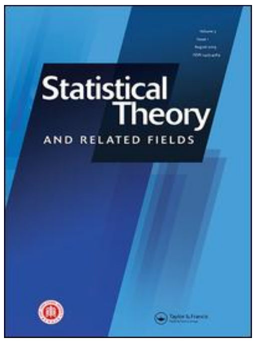

Statistical Theory and Related Fields

ISSN: 2475-4269 (Print) 2475-4277 (Online) Journal homepage: https://www.tandfonline.com/loi/tstf20

Efficient estimation of smoothing spline with exact shape constraints

Vincent Chan, Kam-Wah Tsui, Yanran Wei, Zhiyang Zhang & Xinwei Deng

STATISTICAL THEORY AND RELATED FIELDS
https://doi.org/10.1080/24754269.2020.1722604

Efficient estimation of smoothing spline with exact shape constraints

Vincent Chan[a], Kam-Wah Tsui[a], Yanran Wei[b], Zhiyang Zhang[b] and Xinwei Deng[b]

[a]Department of Statistics, University of Wisconsin-Madison, Madison, WI, USA; [b]Department of Statistics, Virginia Tech, Blacksburg, VA, USA

ABSTRACT
Smoothing spline is a popular method in non-parametric function estimation. For the analysis of data from real applications, specific shapes on the estimated function are often required to ensure the estimated function undeviating from the domain knowledge. In this work, we focus on constructing the exact shape constrained smoothing spline with efficient estimation. The 'exact' here is referred as to impose the shape constraint on an infinite set such as an interval in one-dimensional case. Thus the estimation becomes a so-called semi-infinite optimisation problem with an infinite number of constraints. The proposed method is able to establish a sufficient and necessary condition for transforming the exact shape constraints to a finite number of constraints, leading to efficient estimation of the shape constrained functions. The performance of the proposed methods is evaluated by both simulation and real case studies.

ARTICLE HISTORY
Received 14 February 2019
Revised 10 November 2019
Accepted 24 January 2020

KEYWORDS
Monotonicity; non-linear constraint; non-parametric regression

1. Introduction

In recent years, non-parametric smoothing methods have gained popularity in various science and engineering areas such as economics, biology, smart manufacturing, etc. One advantage of these methods is that they do not assume strong parametric structure for the underlying model. While in the analysis of data from real applications, researchers often demand a specific shape on the estimated function to ensure the estimated function undeviating from their domain knowledge. For example, shape with monotonicity are often required in the estimation of dose-response function (Kelly & Rice, 1990) in medicine. The degradation curve of scaffolds in engineered scaffold fabrication often requires to be monotonic and concave (Zeng, Deng, & Yang, 2016). The estimation of human growth curve (Ducharme & Fontez, 2004) in biometrics and estimation of utility function (Matzkin, 1991) in economics often needs the concavity in shape.

Among various non-parametric smoothers, spline smoothing and kernel smoothing are quite popular. Theoretical and numerical properties of these techniques have been well studied. See Wahba (1990) and Green and Silverman (1993) for thorough discussions of the spline smoothing problem, and Fan and Gijbels (1996) and Wand and Jones (1995) for the kernel smoothing problem. Unlike its unconstrained counterpart, shape constrained smoother has not received large attentions in the statistics literature. As pointed in Delecroix and Thomas-Agnan (2000), most of the isotonic estimates are based on splines rather than on kernels since enforcing the restrictions at the minimisation step appears to be a natural solution.

There are various splines to enable the shape constraints. For example, B-spline is a popular approach because of its special properties: non-decreasing coefficients imply non-decreasing resulting function (Brezger & Steiner, 2003; Dierckx, 1980; He & Shi, 1998; Kelly & Rice, 1990). There are also I-spline methods (Curry & Schoenberg, 1966; Ramsay, 1988) to integrate over non-negative M-splines for constructing monotone smoothers. Meyer (2008) defined the C-splines that at each observation to impose the shape constraint. Combinations of monotonicity and convexity can be imposed by the regression splines. Meyer (2012) also developed penalised splines under shape constraints. (Liao & Meyer, 2017) proposed estimators of change-point based on constrained splines. Meyer (2018) proposed the constrained generalised additive model using iteratively reweighted cone projection algorithm. The smoothing spline is another popular approach, of which the most notable works include (Wang & Li, 2008)'s second order cone programming to create monotone smoothing spline and (Turlach, 2005)'s approach of adaptively adding constraints to create shape constrained smoothing spline.

In this work, we consider the smoothing spline to study the exact shape constraints. Here the meaning of 'exact' is referred as to impose the shape constraint on an infinite set such as an interval in one-dimensional case. It leads to a so-called *semi-infinite optimisation* problem with an infinite number of constraints.

CONTACT Xinwei Deng xdeng@vt.edu

Suppose that the observational data are (x_i, y_i) for $i = 1, \ldots, n$, and we assume $L \leq x_1 < \cdots < x_n \leq U$. The exact shape constrained smoothing spline is defined as the solution of the following optimisation problem:

$$\underset{f}{\text{minimise}} \quad \sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda \int_a^b [f^{(2)}(t)]^2 \, \mathrm{d}t, \tag{1a}$$

$$\text{subject to} \quad f^{(r)}(x) \geq 0 \quad \text{for } x \in [L, U], \tag{1b}$$

where $f^{(r)}(x)$ is the rth derivative of $f(x)$ and $\lambda \geq 0$ is a tuning parameter. The formulation in (1a) without constraint is the well-known smoothing spline problem, the solution of which is known as the *cubic smoothing spline* over the class of twice differential functions. The framework for $r = 1$ and $r = 2$ corresponds to the monotone non-decreasing and convex shape constraint, respectively. The monotone decreasing or concave constraint can be easily obtained by reversing the inequality sign in (1b). One can pursue either one of the constraints or some of them under this framework. For example, non-global constraint, such as convex for $x \leq 0$ and concave for $x > 0$ is possible. A mixed constraint, such as combination of concave and monotone increasing, can also be applied.

The challenges for the estimation in (1) is the constraints are defined on an infinite set, which implies an infinite number of constraints. By taking advantage of the close connection between the natural cubic spline and the smoothing spline, the proposed method utilises a good representation of smoothing spline to establish a sufficient and necessary condition for transforming the exact shape constraints in (1b) to a finite number of constraints. The resultant solution to the case of $r = 2$ is straightforward, and the challenge arises when $r = 1$. To the best of our knowledge, an exact solution for $r = 1$ has yet to be found in the literature. To facilitate the computation of parameter estimation, we also develop efficient algorithm based on matrix approximation for the large data.

The remaining paper is organised as follows. Section 2 revisits the connection between the natural cubic splines and smoothing splines. The proposed exact shape-constraint smoothing spline is detailed in Section 3. An efficient computation algorithm for parameter estimation are developed in Section 4. Sections 5 and 6 evaluate the performance of the proposed method from a simulation study and an application to real life data. We conclude this work with some discussion in Section 7.

2. Revisit of natural cubic spline and smoothing spline

The natural cubic spline (NCS) plays an essential role for the smoothing spline. Suppose that the observed data are $(x_1, y_1), \ldots, (x_n, y_n)$ with $x_1 < \cdots < x_n$. An NCS function $f(x)$ with knots at $x_1, \ldots, x_n$ is a piecewise polynomial of degree up to three with breakpoints at $x_1, \ldots, x_n$. In addition, $f(x)$ is twice continuously differentiable at the knots and linear beyond the boundary.

