2)$ for $i = 1, \dots, n$ and σ is set 0.25 for all examples.

Next, we present three examples, as follows.

Example 1. Piecewise linear function with a single change point

$$f(x) = \begin{cases} 13x & \text{if } 0 \leq x < 0.333 \\ -10x + 3 & \text{if } 0.333 \leq x \leq 1 \end{cases}$$

Example 2. Piecewise linear function with two change points

$$f(x) = \begin{cases} 8x & \text{if } 0 \leq x < 0.333 \\ 8x - 2 & \text{if } 0.333 \leq x < 0.570 \\ 2.56 & \text{if } 0.570 \leq x < 0.666 \\ -10x + 11.27 & \text{if } 0.666 \leq x \leq 1 \end{cases}$$

Example 3. Piecewise cubic function with a single change point

$$f(x) = \begin{cases} 20x^3 + 50x^2 & \text{if } 0 \leq x < 0.333 \\ -12x^2 + 15 & \text{if } 0.333 \leq x \leq 1 \end{cases}$$

In Examples 1–2, a linear model with the jump(s) is considered. In Example 1, the true knot is 0.333, whereas they are 0.333 and 0.666 in Example 2. Example 3 is designed as a nonlinear function, where the true function is of quadratic and cubic form. Figure 2 shows the true functions of Examples 1–3.

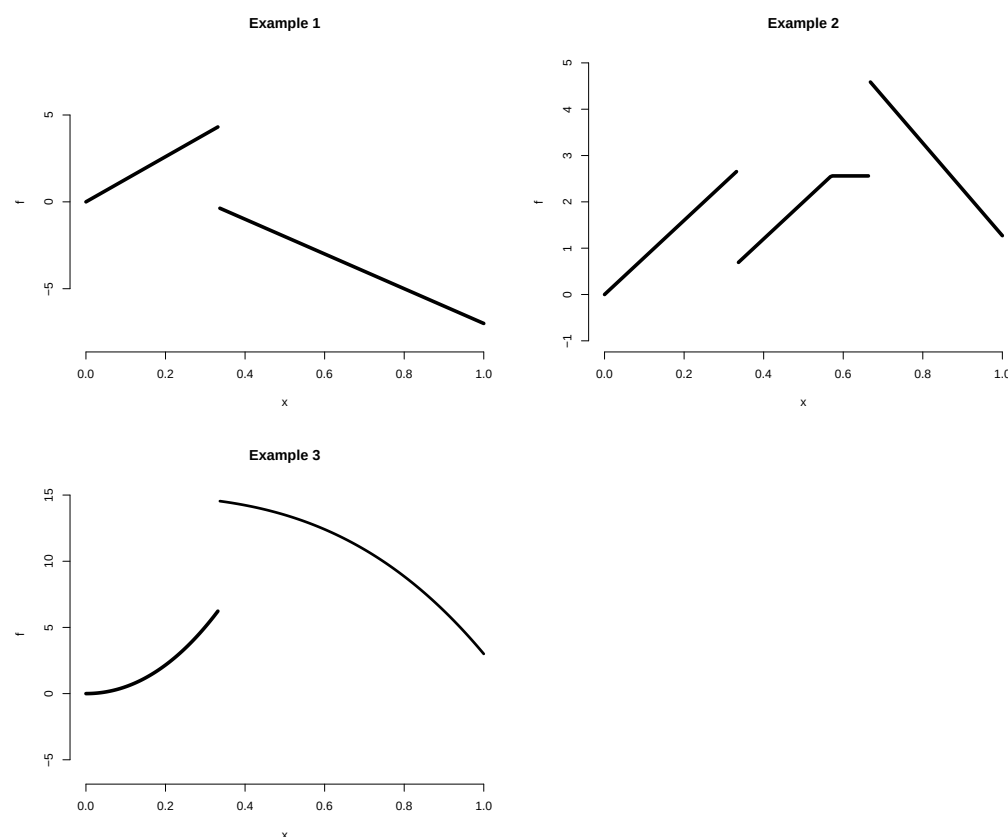


Figure 2. Plot showing true function of Examples 1–3. Top-left panel is for Example 1, which has a true knot of 0.333. Top-right panel is for Example 2, which has two true knots of 0.333 and 0.666. Bottom-left panel is for Example 3, which exhibits a nonlinear form with a true knot of 0.333.

We compare the proposed method with the fused lasso (FL) of Tibshirani et al. [17], the trend filtering (TF) of Kim et al. [25], the smoothing spline (SS) of Kim and Gu [26] and a fitting method for the structural model for a time series (ST) by Harvey [27].

The FL is a method that yields a solution path for the general fused lasso problem. The TF is the method of computing the solution path for the trend filtering problem of an arbitrary polynomial order. The SS is a method of fitting smoothing spline ANOVA models in Gaussian regression. The ST fits a structural model for a time series using the maximum likelihood. It is noteworthy that the FL, TF and SS estimators use nonparametric regression function estimation models. In particular, the TF and SS are function estimators based on linear combinations of spline basis functions, similar to ours. In addition, the FL is an estimator that specializes in change point detection as a zero-degree piecewise constant function.