Let $f(x)$ be a NCS with knots at $x_1, \ldots, x_n$. By definition, $f(x)$ can be expressed as

$$f(x) = \begin{cases} f_0(x) = c_0 x + d_0, & x < x_1, \\ f_i(x) = a_i x^3 & \\ \quad + b_i x^2 + c_i x + d_i, & x_i \leq x < x_{i+1} \\ & \quad \text{for } i = 1, \ldots, n-1, \\ f_n(x) = c_n x + d_n, & x \geq x_n, \end{cases} \tag{2a}$$

with restrictions

$$f_i(x_{i+1}) = f_{i+1}(x_{i+1}), f_i'(x_{i+1}) = f_{i+1}'(x_{i+1}), f_i''(x_{i+1}) = f_{i+1}''(x_{i+1}) \text{ for } i = 0, \ldots, n-1. \tag{2b}$$

The derivatives of $f(x)$ can be obtained by taking derivative on each polynomial and maintaining the relevant constraints. Please refer to Appendix 1 for explicit expressions. This *piecewise polynomial representation* of an NCS in (2) is a key for formulating the shape constraints in Section 3. For estimation of $f(x)$, however, there exists another presentation for computational purpose. Specifically, we first estimate $f(x_1), \ldots, f(x_n)$ by writing them as a linear combination of specific basis functions and estimate the corresponding coefficients. Then we can utilise the *value-second derivative representation* (Green & Silverman, 1993) to recover the entire function $f(x)$. As a result, the problem in (1a) can be converted into a ridge regression-like problem that can be efficiently solved.

Let $\mathbf{1}_n$ be length-n vector of ones, $\boldsymbol{x} = (x_1, \ldots, x_n)^T$ and $\boldsymbol{g} = (f(x_1), \ldots, f(x_n))^T$. Without loss of generality, we assume $\boldsymbol{x}$ is centred with zero mean. We can construct the banded matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ according to Equations (A1) and (A2) in Appendix A.2. The *linear mixed model representation* described in Appendix A.3 allows us to rewrite the NCS formulation in (2) as

$$\boldsymbol{g} = \mathbf{1}_n \theta_0 + \boldsymbol{x}\theta_1 + \boldsymbol{A}\boldsymbol{\beta}, \tag{3}$$

here $\boldsymbol{A} = \boldsymbol{Q}(\boldsymbol{Q}^T\boldsymbol{Q})^{-1}\boldsymbol{R}^{1/2} \in \mathbb{R}^{n \times (n-2)}$. The θ_1, θ_2 and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_{n-2})'$ are parameters. By construction, matrix $\boldsymbol{A}$ has full rank. It is easy to check that $\mathbf{1}_n^T \boldsymbol{x} = 0$, $\mathbf{1}_n^T \boldsymbol{A} = 0$ and $\boldsymbol{x}^T \boldsymbol{A} = 0$. Hence $\{\mathbf{1}_n, \boldsymbol{x}, \boldsymbol{A}\}$ form a basis of the n-dimensional euclidean space. Furthermore, if we define matrix $\boldsymbol{K} = \boldsymbol{Q}\boldsymbol{R}^{-1}\boldsymbol{Q}^T \in \mathbb{R}^{n \times n}$, then we get:

$$\int [f''(t)]^2 \mathrm{d}t = \boldsymbol{g}^T \boldsymbol{K} \boldsymbol{g} = \boldsymbol{\beta}^T \boldsymbol{\beta}. \tag{4}$$

It is worth to pointing out that the underlying model for the smoothing spline can be considered as a natural cubic spline with knots at $x_1, \ldots, x_n$ and at most

$k = 2[n/2] + 2$ additional knots of which the locations are unknown (Utreras, 1985). Here we made a mild assumption that $f(x)$ is a natural cubic spline with knots at $x_1, \ldots, x_n$. Combining this assumption with results in (3) and (4), the smoothing spline expressed in (1a) is equivalent to the problem of

$$\underset{(\theta_0,\theta_1,\boldsymbol{\beta})}{\text{minimise}} \quad \sum_{i=1}^{n}(y_i - \theta_0 - x_i\theta_1 - \boldsymbol{A}_i\boldsymbol{\beta})^2 + \lambda_1\boldsymbol{\beta}^T\boldsymbol{\beta}, \tag{5a}$$

where $\boldsymbol{A}_i$ is the ith row of matrix $\boldsymbol{A}$. This is simply a ridge regression problem with response vector $\boldsymbol{y}$ and covariates matrix $(\boldsymbol{1}_n, \boldsymbol{x}, \boldsymbol{A})$ without penalising θ_0 and θ_1.

When one obtain the estimate of $\boldsymbol{\theta} = (\theta_0, \theta_1, \boldsymbol{b})^T$, the entire estimated function can be constructed by following the procedure in Appendix A.3. one can actually obtain the piecewise polynomial representation of the estimated function by following the steps in Appendix A.4. That is, the piecewise polynomial representation is fully specified when $(\theta_1, \theta_2, \boldsymbol{\beta})$ are known.

3. Exact shape-constrained smoothing spline

To have the exact shape constraint, one major difficulty is that the inequality constraint in (1b) cannot be guaranteed by simply enforcing constraints at $x_1, \ldots, x_n$. Due to these challenges, many computable 'solutions' to the shape constrained smoothing spline problem described in ((1a) and (1b)) are approximations of some kind. Some approximate by assuming the resulting function being a natural cubic spline with knots at all data points (Turlach, 2005; Wang & Li, 2008), others approximate by discretisation of infinite constraint (1b) to a finite number of constraints (Mammen & Thomas-Agnan, 1999; Nagahara & Martin, 2013; Villalobos & Wahba, 1987).

It is known that the solution to the exact shape-constraint smoothing spline in (1a) and (1b) is a natural cubic spline with knots at $x_1, \ldots, x_n$ and at most $k = 2[n/2] + 2$ additional knots of which the locations are unknown as proved in Theorem 3.3 in (Utreras, 1985). Unfortunately, it does not provide much practical use because of the unknown locations of those additional knots. However, it sheds some lights that a natural cubic spline with knots at all data points is an adequate approximation to the theoretically correct model proved by Utreras (1985). Therefore, *we develop our proposed method under the consideration that the estimated model is a natural cubic spline with knots at all data points*. Specifically, we propose a representation only using $n-1$ constraints that is equivalent to the infinite constraint (1b) for $r = 1$or$2$. Compared to Turlach (2005) who took an adaptive approach to adding constraints and thus changing the underlying quadratic program for parameter estimation, the proposed method optimises over a larger underlying model space yet it maintains the exactness of the shape constraint. Different from Wang Li (2008) who only works on monotonicity constraint ($r = 1$), our proposed method also works on convexity constraint ($r = 2$) and can easily be extended to mixed and non-global constraint.

The key idea of our proposed method is to utilise the piecewise polynomial representation of NCS to provide a sufficient and necessary condition in converting constraint (1b) for $r = 2$ and $r = 1$ to the form of $c(\boldsymbol{\theta}; \boldsymbol{x}) \succeq \boldsymbol{0}$, where we define notation $\succcurlyeq$ as element-wise greater than or equal to. Then we can express the shape constrained smoothing spline problem as

$$\underset{(\theta_0,\theta_1,\boldsymbol{\beta})}{\text{minimise}} \quad \sum_{i=1}^{n}(y_i - \theta_0 - x_i\theta_1 - \boldsymbol{A}_i\boldsymbol{\beta})^2 + \lambda_1\boldsymbol{\beta}^T\boldsymbol{\beta}, \tag{5a}$$

$$\text{subject to} \quad c(\boldsymbol{\theta}; \boldsymbol{x}) \succeq \boldsymbol{0}. \tag{5b}$$

The formulation above can be optimised by many standard optimisation methods that take non-linear constraints.

The shape constraint (1b) for $r = 1$ (monotonicity) and $r = 2$ (convexity) are presented below as Theorems 3.1 and 3.2, respectively. The mixed constraints can be achieved by combining the corresponding constraints.

Theorem 3.1: *For the smoothing spline, the monotone non-decreasing constraint, defined as $f'(x) \geq 0$, holds if and only if constraint* (5b) *with*

$$c(\boldsymbol{\theta}; \boldsymbol{x}) = (c_1(\boldsymbol{\theta}; \boldsymbol{x}), \ldots, c_{n-1}(\boldsymbol{\theta}; \boldsymbol{x}))^T,$$

where

$$c_i(\boldsymbol{\theta}; \boldsymbol{x}) = \begin{cases} \min\left(f'(x_i), f'(x_{i+1}), f'\left(\frac{-b_i}{3a_i}\right)\right), \\ \quad \textit{if } \frac{-b_i}{3a_i} \in (x_i, x_{i+1}) \textit{ and } a_i > 0 \\ \min(f'(x_i), f'(x_{i+1})), \quad \textit{otherwise} \end{cases} \tag{6}$$

holds.

Proof: Based on the polynomial representation of NCS, it is easy to get the first derivative $f'(x)$ as

$$f'(x) = \begin{cases} f_0'(x) = c_0, & x < x_1, \\ f_i'(x) = 3a_ix^2 & \\ \quad +2b_ix + c_i, & x_i \leq x < x_{i+1} \text{ for} \\ & \quad i = 1, \ldots, n-1, \\ f_n'(x) = c_n, & x \geq x_n, \end{cases}$$

$$\text{with restrictions } f_i'(x_{i+1}) = f_{i+1}'(x_{i+1}), f_i''(x_{i+1}) = f_{i+1}''(x_{i+1}), \text{ for } i = 0, \ldots, n-1. \tag{7}$$

Clearly, $f'(x)$ is a continuous piecewise polynomial of at most second order on each interval $[x_1, x_2), \ldots,$

$[x_{n-1}, x_n)$, and constant on the boundary interval $(-\infty, x_1)$, and $[x_n, \infty)$. For $i = 1, \dots, n-1$, if $a_i = 0$, $f_i'(x)$ is a linear function on $[x_i, x_{i+1})$, so $f'(x) \geq \min(f'(x_i), f'(x_{i+1}))$ on the interval. If $a_i \neq 0, f_i'(x)$ is a parabola on $[x_i, x_{i+1})$ with stationary point at $-b_i/3a_i$, specifically

(1) if $a_i < 0$, $f_i'(x)$ is concave. So regardless of the location of the stationary point, $f'(x) \geq \min(f'(x_i), f'(x_{i+1}))$ on $[x_i, x_{i+1})$.
(2) if $a_i > 0$, $f_i'(x)$ is a convex parabola where the stationary point may or may not lie on $[x_i, x_{i+1})$.
  (a) If the stationary point is not on the interval, $f_i'(x)$ is monotone on $[x_i, x_{i+1})$, so $f'(x) \geq \min(f'(x_i), f'(x_{i+1}))$ on the interval.
  (b) If the stationary point is on the interval, global minimum could be at either the boundary or the stationary point, so $f'(x) \geq \min(f'(x_i), f'(x_{i+1}), f'(-b_i/3a_i))$ on the interval.

Non-negativity on the boundary interval $(-\infty, x_1)$ and $[x_n, \infty)$ hold if $c_0 \geq 0$ and $c_n \geq 0$. But no extra constraint is needed because by continuity of $f'(x)$, $c_0 = f_1'(x_1)$ and $c_n = f_{n-1}'(x_n)$, non-negativitiy is already ensured by $c_1(\boldsymbol{\theta}; \boldsymbol{x}) \geq 0$ and $c_{n-1}(\boldsymbol{\theta}; \boldsymbol{x}) \geq 0$.

If $c(\boldsymbol{\theta}, \boldsymbol{x}) \succeq \mathbf{0}$, then each piecewise polynomial, $f_i'(x)$ for $i = 0, \dots, n$, is non-negative, which in turn implies that the whole function $f'(x)$ must be non-negative. Therefore, validity of inequality (6b) implies the monotone non-decreasing constraint.

The other direction is obvious by definition. ■

Theorem 3.2: *For the smoothing spline, the convexity constraint, defined as $f'(x) \geq 0$, holds if and only if constraint* (5b) *with*

$$c(\boldsymbol{\theta}; \boldsymbol{x}) = (c_1(\boldsymbol{\theta}; \boldsymbol{x}), \dots, c_{n-1}(\boldsymbol{\theta}; \boldsymbol{x}))^T,$$

where

$$c_i(\boldsymbol{\theta}; \boldsymbol{x}) = \min(f''(x_i), f''(x_{i+1})), \tag{8}$$

holds.

Proof: Based on the polynomial representation of NCS, it is easy to get the first derivative $f''(x)$ as

$$f''(x) = \begin{cases} f_0''(x) = 0, & x < x_1, \\ f_i''(x) = 6a_i x + 2b_i, & x_i \leq x < x_{i+1} \quad \text{for} \\ & \quad i = 1, \dots, n-1, \\ f_n''(x) = 0, & x \geq x_n, \end{cases}$$

$$\text{with restrictions } f_i''(x_{i+1}) = f_{i+1}''(x_{i+1}),$$
$$\text{for } i = 0, \dots, n-1. \tag{9}$$

The $f''(x)$ is a continuous piecewise linear polynomial on each interval $(-\infty, x_1)$, $[x_1, x_2), \dots, [x_{n-1}, x_n)$ and $[x_n, \infty)$. Since $f''(x) = 0$ for any $x \leq x_1$ and $x \geq x_n$, we only need to consider $f''(x)$ when $x \in (x_1, x_n)$. For any i, linearity of $f_i''(x)$ implies $f''(x) \geq \min(f''(x_i), f''(x_{i+1}))$ on interval $[x_i, x_{i+1})$. If $c(\boldsymbol{\theta}; \boldsymbol{x}) \succeq \mathbf{0}$, each piecewise polynomial, $f_i''(x)$ for $i = 0, \dots, n$, is non-negative, which in turn implies that the whole function $f''(x)$ is non-negative. Therefore, inequality (6b) implies convexity constraint.

The other direction is obvious by definition. ■

The above theorems are defined for global constraint, i.e. constraint that holds on the entire domain of the function $[L, U]$. To extend the results to mixed constraint and non-global constraint, we can easily apply Theorem 3.2 and Theorem 3.1 to different local intervals $[L_j, U_j]$, where $L \leq L_j \leq U_j \leq U$. In addition, we can impose up to second-order smooth constraint on the boundary of local intervals. A general procedure is described as follows:

Step 1. Let L_i for $i = 1, \dots, M$ be the points where monotonicity/convexity changes, and $L_0 = L$ and $L_{M+1} = U$. Partition the domain of $f(x)$ into $(L_0, L_1], \dots, (L_M, L_{M+1})$. As a result, $f(x)$ on each interval only requires one shape constraint.

Step 2. For each interval, impose constraint according to Theorem 3.2 and Theorem 3.1.

Step 3. For $i = 1, \dots, M$, add $f'(L_i) = 0$ for monotone constraint or $f''(L_i) = 0$ for convexity constraint.

The importance of Step 3 is that it can prevent the stationary/inflection point of the estimated function from floating between the knots immediately smaller and larger than the point where monotonicity/convexity changes.

4. Efficient algorithm for parameter estimation

Note that the optimisation problem described in (6a) and (6b) is a quadratic programming with nonlinear constraints. Using *Python*, our implementation algorithm is based on the *ralg* solver under the *OpenOPT* platform. The *ralg* solver resembles the quasi-Newton Method with adaptive space dilation developed by Shor and Zhurbenko (1971). Two advantages for this choice are that it accepts user-provided first-derivative and allows large number of constraints.

4.1. Computation of large data

When the number of observation n is large, the growing dimension of the $n \times (n-2)$ matrix $\boldsymbol{A}$ and vector of constraints $c(\boldsymbol{\theta}; \boldsymbol{x})$ are the bottleneck for efficient computation. To address this challenge, we consider to approximate the matrix $\boldsymbol{A}$ by a $n \times K$ dimensional matrix $\boldsymbol{A}^*$, where $K \leq n-2$ is independent of n. It also allows the dimension of $\boldsymbol{\theta}$ to be some fixed integer $K+2 \leq n$.

Following Wand and Ormerod's (2008) *mixed model* formulation in the semiparametric regression, we adopt a good way to approximate $\boldsymbol{A}$ with $\boldsymbol{A^*} = \boldsymbol{BL}$, where $\boldsymbol{B}$ is an $n \times (K+4)$ matrix and $\boldsymbol{L}$ is an $(K+4) \times (K+2)$ matrix.