The FL and TF estimators are provided in the `genlasso` package in R [28]. They require the use of cross-validation for parameter selection. In addition, the SS and ST can be used

in R via the `gss` [29] and `stats` packages [30], respectively. To reduce the computational burden, we compute all the estimators using the default settings for each package.

We consider the mean squared error (MSE), mean absolute error (MAE) and maximum deviation (MXDV) criterion as loss functions that measure the discrepancy between the true function f and each function estimator $\hat{f}$. They are expressed as follows:

$$\text{MSE}(\hat{f}) = \frac{1}{n} \sum_{i=1}^n (f(x_i) - \hat{f}(x_i))^2, \quad \text{MAE}(\hat{f}) = \frac{1}{n} \sum_{i=1}^n |f(x_i) - \hat{f}(x_i)|$$

$$\text{and } \text{MXDV}(\hat{f}) = \max_{1 \leq i \leq n} |f(x_i) - \hat{f}(x_i)|.$$

Examples 1–2 are simulated based on $d = 1$, whereas Example 3 is simulated based on $d = 3$, with $K = 20$ and $K = 50$, respectively. In addition, a simulation is performed by setting TF with orders 1 and 3 for linear and cubic functions, respectively. The simulation is repeated 100 times for each example based on sample sizes of 200, 300 and 500. Tables 1–3 show the experimental results of various scenarios for each of the examples.

As shown in Table 1, because QP not only can detect change points (or jumps), but can also fit piecewise polynomial regression, it demonstrates the best performance compared with the other methods. Comparing the QP results based on K , it is discovered that the best performance is obtained when $K = 20$. Figure 3 shows the plot for the results of Example 1 when $n = 500$. QP detects the change point and fits the linear regression accurately, particularly when $K = 20$. Although the performances of FL and TF with order 1 improve in general as the sample size n increases, it cannot detect the change point. Herein, “cannot detect the change point” means that, around the true knot, the plot has point(s) alone or successive.

Table 1. Average ($\times 100$) of each criterion over 100 trials of QP, FL, TF, SS and ST for Example 1 (for sample size $n = 200, 300$ and 500; standard error in parentheses). Bold text represents smallest criterion for each scenario.

n		QP (K = 20)	QP (K = 50)	FL	TF (ord = 1)	TF (ord = 3)	SS	ST
200	MSE	0.52 (3×10^{-3})	0.80 (6×10^{-3})	1.72 (3×10^{-3})	1.96 (7×10^{-3})	3.54 (8×10^{-3})	9.47 (3×10^{-2})	4.55 (9×10^{-3})
	MAE	4.27 (1×10^{-2})	5.35 (2×10^{-2})	10.41 (1×10^{-2})	8.45 (1×10^{-2})	11.77 (2×10^{-2})	13.16 (2×10^{-2})	16.18 (4×10^{-3})
	MXDV	37.70 (2×10^{-1})	49.67 (3×10^{-1})	39.71 (7×10^{-2})	102.70 (4×10^{-1})	133.93 (3×10^{-1})	217.82 (2×10^{-1})	117.69 (2×10^{-2})
300	MSE	0.22 (1×10^{-3})	0.40 (4×10^{-3})	0.94 (2×10^{-3})	1.11 (4×10^{-3})	2.11 (4×10^{-3})	7.85 (3×10^{-2})	2.91 (7×10^{-3})
	MAE	2.74 (8×10^{-3})	3.78 (1×10^{-2})	7.72 (7×10^{-3})	6.15 (1×10^{-2})	8.66 (1×10^{-2})	10.70 (2×10^{-2})	12.76 (3×10^{-3})
	MXDV	25.84 (1×10^{-1})	37.02 (2×10^{-1})	29.90 (6×10^{-2})	95.21 (3×10^{-2})	133.34 (3×10^{-2})	222.44 (1×10^{-2})	120.09 (2×10^{-2})
500	MSE	0.13 (9×10^{-4})	0.22 (2×10^{-3})	0.65 (1×10^{-3})	0.92 (2×10^{-3})	4.57 (7×10^{-3})	7.07 (2×10^{-2})	2.55 (5×10^{-3})
	MAE	2.01 (5×10^{-3})	2.70 (1×10^{-2})	6.44 (4×10^{-3})	5.43 (8×10^{-3})	6.91 (5×10^{-3})	9.43 (2×10^{-2})	11.62 (2×10^{-3})
	MXDV	19.84 (1×10^{-1})	28.95 (2×10^{-1})	26.03 (5×10^{-2})	113.64 (3×10^{-1})	218.35 (6×10^{-2})	232.12 (1×10^{-1})	159.22 (2×10^{-1})