The construction of $\boldsymbol{B}$ is described as follows. First we define $\kappa_1, \ldots, \kappa_{K+8}$ as

$$\begin{aligned} a &= \kappa_1 = \kappa_2 = \kappa_3 = \kappa_4 = x_1 - \epsilon, \kappa_5 = x_1, \\ &\kappa_6, \ldots, \kappa_{K+3} \text{ to be the} \\ &\times \left(\frac{1}{K+1}, \frac{2}{K+1}, \ldots, \frac{K}{K+1} \right) \\ &\times 100^{th} \text{ percentile of } x_1, \ldots, x_n, \\ \kappa_{K+4} &= x_n, b = \kappa_{K+5} = \kappa_{K+6} \\ &= \kappa_{K+7} = \kappa_{K+8} = x_n + \epsilon. \end{aligned}$$

Then we construct B-spline basis functions (Hastie, Tibshirani, & Friedman, 2001) $B_{1,4}(x), \ldots, B_{K+4,4}(x)$ based on knots sequence $\kappa_1, \ldots, \kappa_{K+8}$ as

$$B_{i,1}(x) = \begin{cases} 1 & \text{if } x \in [\kappa_i, \kappa_{i+1}) \\ 0 & \text{otherwise,} \end{cases}$$

for $i = 1, \ldots, K+7$, and

$$\begin{aligned} B_{i,m}(x) = &\frac{x - \kappa_i}{\kappa_{i+m-1} - \kappa_i} B_{i,m-1}(x) \\ &+ \frac{\kappa_{i+m} - x}{\kappa_{i+m} - \kappa_{i+1}} B_{i+1,m-1}(x), \end{aligned}$$

for $m = 2,\ 3,\ 4$ and $i = 1, \ldots, K+8-m$. Consequently, the matrix $\boldsymbol{B}$ is constructed with its $(i,j)^{th}$ entry $b_{i,j} = B_{j,4}(x_i)$.

The construction of matrix $\boldsymbol{L}$ is described as follows. First we define an $(K+4) \times (K+4)$ matrix $\boldsymbol{\Omega}$ with its $(i,j)^{th}$ entry as

$$\boldsymbol{\Omega}_{i,j} = \int_a^b B''_{i,4}(x) B''_{j,4}(x)\, \mathrm{d}x, \tag{10}$$

where function $B''_{i,4}(x)$ is the second derivative function of $B_{i,4}(x)$ for $i = 1, \ldots, K+4$ as following:

$$\begin{aligned} B''_{i,4}(x) = 6 \Bigg\{ &\frac{B_{i,2}(x)}{(\kappa_{i+3} - \kappa_i)(\kappa_{i+2} - \kappa_i)} \\ &- \left[\frac{1}{(\kappa_{i+4} - \kappa_{i+1})(\kappa_{i+3} - \kappa_{i+1})} \right. \\ &\left. + \frac{1}{(\kappa_{i+3} - \kappa_i)(\kappa_{i+3} - \kappa_{i+1})} \right] B_{i+1,2}(x) \\ &+ \frac{B_{i+2,2}(x)}{(\kappa_{i+4} - \kappa_{i+1})(\kappa_{i+4} - \kappa_{i+2})} \Bigg\} \end{aligned}$$

Based on the spectral decomposition, $\boldsymbol{\Omega}$ can be written as $\boldsymbol{\Omega} = \boldsymbol{UDU}^T$, where $\boldsymbol{U} \in \mathbb{R}^{(K+4)\times(K+4)}$ is an orthogonal matrix and $\boldsymbol{D} \in \mathbb{R}^{(K+4)\times(K+4)}$ is diagonal matrix with $K+2$ positive entries and two zero entries on the diagonal. Let $\boldsymbol{d}$ be a vector that contains the $K+2$ positive entries in matrix $\boldsymbol{D}$, and matrix $\boldsymbol{U}_x \in \mathbb{R}^{(K+4)\times(K+2)}$ contains the columns in $\boldsymbol{U}$ of which their positions correspond to those positive entries in $\boldsymbol{D}$. Then $\boldsymbol{L}$ is constructed as $\boldsymbol{L} = \boldsymbol{U}_x \mathrm{diag}(\boldsymbol{d}^{-\frac{1}{2}})$, where $\mathrm{diag}(\boldsymbol{d})$ denotes a diagonal matrix with diagonal entries equal to vector $\boldsymbol{d}$.

Wand and Ormerod (2008) shows that when $K = n-2, \boldsymbol{A}^* = \boldsymbol{BL} = \boldsymbol{A}$. When $K < n-2$, by substituting matrix $\boldsymbol{A}$ by $\boldsymbol{A}^*$, the length of parameter vector $\boldsymbol{\theta}^* = (\theta_0, \theta_1, \boldsymbol{b}^*)$ reduces to $K+2$ since the length of vector $\boldsymbol{b}^*$ is K. Another advantage of using $\boldsymbol{A}^*$ is that other than the fact that $K \leq n$, the choice of K is independent to n.

The essence of this approximation attributes to the construction of matrix $\boldsymbol{B}$ in Step 1(c) above. The choice of K controls the length of knots sequence being used in the construction of the B-spline basis functions. When K is at its maximum of $n-2$, all data are used as the knots sequence. Then no approximation will occur. When $K < n-2$, a proper subset of data is used as the knots sequence. One can understand this reduction in the length of knots sequence as approximating the full data with a properly chosen (equally spaced in terms of percentile) subset of data. As a result of this approximation, there is a reduction in dimension from $\boldsymbol{A}$ to $\boldsymbol{A}^*$. For a proper choice of K, a large value of K close to n may not lead to significant computational gain. While a small value of K could make the approximation low accurate. From our empirical experience, $K = 30$ provides a good balance between computational gain and approximation quality.

4.2. *Tuning parameter selection*

The tuning parameter λ controls the smoothness of the estimated function. As $\lambda \to \infty$, the estimated function approaches linearity (smoothest); whereas when $\lambda \to 0$, the estimated function tends to the natural cubic spline interpolant (roughest). Although the incorporation of appropriate monotonicity and/or convexity constraints helps smooth out unnecessary roughness in the estimated function, the choice of tuning parameter λ for the exact shape constrained smoothing spline is still important in obtaining a good fit. In this work, we select the optimal value of λ that minimises mean squared error using k-fold cross-validation.

5. Simulations

In this section, we evaluate the performance of our proposed method, the shape constraint smoothing spline (SCSS), under various non-linear functions and error combinations. The methods of comparison include the Brunk's isotonic non-parametric estimator (Brunk), the unconstrained smoothing spline (SS), the proposed SSCS with global monotone non-decreasing constraint (SSCS-Monotone), the proposed SSCS with problem specific convexity constraint (SSCS-Convex), and the proposed SSCS with global monotone non-decreasing and problem specific convexity constraints (SSCS-Mixed). In addition, we also compare the regression spline under the aforementioned three types of constraints, respectively, denoted as RS-Monotone, RS-Convex, and RS-Mixed. We also compare the B-spline method under the monotone non-decreasing constraint (BS-Monotone) and problem specific convexity constraint (BS-Convex). The regression spline and the B-spline methods are implemented using R Package *cgam* and *spline2*, respectively. Note that Wang and Li (2008) kindly provided their code for comparison, but requires a commercial optimiser, MOSEK, which we currently have no access to.

The simulation data are generated from the following model:

$$y_i = f(x_i) + \epsilon_i, \quad \text{i} = 1, \ldots, 50,$$

where x_i distributes uniformly between -10 to 10. The function $f(x)$ varies among examples as follows:

- Example 1: $f(x) = 1/(1 + e^{-x})$ is an increasing function which is convex for $x < 0$ and concave when $x > 0$,
- Example 2: $f(x) = x^3/10^3$ is an increasing function which is concave for $x < 0$ and convex when $x > 0$,
- Example 3: $f(x) = 0I_{-10 \le x \le -3} + 0.2I_{-3 < x \le 0} + 0.5I_{0 < x \le 5} + 0.8I_{5 < x \le 8} + 1I_{8 < x \le 10}$ is a non-decreasing step function,
- Example 4: $f(x) = (20x^2 + x^3)/3000$ is a non-monotone function which is convex for $x > \frac{-20}{3}$ and concave when $x < \frac{-20}{3}$,
- Example 5: $f(x) = (e^{x/20} - e^{-10/20})/e^{10/20} - e^{-10/20}$ is an increasing function which is convex everywhere.

Figure 1 shows the visualisation of above functions. For each example above, three different distributions for ϵ are considered: 1) the normal distribution $N(0, \sigma^2)$; 2) student t distribution with 10 degree of freedom; and 3) Beta distribution $Beta(3, 2)$. These error distributions have zero mean and standard deviation $\sigma = 0.4$. These simulation setup is identical to Wang and Li (2008), except we added a globally convex function in Example 5. For each setting, the simulation is repeated for 500 times. The mean squared prediction error $MSPE(\hat{f}) = 1/n \sum_{i=1}^{n} (\hat{f}(x_i) - f(x_i))^2$ is used as an evaluation criterion.

Table 1 reports the averaged MSPE ($\times 100$) and its standard error ($\times 100$) over 500 repetitions for methods in comparison. It is clear that the proposed methods with appropriate constraint have smaller MSPE than other methods in comparison. Note that it is important to impose appropriate constraint. In Example 4 which has a quadratic-shaped function, The performance of SCSS-Monotone is not as good as the SS since the monotone constraint is not proper here. When imposing the convexity constraint, the SCSS-convex performs much better than other methods. It is worth pointing out the B-spline method has comparable performance to the proposed method in Example 1, but not as good as the proposed method in other examples.

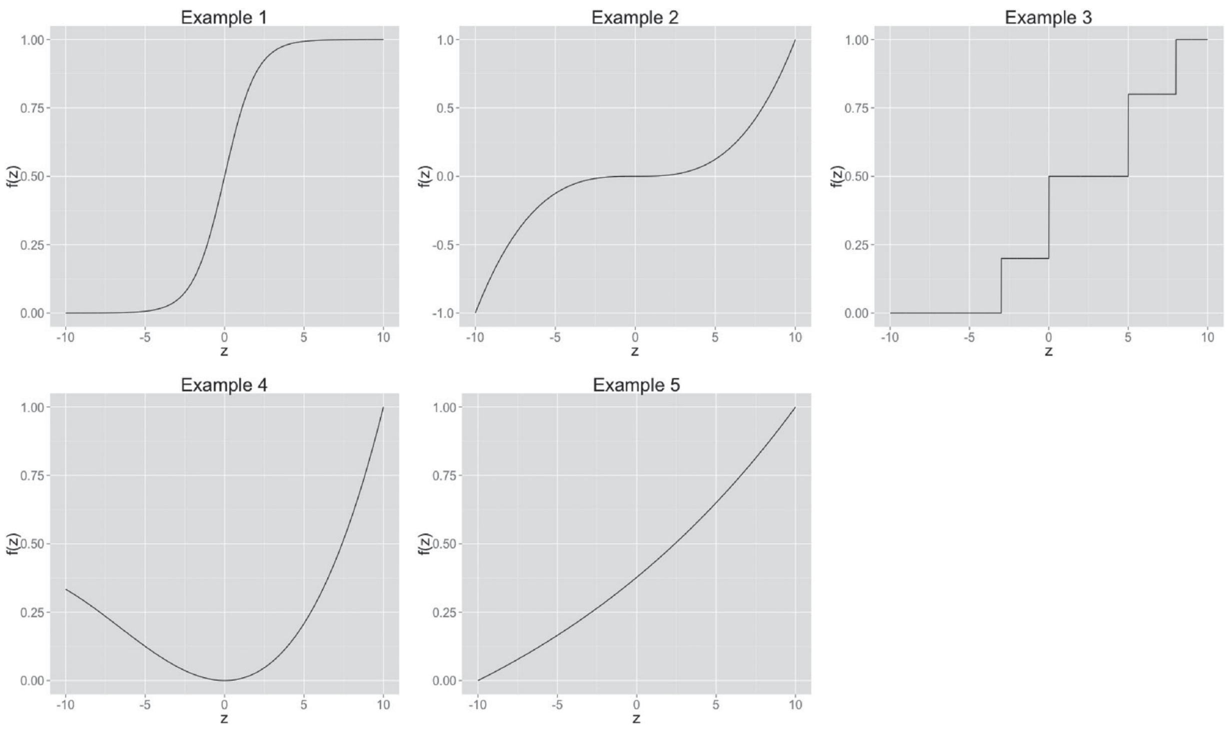

Figure 1. Comparison of the five true functions used in simulation studies.

Table 1. Simulation studies comparing SCSS and other estimators under different functions and errors settings.Averaged MSPE ($\times$100) and its standard error ($\times$100) over 500 repetitions are reported. NA entries correspond to methods with no applicable settings.

E.g.		Brunk's	(SE)	SS	(SE)	RS-Monotone	(SE)	RS-Convex	(SE)	RS-Mixed	(SE)	BS-Monotone	(SE)	BS-Convex	(SE)	SCSS-Monotone	(SE)	SCSS-Convex	(SE)	SCSS-Mixed	(SE)
1	normal	2.69	(0.08)	1.90	(0.05)	1.72	(0.05)	1.90	(0.05)	1.89	(0.04)	1.62	(0.02)	1.60	(0.02)	1.56	(0.05)	1.67	(0.07)	1.66	(0.08)
	t	2.75	(0.08)	1.96	(0.05)	1.74	(0.06)	1.91	(0.06)	1.89	(0.05)	1.61	(0.03)	1.60	(0.03)	1.66	(0.05)	1.74	(0.10)	1.73	(0.09)
	beta	2.87	(0.07)	1.97	(0.05)	1.72	(0.05)	1.89	(0.04)	1.89	(0.04)	1.62	(0.02)	1.61	(0.02)	1.67	(0.05)	1.69	(0.05)	1.67	(0.07)
2	normal	2.90	(0.07)	1.83	(0.05)	1.74	(0.05)	1.83	(0.05)	1.83	(0.05)	1.79	(0.03)	2.11	(0.06)	1.69	(0.05)	1.65	(0.06)	1.50	(0.05)
	t	3.07	(0.08)	1.84	(0.05)	1.74	(0.06)	1.83	(0.06)	1.83	(0.05)	1.79	(0.04)	2.11	(0.06)	1.71	(0.05)	1.69	(0.07)	1.51	(0.05)
	beta	2.91	(0.06)	1.88	(0.05)	1.74	(0.04)	1.83	(0.05)	1.83	(0.04)	1.79	(0.03)	2.12	(0.06)	1.71	(0.05)	1.72	(0.05)	1.53	(0.05)
3	normal	2.87	(0.07)	1.67	(0.04)	1.78	(0.05)	1.76	(0.05)	1.75	(0.04)	1.64	(0.02)	1.64	(0.02)	1.57	(0.03)	NA	(NA)	NA	(NA)
	t	2.86	(0.07)	1.71	(0.04)	1.79	(0.06)	1.77	(0.06)	1.75	(0.05)	1.64	(0.03)	1.64	(0.03)	1.62	(0.04)	NA	(NA)	NA	(NA)
	beta	2.85	(0.07)	1.70	(0.04)	1.78	(0.04)	1.76	(0.04)	1.75	(0.04)	1.64	(0.02)	1.64	(0.02)	1.63	(0.04)	NA	(NA)	NA	(NA)
4	normal	2.99	(0.07)	1.49	(0.06)	1.76	(0.04)	1.73	(0.05)	1.76	(0.04)	1.62	(0.02)	1.70	(0.02)	1.87	(0.04)	1.34	(0.05)	NA	(NA)
	t	2.95	(0.07)	1.49	(0.05)	1.77	(0.05)	1.74	(0.06)	1.76	(0.05)	1.62	(0.03)	1.70	(0.03)	1.86	(0.04)	1.34	(0.05)	NA	(NA)
	beta	2.97	(0.07)	1.47	(0.05)	1.76	(0.04)	1.73	(0.04)	1.76	(0.04)	1.62	(0.02)	1.70	(0.02)	1.91	(0.04)	1.42	(0.06)	NA	(NA)
5	normal	2.36	(0.06)	1.02	(0.04)	1.72	(0.04)	1.70	(0.05)	1.69	(0.04)	1.60	(0.02)	1.60	(0.02)	0.92	(0.03)	0.91	(0.08)	0.76	(0.03)
	t	2.39	(0.07)	1.00	(0.04)	1.73	(0.05)	1.71	(0.05)	1.70	(0.05)	1.60	(0.03)	1.60	(0.03)	0.92	(0.03)	0.88	(0.04)	0.74	(0.03)
	beta	2.25	(0.06)	0.98	(0.04)	1.72	(0.04)	1.70	(0.04)	1.69	(0.04)	1.60	(0.02)	1.60	(0.02)	0.88	(0.03)	0.87	(0.06)	0.82	(0.06)

Notes: Averaged MSPE ($\times$100) and its standard error ($\times$100) over 500 repetitions are reported. NA entries correspond to methods with no applicable settings.