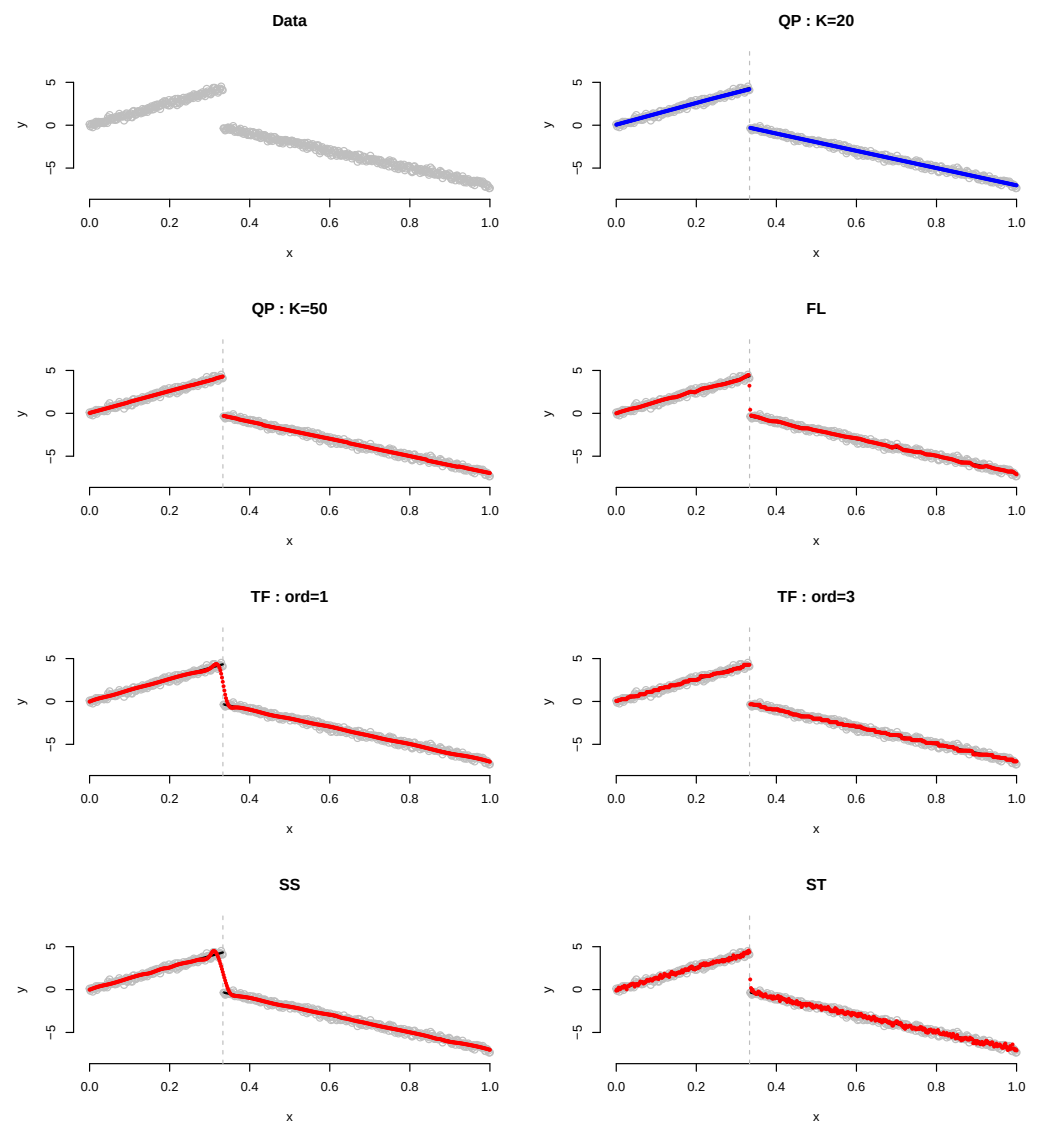


Figure 3. Plot for Example 1 when $n = 500$. Gray dots, data points; blue dots, best-fitted values; red dots, remaining fitted values.

Table 2 lists the results of Example 2, which is involved the same slope, two jumps and no jump at $x = 0.57$. It is observed that QP demonstrates the best performance when $K = 20$, not $K = 50$. Figure 4 shows the plot for Example 2 when $n = 500$. QP detects the change point and fits the linear regression accurately, particularly when $K = 20$. Although most performances of FL and TF with order 1 improve in general, as the sample size n increases, they cannot detect the change point. Herein, “cannot detect the change point” means that, around the true knot, the plot has point(s) alone or successive.

Table 2. Average ($\times 100$) of each criterion over 100 trials of QP, FL, TF, SS and ST for Example 2 (for sample sizes $n = 200, 300$ and 500; standard error in parentheses). The bold text represents the smallest criterion for each scenario.

n		QP (K = 20)	QP (K = 50)	FL	TF (ord = 1)	TF (ord = 3)	SS	ST
200	MSE	0.62 (3×10^{-3})	1.39 (1×10^{-2})	1.51 (3×10^{-3})	2.10 (4×10^{-3})	2.89 (8×10^{-3})	4.44 (9×10^{-3})	3.51 (8×10^{-3})
	MAE	5.65 (1×10^{-2})	8.32 (3×10^{-2})	9.65 (1×10^{-2})	9.20 (2×10^{-2})	11.46 (2×10^{-2})	11.93 (1×10^{-2})	13.96 (4×10^{-3})
	MXDV	28.50 (1×10^{-1})	39.88 (2×10^{-1})	38.92 (8×10^{-2})	83.52 (1×10^{-1})	86.71 (1×10^{-1})	100.90 (6×10^{-2})	93.29 (1×10^{-1})
300	MSE	0.43 (2×10^{-3})	0.58 (4×10^{-3})	1.13 (2×10^{-3})	1.64 (3×10^{-3})	2.19 (3×10^{-3})	3.82 (6×10^{-3})	3.05 (9×10^{-3})
	MAE	4.72 (1×10^{-2})	5.50 (1×10^{-2})	8.37 (8×10^{-3})	7.87 (1×10^{-2})	9.72 (1×10^{-2})	10.53 (1×10^{-2})	12.67 (4×10^{-3})
	MXDV	25.33 (9×10^{-2})	29.66 (1×10^{-1})	36.30 (1×10^{-1})	86.71 (1×10^{-1})	89.47 (1×10^{-1})	103.02 (5×10^{-2})	107.01 (1×10^{-2})
500	MSE	0.25 (1×10^{-3})	0.63 (5×10^{-3})	0.76 (1×10^{-3})	1.20 (1×10^{-3})	1.66 (1×10^{-3})	3.21 (6×10^{-3})	2.49 (6×10^{-3})
	MAE	3.52 (8×10^{-3})	5.21 (2×10^{-2})	6.87 (6×10^{-3})	6.54 (1×10^{-2})	7.76 (9×10^{-3})	9.09 (1×10^{-2})	11.23 (2×10^{-3})
	MXDV	20.33 (9×10^{-2})	32.89 (1×10^{-1})	32.42 (1×10^{-1})	91.12 (8×10^{-2})	96.79 (5×10^{-2})	105.51 (6×10^{-2})	122.67 (1×10^{-1})

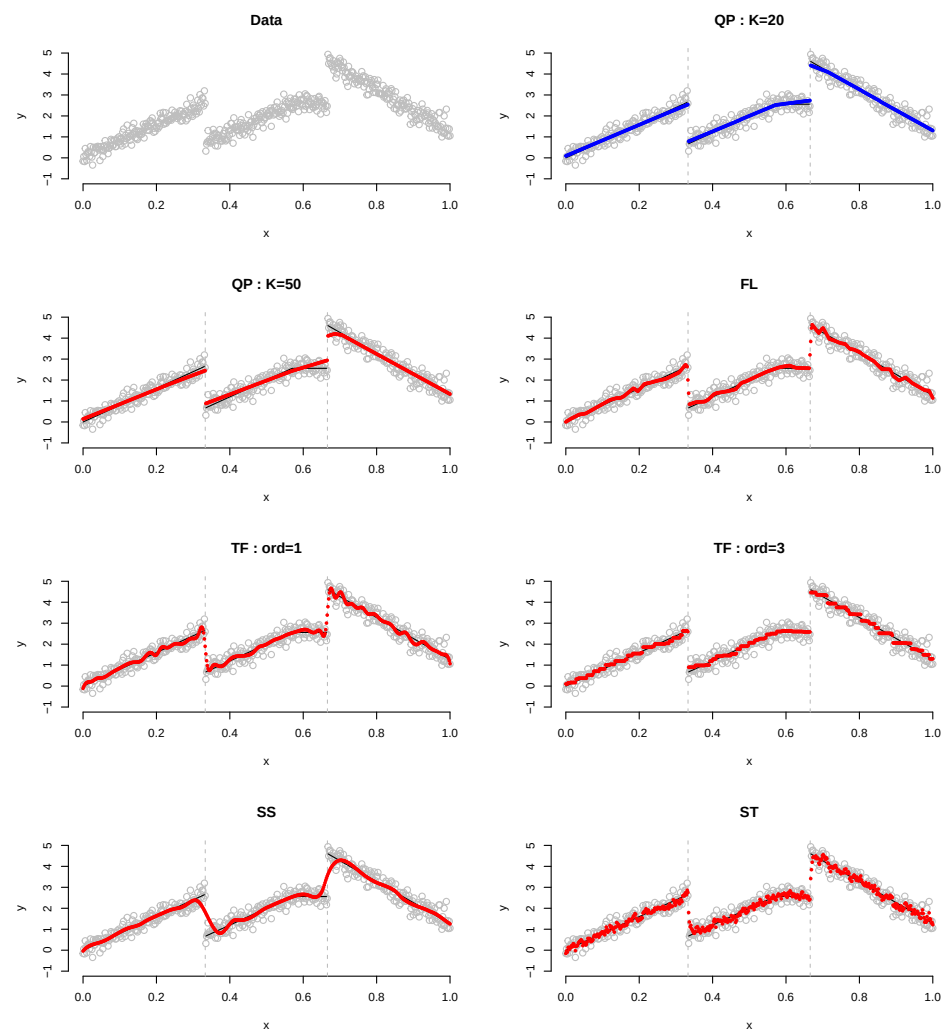


Figure 4. Plot for Example 2 when $n = 500$. Gray dots, data points; blue dots, best-fitted values; red dots, remaining fitted values.

Table 3 lists the results of Example 3, which is involved one jump as well as quadratic and cubic forms. Similarly, the proposed QP estimator yields the best results. $K = 20$ yields the best results in terms of the MSE and MAE, whereas $K = 50$ in terms of the MXDV. Figure 5 shows the results of Example 3 for all methods when $n = 500$. As shown, even when it do not fit the polynomial regression line well at $K = 50$, it detects the point of change accurately at the true knot.

QP detects the change point and fits the linear regression accurately, particularly when $K = 20$. Although the performances of the FL and TF with order 1 improve as the sample size n increases, they cannot detect the change point. Herein, “cannot detect the change point” means that, around the true knot, the plot has point(s) alone or successive.