Table 2. Simulation studies measuring the convergence of SCSS based on the integrated mean squared errors.

E.g. 5		50	100	200	300
Normal	Monotone	6.81	5.82	2.44	2.29
	Convex	13.33	4.23	2.44	1.18
	Mixed	4.95	2.41	1.39	1.17

Figure 5 to 10 in Appendix 3 report the estimator percentiles (2.5% and 97.5%) of SCSS and other estimators for Example 4 and Example 5. The SCSS-Monotone, SCSS-Convex and SCSS-Mixed produce slightly smoother percentile intervals compared to other methods, especially at the left and right boundaries on the x axis. Example 4 reveals the behaviour of SCSS under a mis-specified monotone constraint. On the interval $[-10, 0]$, the true function in Example 4 is monotone decreasing but SCSS-Monotone is constrained to be non-decreasing.

To further examine the rate of convergence, we consider a numerical study to check the proposed method's convergence as the sample size gets large. Taking Example 5 for elaboration, we allow the same size increasing gradually from $n = 50$ to $n = 300$. At each given sample size, we compare the discrepancy between $f(x)$ and $\hat{f}(x)$ by $\int (f(x_i) - \hat{f}(x_i))^2 dx$, Table 2 shows that as sample size increases, the function estimated by SCSS is getting closer to the true function. Figure 2 reports the convergence of the estimated function in Example 5 under the normal error and convex constraint. Results for the other two constraints can be found in the Appendix 3.

6. Real data analysis

In this section, we evaluate the performance of the proposed SCSS methods in comparison with the regular smoothing spline (SS). The Auto MPG Data from UCI Machine Learning Repository (Lichman, 2013) is used for our demonstration. This dataset concerning fuel consumption contains 398 observations with nine attributes: fuel consumption (miles per gallon), number of cylinders, engine displacement (cubic inches), horsepower, vehicle weight (pounds), time to accelerate from 0 to 60 mph (sec.), model year (modulo 100), origin of car (1. American, 2. European, 3. Japanese) and car names. For both methods, the optimal value of tuning parameter λ by minimising the average mean squared error from 10-fold cross-validation.

We first analyse relationship between the weight (*weight*) and the fuel consumption (*mpg*) of vehicles. Figure 3 confirms the intuition of a negative correlation between *weight* and *mpg*. Without any constraint, SS provided a monotone estimate that is consistent with the intuition. On the other hand, it is assuring to see that SCSS-Monotone provides an estimate that almost overlaps its unconstrained counterpart.

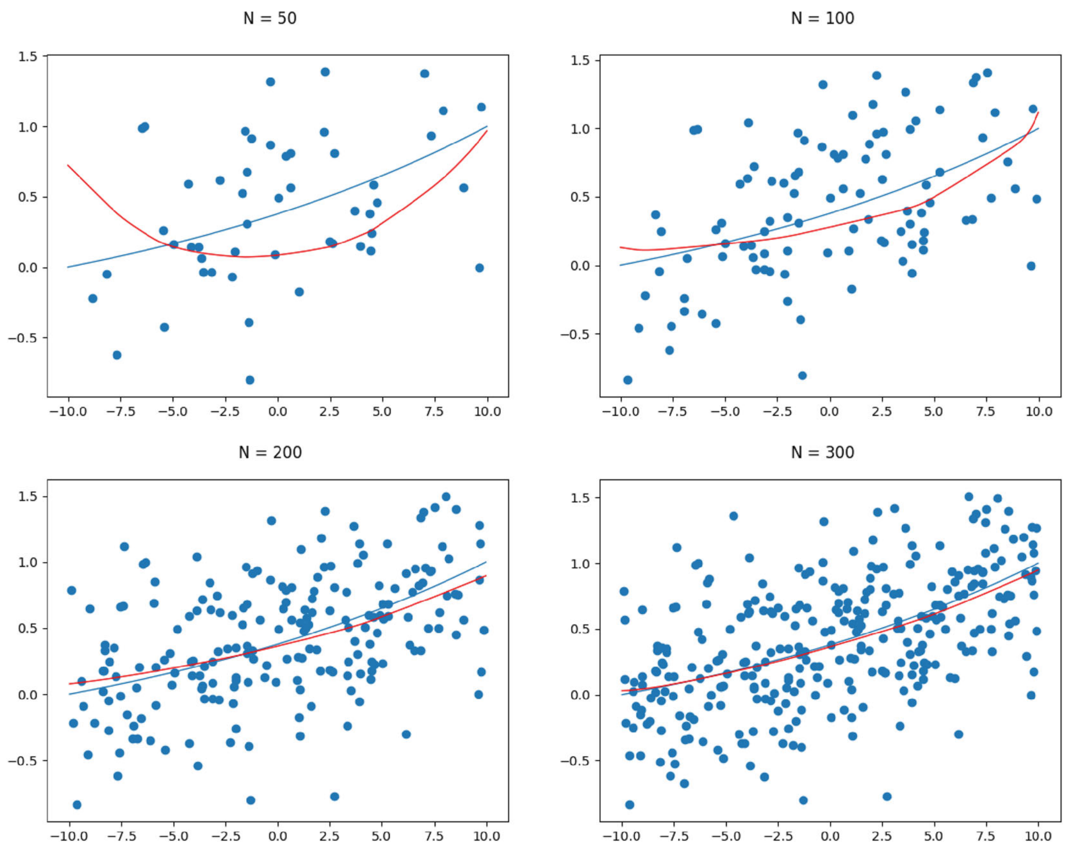

Figure 2. SCSS convergence for function $f(x) = (e^{x/20} - e^{-10/20})/(e^{10/20} - e^{-10/20})$ in Example 5 under normal error and convex constraint.

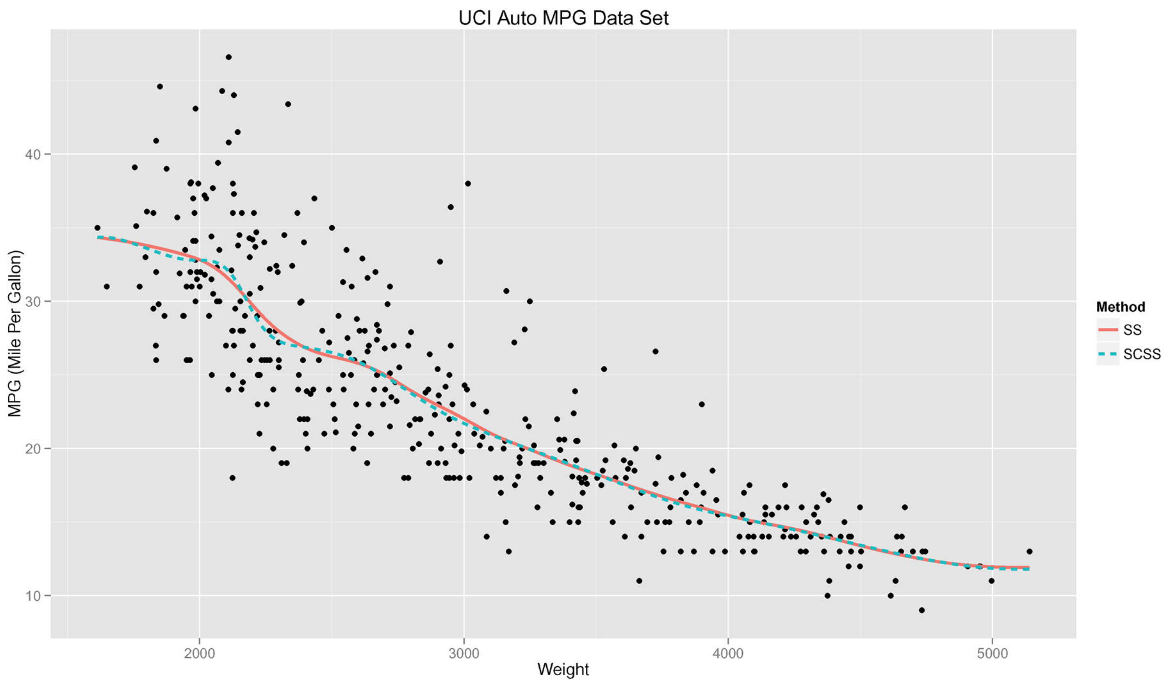

Figure 3. Comparison between unconstrained (SS) and monotone constrained (SCSS) smoothing spline for the Auto MPG Data. The response is *mpg*, modelled as a function of *weight*.

Next we consider the vehicle's volume (*displacement*) to be the predictor variable instead of *weight*. In general, one would expect a negative correlation between *mpg* and *displacement*. Figure 4 reveals the potential problem when prior knowledge on the function shape is not incorporated. The wiggly function estimated by SS contradicts the expectation of a monotone decreasing relationship between *mpg* and *displacement*. While the proposed SCSS-monotone capture the pattern of data quite well.

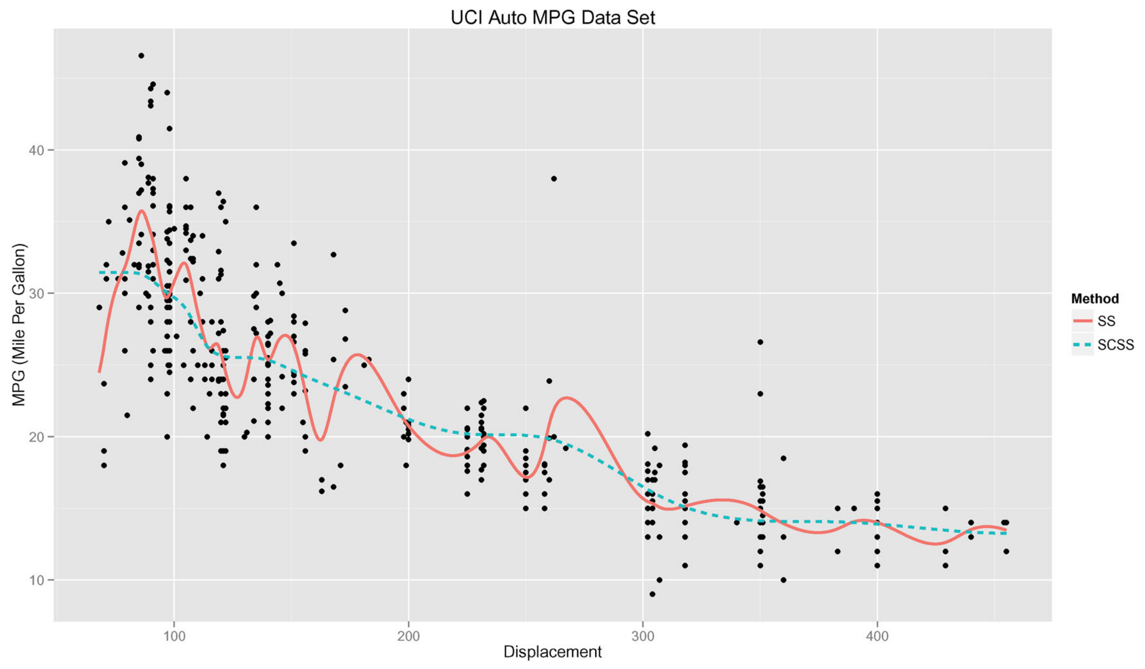

Figure 4. Comparison between unconstrained (SS) and monotone constrained (SCSS) smoothing spline for the Auto MPG Data. The response is *mpg*, modelled as a function of *displacement*.

In a short summary, when one has prior shape information about the function to be estimated, it would be better to incorporate it into the estimation process. The proposed shape constraint smoothing spline can effectively help avoid unexpected results from non-parametric method.

7. Discussion

In this work, we proposed to impose the exact shape constraint on the smoothing spline, and enable efficient estimation. Compared to other methods also based on the fundamental assumption that the resulted function is a NCS with knots at all data points, the proposed SCSS method guarantees constraints to be followed exactly and also allows mixed and/or non-global constraints.

The technique developed in the SSCS method can be extended for the additive model, partially linear regression model, the non-parametric model with covaraites, etc. The theoretical investigation of the asymptotic convergence of SSCS can be quite challenging due to the exact (i.e. over an interval) constraint. Some theoretical results are available for functional ANOVA using spline with shape constraint at given finite locations (Dai & Chien, 2017). However, such results can not be easily extended to the smoothing spline with exact constraint. Another future study is to formulate necessary and sufficient conditions for function bound constraint and log-convexity constraint. In addition, efficient optimisation methods that take advantage of a quadratic program with non-linear constraints could be useful for further enhance our proposed method.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by National Science Foundation [1634867].

Notes on contributors

Vincent Chan obtained his Ph.D. from the department of statistics at University of Wisconsin-Madison. His research interests include single index model and regularization.

Kam-Wah Tsui is a professor in the department of statistics at University of Wisconsin-Madison. His research interests include Bayesian analysis, decision theory, survey sampling, and statistical inference.

Yanran Wei is a Ph.D. student in the department of statistics at Virginia Tech. Her research interests include Big Data analytics, and statistical modeling in financial application.

Zhiyang Zhang is a faculty in the department of statistics at Virginia Tech. Her research interests include modeling complex data, and statistical modeling in chemistry applications.

Xinwei Deng is an associate professor in the department of statistics at Virginia Tech. His research interests include machine learning, design of experiment, and interface between machine learning and experimental design.

ORCID

Yanran Wei http://orcid.org/0000-0002-6745-7326
Xinwei Deng http://orcid.org/0000-0002-1560-2405

References

Brezger, A., & Steiner, W. J. (2003). *Monotonic regression based on Bayesian P-splines: An application to estimating price response functions from store-level scanner data*. Tech.

rep., Discussion paper//Sonderforschungsbereich 386 der Ludwig-Maximilians-Universität München.

Curry, H. B., & Schoenberg, I. J. (1966). On Pólya frequency functions IV: The fundamental spline functions and their limits. *Journal D'analyse Mathématique*, *17*, 71–107.

Dai, X., & Chien, P. (2017). *Minimax optimal rates of estimation in functional anova models with derivatives*. arXiv:1706.00850.

Delecroix, M., & Thomas-Agnan, C. (2000). Spline and Kernel regression under shape restrictions. In *Smoothing and Regression: Approaches, Computation, and Application* (pp. 109–133).

Dierckx, I. P (1980). Algorithm/algorithmus 42 an algorithm for cubic spline fitting with convexity constraints. *Computing*, *24*, 349–371.

Ducharme, G. R., & Fontez, B. (2004). A smooth test of goodness-of-fit for growth curves and monotonic nonlinear regression models. *Biometrics*, *60*, 977–986.

Fan, J., & Gijbels, I. (1996). *Local polynomial modelling and its applications: Monographs on statistics and applied probability*. New York: CRC Press.

Green, P. J. (1987). Penalized likelihood for general semiparametric regression models. *International Statistical Review / Revue Internationale de Statistique*, *55*(3), 245.

Green, P. J., & Silverman, B. W. (1993). *Nonparametric regression and generalized linear models: A roughness penalty approach*. New York: CRC Press.

Hastie, T., Tibshirani, R., & Friedman, J, *The elements of statistical learning*, Springer series in statistics Springer (Vol. 1). Berlin, 2001.