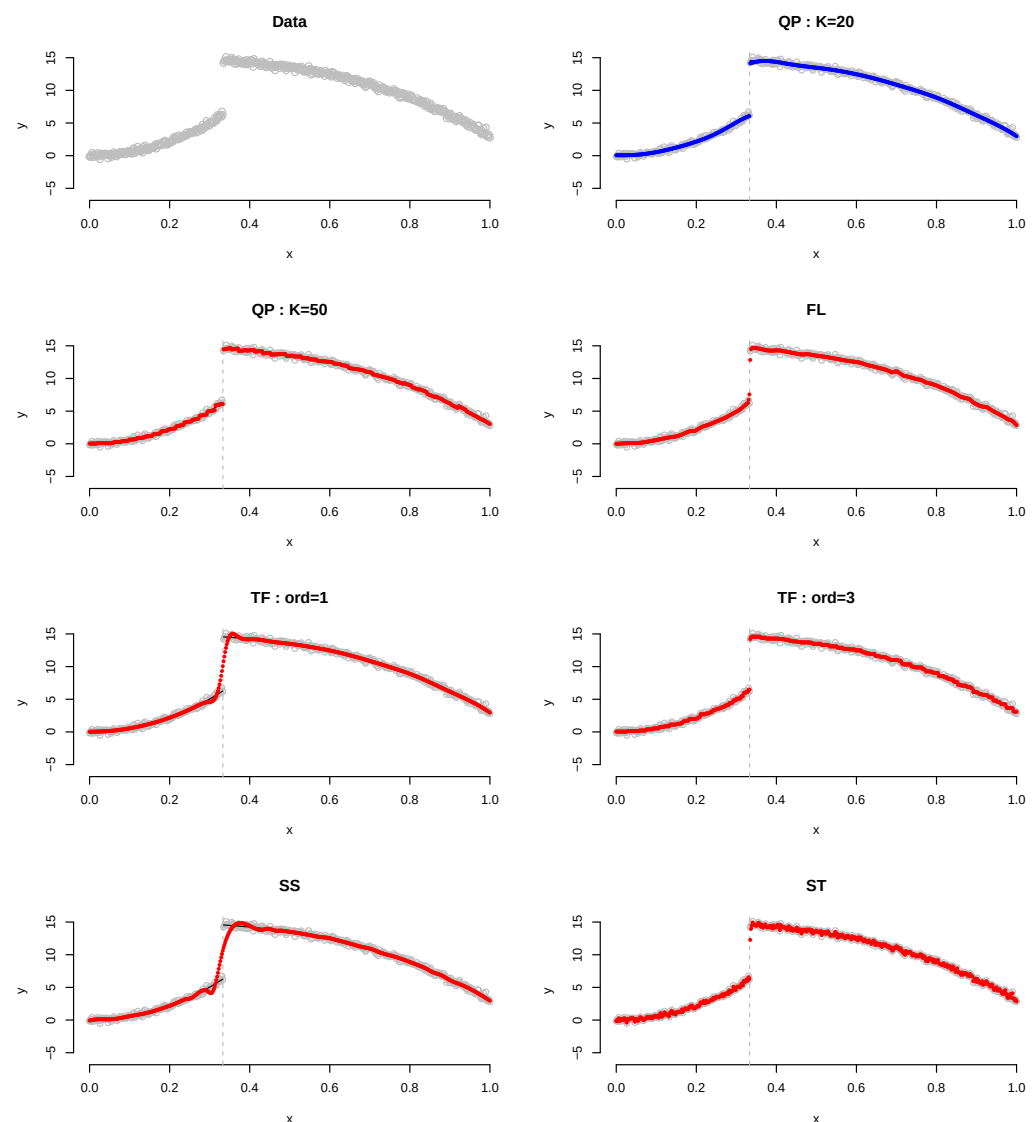


Figure 5. Plot for Example 3 when $n = 500$. Gray dots, data points; blue dots, best-fitted values; red dots, remaining fitted values.

Table 3. Average ($\times 100$) of each criterion over 100 trials of QP, FL, TF, SS and ST for Example 3 (for sample sizes $n = 200, 300$ and 500; standard error in parentheses). Bold text represents smallest criterion for each scenario.

n		QP (K = 20)	QP (K = 50)	FL	TF (ord = 1)	TF (ord = 3)	SS	ST
200	MSE	0.95	1.97	2.30	1.77	4.81	26.96	5.53
		(6×10^{-3})	(3×10^{-3})	(6×10^{-3})	(5×10^{-3})	(1×10^{-2})	(1×10^{-1})	(1×10^{-2})
	MAE	6.73	11.11	11.90	8.62	12.59	18.83	18.58
		(2×10^{-2})	(9×10^{-3})	(1×10^{-2})	(2×10^{-2})	(2×10^{-2})	(4×10^{-2})	(6×10^{-3})
	MXDV	50.06	41.95	46.85	91.63	165.54	382.14	81.19
		(1×10^{-1})	(6×10^{-2})	(9×10^{-2})	(3×10^{-1})	(6×10^{-1})	(3×10^{-1})	(2×10^{-1})
300	MSE	0.63	1.46	1.65	1.57	3.97	23.48	5.20
		(2×10^{-3})	(2×10^{-3})	(2×10^{-3})	(3×10^{-3})	(6×10^{-3})	(8×10^{-2})	(9×10^{-3})
	MAE	5.46	9.52	10.10	7.69	11.20	16.70	17.69
		(9×10^{-3})	(8×10^{-3})	(7×10^{-3})	(1×10^{-2})	(1×10^{-2})	(4×10^{-2})	(5×10^{-3})
	MXDV	47.32	38.72	42.29	113.08	191.36	395.16	119.95
		(1×10^{-1})	(6×10^{-2})	(8×10^{-2})	(3×10^{-1})	(4×10^{-1})	(2×10^{-1})	(3×10^{-1})
500	MSE	0.50	1.02	1.17	1.50	15.90	21.60	4.79
		(1×10^{-3})	(2×10^{-3})	(1×10^{-3})	(7×10^{-3})	(2×10^{-2})	(8×10^{-2})	(7×10^{-3})
	MAE	4.56	7.79	8.50	6.96	11.23	15.17	16.37
		(7×10^{-3})	(9×10^{-3})	(5×10^{-3})	(9×10^{-3})	(9×10^{-3})	(4×10^{-2})	(4×10^{-3})
	MXDV	47.87	38.85	39.65	146.05	390.08	407.96	193.31
		(8×10^{-2})	(6×10^{-2})	(9×10^{-2})	(5×10^{-1})	(8×10^{-2})	(2×10^{-1})	(3×10^{-1})

4.2. Real Data Analysis

The data obtained from [Investing.com](https://www.investing.com) (last accessed date : 1 October 2021) the exchange rate change of the Icelandic króna (ISK) per US dollar (USD) from January 2004 to December 2015, measured as the monthly average.

The data contained 144 sets of observations associated with the subprime mortgage crisis, which occurred between 2007 and 2008. This crisis refers to a series of economic crises that began with the bankruptcy of the largest mortgage lenders in the United States and caused a credit crunch not only in the United States, but also in the international financial market. One of affected countries was Iceland, whose three largest banks were affected by the crisis in September 2008. To revive the economy, the Icelandic government has implemented various policies, including the nationalization of banks. Consequently, the ISK exchange rate per USD increased. In Figure 6, the top panel shows the starting point for the data. In January 2004, the observed value is 69.345; in March 2008, the observed value increases gradually to 75.805 and then increases significantly to 105.965 in September. In the last observation, performed in December 2016, a continuous increase of the value is observed (129.960).

Because the input variable is a date and not numeric data, 144 samples are equally spaced in the interval $[0, 1]$ for the proposed QP implementation. A degree exceeding four can result in a more complicated interpretation and overfitting. Therefore, we apply $d = 1, 2$ and 3.

Figure 6 shows the fitted results. The top panel of Figure 6 shows a scatterplot of the data. The remaining panels show the results of the QP model fitting when $d = 1, 2$ and 3. It appears that the proposed MDSE based on the BIC detects a change point in September 2008 and fits the flow of the exchange rate well. Prior to 2008, no significant change is detected, but from January 2008, the exchange rate of ISK per USD increases gradually. Subsequently, beginning from September 2008, the exchange rate of ISK per USD increases significantly, as indicated by the vertical black dashed line in the figure (representing September 2008).

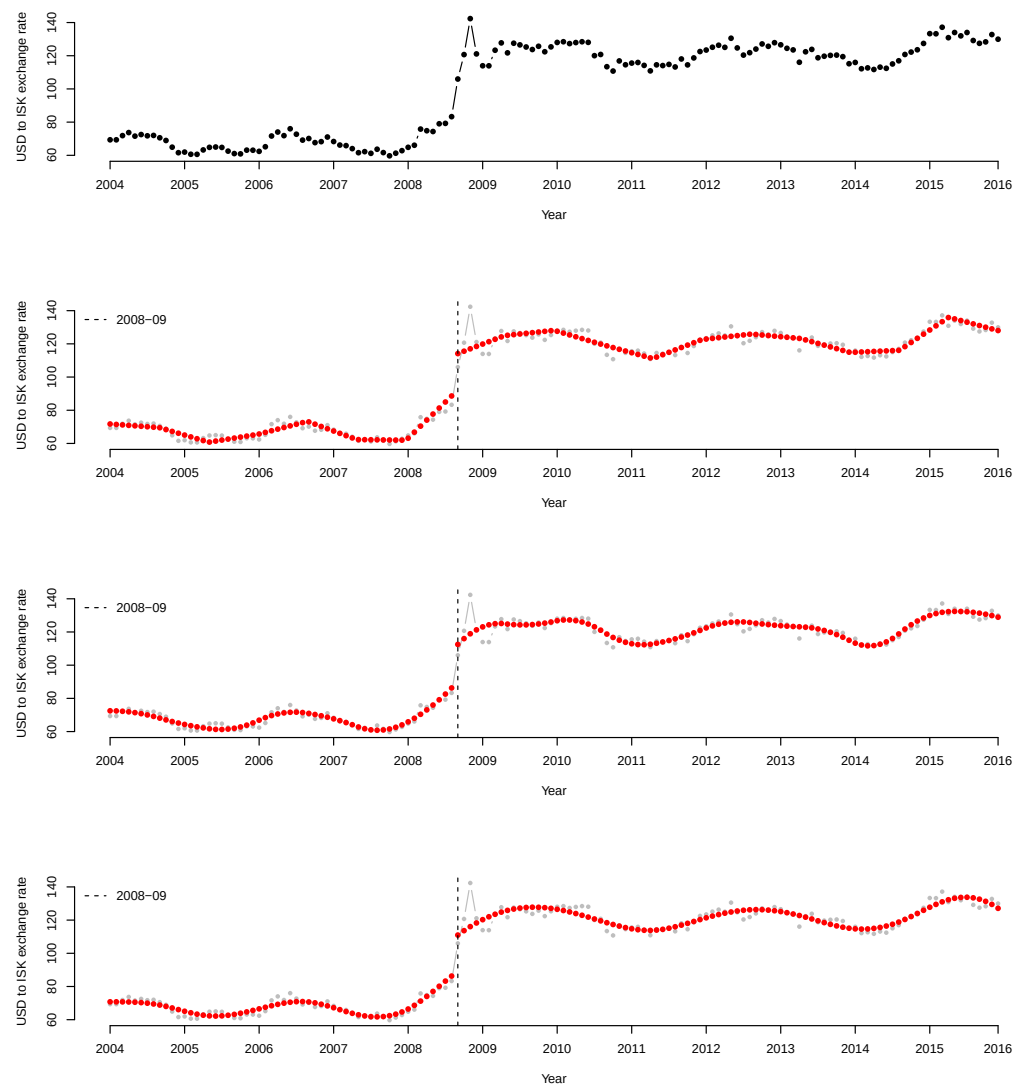


Figure 6. (Top) USD to ISK exchange rate data. The x -axis corresponds to the input variable in a year, i.e., each month from January 2004 to December 2015. The y -axis shows the monthly average of the ISK exchange rate against USD for each month. First panel in the middle, fit result when $d = 1$ (linear); second panel in the middle, fit result when $d = 2$ (quadratic); **bottom panel**, fit result when $d = 3$ (cubic). The bottom three panels are indicated by a vertical black dashed line between 2008 and 2009, as well as red points. The dashed line indicates the change point, whereas the red dots indicate the fitted results.

5. Conclusions

In this study, we develop a nonparametric regression function estimation method with change point detection. In order to provide a data-driven knot selection method, we consider elastic-net-type L^1 -norm penalty for the estimating regression function. Once we expressed the estimator with a linear combination of truncated power splines, the coefficients were estimated by minimizing a penalized residual sum of squares. A new coordinate descent algorithm and an algorithm based on quadratic programming are introduced to handle penalty terms determined by the regression coefficients. In the numerical analysis, we only use an algorithm based on quadratic programming because of the high computational cost of the coordinate descent algorithm when the number of knots increases. To verify the performance of the proposed estimator (MDSE), in the three simulations, we compared MDSE with four methods; the fused lasso (FL), trend filtering

(TF), smoothing spline (SS) and structural model for a times series (ST). As a result, in all simulations, the MDSE detected the change point(s) and fitted linear and cubic trends of the piecewise polynomial well. Furthermore, we performed an actual data analysis about the sharp change in exchange rate of the Icelandic króna per US dollar in September 2008 caused by the subprime mortgage crisis in the United States between 2007 and 2008. In conclusion, MDSE can accurately detect September 2008, the time when a sharp increase in the exchange rate occurred, and can also represent the exchange rate trend before and after September 2008 well.

To improve the proposed coordinate descent algorithm, one may apply the B-spline basis and total variation penalty terms. The B-spline basis function is a numerically superior alternative to the truncated power basis. Because it has small supports, the predictor matrix is sparse and the information matrix is banded (chapter 1 and 8 of [31]). The B-spline-based total variation penalty term is a generalized lasso problem [32,33] that is difficult to implement, but affords high computational power. These approaches will be investigated in future studies.

Supplementary Materials: The following are available at <https://www.mdpi.com/article/10.3390/axioms10040331/s1>, The supplementary files contain a data set and R code to perform the proposed method described in the article (implemented by the software R version 4.0.2.).

Author Contributions: The authors contributed equally to this study; methodology, E.-J.L. (numerical framework) and J.-H.J. (modeling and defining estimator); implementation, E.-J.L. (CDA), J.-H.J. (QP); writing, E.-J.L. (numerical analysis), J.-H.J. (model and estimator). All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (NRF-2020R1G1A1A01100869).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are provided in the Supplementary Materials.

Acknowledgments: The authors wish to thank Jae-Kwon Oh for assistance with the numerical study.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tsybakov, A.B. *Introduction to Nonparametric Estimation*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2008.
2. Fan, J.; Gijbels, I.; Hu, T.C.; Huang, L.S. A study of variable bandwidth selection for local polynomial regression. *Stat. Sin.* **1996**, *6*, 113–127.
3. Efromovich, S. *Nonparametric Curve Estimation: Methods, Theory, and Applications*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2008.
4. Green, P.J.; Silverman, B.W. *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach*; CRC Press: Boca Raton, FL, USA, 1993.
5. Massopust, P. *Interpolation and Approximation with Splines and Fractals*; Oxford University Press, Inc.: Oxford, UK, 2010.
6. James, G.; Witten, D.; Hastie, T.; Tibshirani, R. *An introduction to Statistical Learning*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 112.
7. Wright, S.J. Coordinate descent algorithms. *Math. Program.* **2015**, *151*, 3–34. [[CrossRef](#)]
8. Tseng, P. Convergence of a block coordinate descent method for nondifferentiable minimization. *J. Optim. Theory Appl.* **2001**, *109*, 475–494. [[CrossRef](#)]
9. Tibshirani, R. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. (Methodol.)* **1996**, *58*, 267–288. [[CrossRef](#)]
10. Meier, L.; Van De Geer, S.; Bühlmann, P. The group lasso for logistic regression. *J. R. Stat. Soc. Ser. (Stat. Methodol.)* **2008**, *70*, 53–71. [[CrossRef](#)]
11. Zou, H.; Hastie, T. Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. (Stat. Methodol.)* **2005**, *67*, 301–320. [[CrossRef](#)]
12. Goldfarb, D.; Idnani, A. A numerically stable dual method for solving strictly convex quadratic programs. *Math. Program.* **1983**, *27*, 1–33. [[CrossRef](#)]
13. Osborne, M.; Presnell, B.; Turlach, B. Knot selection for regression splines via the lasso. *Comput. Sci. Stat.* **1998**, *30*, 44–49.

14. Leitenstorfer, F.; Tutz, G. Knot selection by boosting techniques. *Comput. Stat. Data Anal.* **2007**, *51*, 4605–4621. [[CrossRef](#)]
15. Garton, N.; Niemi, J.; Carriquiry, A. Knot selection in sparse Gaussian processes. *arXiv* **2020**, arXiv:2002.09538.
16. Aminikhanghahi, S.; Cook, D.J. A survey of methods for time series change point detection. *Knowl. Inf. Syst.* **2017**, *51*, 339–367. [[CrossRef](#)] [[PubMed](#)]
17. Tibshirani, R.; Saunders, M.; Rosset, S.; Zhu, J.; Knight, K. Sparsity and smoothness via the fused lasso. *J. R. Stat. Soc. Ser. (Stat. Methodol.)* **2005**, *67*, 91–108. [[CrossRef](#)]
18. Turlach, B.A.; Weingessel, A. *Quadprog: Functions to Solve Quadratic Programming Problems*; R Package Version 1.5-8; 2019. Available online: <https://cran.r-project.org/package=quadprog> (accessed on 15 July 2021).
19. Friedman, J.; Hastie, T.; Tibshirani, R. Regularization paths for generalized linear models via coordinate descent. *J. Stat. Softw.* **2010**, *33*, 1–22. [[CrossRef](#)] [[PubMed](#)]
20. Jhong, J.H.; Koo, J.Y.; Lee, S.W. Penalized B-spline estimator for regression functions using total variation penalty. *J. Stat. Plan. Inference* **2017**, *184*, 77–93. [[CrossRef](#)]
21. Bozdogan, H. Model selection and Akaike's information criterion (AIC): The general theory and its analytical extensions. *Psychometrika* **1987**, *52*, 345–370. [[CrossRef](#)]
22. Schwarz, G. Estimating the dimension of a model. *Ann. Stat.* **1978**, *6*, 461–464. [[CrossRef](#)]
23. Luenberger, D.G.; Ye, Y. *Linear and Nonlinear Programming*; Springer: Berlin/Heidelberg, Germany, 1984; Volume 2.
24. Rockafellar, R.T. *Convex Analysis*; Princeton University Press: Princeton, NJ, USA, 2015.
25. Kim, S.J.; Koh, K.; Boyd, S.; Gorinevsky, D. ℓ_1 trend filtering. *SIAM Rev.* **2009**, *51*, 339–360. [[CrossRef](#)]
26. Kim, Y.J.; Gu, C. Smoothing spline Gaussian regression: More scalable computation via efficient approximation. *J. R. Stat. Soc. Ser. (Stat. Methodol.)* **2004**, *66*, 337–356. [[CrossRef](#)]
27. Harvey, A.C. *Forecasting, Structural Time Series Models and the Kalman Filter*; Cambridge University Press: Cambridge, UK, 1990.
28. Arnold, T.B.; Tibshirani, R.J. Efficient implementations of the generalized lasso dual path algorithm. *J. Comput. Graph. Stat.* **2016**, *25*, 1–27. [[CrossRef](#)]
29. Gu, C. Smoothing spline ANOVA models: R package gss. *J. Stat. Softw.* **2014**, *58*, 1–25. [[CrossRef](#)]
30. R Core Team. *R: A Language and Environment for Statistical Computing*; R Core Team: Vienna, Austria, 2021.
31. De Boor, C.; De Boor, C. *A Practical Guide to Splines*; Springer: New York, NY, USA; 1978, Volume 27. [[CrossRef](#)]
32. Roth, V. The generalized LASSO. *IEEE Trans. Neural Netw.* **2004**, *15*, 16–28. [[CrossRef](#)] [[PubMed](#)]
33. Tibshirani, R.J.; Taylor, J. The solution path of the generalized lasso. *Ann. Stat.* **2011**, *39*, 1335–1371. [[CrossRef](#)]