He, X., & Shi, P. (1998). Monotone B-spline smoothing. *Journal of the American Statistical Association*, *93*, 643–650.

Kelly, C., & Rice, J. (1990). Monotone smoothing with application to dose-response curves and the assessment of synergism. *Biometrics*, *46*(4), 1071–1085.

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

Lichman, M. (2013), *UCI machine learning repository*.

Mammen, E., & Thomas-Agnan, C. (1999). Smoothing splines and shape restrictions. *Scandinavian Journal of Statistics*, *26*, 239–252.

Matzkin, R. L. (1991). Semiparametric estimation of monotone and concave utility functions for polychotomous choice models. *Econometrica: Journal of the Econometric Society*, *59*,1315–1327.

Meyer, M. C. (2008). Inference using shape-restricted regression splines. *The Annals of Applied Statistics*, *2*, 1013–1033.

Meyer, M. C. (2012). Constrained penalized splines. *Canadian Journal of Statistics*, *40*, 190–206.

Meyer, M. C. (2018). Constrained partial linear regression splines. *Statistica Sinica*, *28*, 277–292.

Nagahara, M., & Martin, C. F. (2013). Monotone smoothing splines using general linear systems. *Asian Journal of Control*, *15*, 461–468.

Ramsay, J. O. (1988). Monotone regression splines in action. *Statistical Science*, *3*,425–441.

Shor, N. Z., & Zhurbenko, N. (1971). The minimization method using space dilatation in direction of difference of two sequential gradients. *Kibernetika*, *3*, 51–59.

Turlach, B. A. (2005). Shape constrained smoothing using smoothing splines. *Computational Statistics*, *20*, 81–104.

Utreras, F. I. (1985). Smoothing noisy data under monotonicity constraints existence, characterization and convergence rates. *Numerische Mathematik*, *47*, 611–625.

Villalobos, M., & Wahba, G. (1987). Inequality-constrained multivariate smoothing splines with application to the estimation of posterior probabilities. *Journal of the American Statistical Association*, *82*, 239–248.

Wahba, G. (1990). *Spline models for observational data*. Philadelphia, Pennsylvania: Siam.

Wand, M., & Jones, M.. (1995), *Kernel smoothing*. Vol. 60 of Monographs on statistics and applied probability.

Wand, M. P., & Ormerod, J. T. (2008). On Semiparametric regression with O'sullivan penalized splines. *Australian & New Zealand Journal of Statistics*, *50*(2), 179–198.

Wang, X., & Li, F. (2008). Isotonic smoothing spline regression. *Journal of Computational and Graphical Statistics*, *17*, 21–37.

Zeng, L., Deng, X., & Yang, J. (2016). Constrained hierarchical modeling of degradation data in tissue-engineered scaffold fabrication. *IIE Transactions*, *48*, 16–33.

Zhang, D., Lin, X., Raz, J., & Sowers, M. (1998). Semiparametric stochastic mixed models for longitudinal data. *Journal of the American Statistical Association*, *93*(442), 710.

Appendices

Appendix 1

We provide some preliminary materials about natural cubic spline (NCS) and smoothing spline. Readers of interest can refer to Wahba (1990) and Green and Silverman (1993) for details.

A.1 Value-second derivative representation

The value-second derivative representation allows specification of a NCS simply by its value and second derivative at each knots. This representation provides a link between the entire NCS $f(x)$ and $(x_i, f(x_i))$ for $i = 1, \dots, n$. Let us define

$$g_i = f(x_i) \quad \text{and} \quad \gamma_i = f''(x_i) \quad \text{for } i = 1, \dots, n.$$

Also, let vector $\boldsymbol{g} = (g_1, \dots, g_n)^T$ and $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_n)^T$. Note that due to the natural boundary conditions of a NCS, $\gamma_1 = \gamma_n = 0$. In addition, construct $n \times (n-2)$ matrix $\boldsymbol{Q}$ and $(n-2) \times (n-2)$ matrix $\boldsymbol{R}$ as follows:

$$\boldsymbol{Q} = \begin{pmatrix} h_1^{-1} & & \\ -h_1^{-1} - h_2^{-1} & \ddots & 0 \\ h_2^{-1} & \ddots & \\ & \ddots & h_{n-2}^{-1} \\ 0 & & -h_{n-2}^{1} - h_{n-1}^{-1} \\ & & h_{n-1}^{-1} \end{pmatrix}, \tag{A1}$$

$$\boldsymbol{R} = \begin{pmatrix} \frac{1}{3}(h_1 + h_2) & \frac{1}{6}h_2 & & 0 \\ \frac{1}{6}h_2 & \ddots & \ddots & \\ & \ddots & \ddots & \frac{1}{6}h_{n-2} \\ 0 & & \frac{1}{6}h_{n-2} & \frac{1}{3}(h_{n-2} + h_{n-1}) \end{pmatrix}, \tag{A2}$$

where $h_i = x_{i+1} - x_i$, for$i = 1, \dots, n-1$. By construction, matrix $\boldsymbol{R}$ is strictly positive-definite.

Lemma A.1 (Theorem 2.1 in Green and Silverman (1993)): *The vectors* $\boldsymbol{g}$ *and* $\boldsymbol{\gamma}$ *specify a natural cubic spline* f *if and only if the condition*

$$\boldsymbol{Q}^T\boldsymbol{g} = \boldsymbol{R}\boldsymbol{\gamma}, \tag{A3}$$

is satified. If (A3) is satified then the roughness penalty will satisfy

$$\int [f''(t)]^2 \mathrm{d}t = \boldsymbol{\gamma}^T \boldsymbol{R} \boldsymbol{\gamma} = \boldsymbol{g}^T \boldsymbol{K} \boldsymbol{g}, \qquad \text{(A4)}$$

where $\boldsymbol{K} = \boldsymbol{Q}\boldsymbol{R}^{-1}\boldsymbol{Q}^T$.

This value-second derivative representation provides a formula to recover the entire NCS with x_i and $f(x_i)$ for $i = 1, \ldots, n$.

A.2 Linear mixed model representation

The linear mixed model representation of an NCS allows us to express $f(x_1), \ldots, f(x_n)$ as a linear combination of a specific basis functions. Let function f be an NCS on interval $[a, b]$. Denote $\mathcal{L}^2[a, b]$ the space of square integrable functions on interval $[a, b]$. Let $\mathcal{H} = \{f : f, f'$ are absolutely continuous, $f'' \in \mathcal{L}^2[a, b]\}$ be a second-order Sobolev space of the NCS functions. Under the following definition of norm

$$||f||^2 = \left[\int_a^b f(x)\,\mathrm{d}x\right]^2 + \left[\int_a^b f'(x)\,\mathrm{d}x\right]^2 + \int_a^b [f''(x)]^2\,\mathrm{d}x.$$

Wahba (1990) shows that $\mathcal{H}$ is a reproducing kernel Hilbert space that can be decomposed into a direct sum of three orthogonal subspaces:

$$\mathcal{H} = \{1\} \oplus \mathcal{H}_0 \oplus \mathcal{H}_1,$$

where $\{1\}$ is the mean subspace, $\mathcal{H}_0 = \{f : f''(x) = 0\}$ is the linear contrast subspace and $\mathcal{H}_1 = \{f : \int_a^b f(x)dx = 0, \int_a^b f'(x)dx = 0, \int_a^b f''(x)dx \in \mathcal{L}[a, b]\}$ is the non-linear subspace. This decomposition means that any NCS function $f \in \mathcal{H}$ can be uniquely decomposed into a sum of a constant part, a linear part and a non-linear part as follows:

$$f(x) = \theta_0 + x\theta_1 + f_1(x), \qquad \text{(A5)}$$

for some functions $f_1 \in \mathcal{H}_1$.

Knowing that the solution is necessarily a NCS with knots at $x_1, \ldots, x_n$, one particular form of Equation (A5) is given by the linear mixed model representation (Green 1987; Zhang et al. 1998) as follows:

$$\boldsymbol{g} = \mathbf{1}_n\theta_0 + \boldsymbol{x}\theta_1 + \boldsymbol{A}\boldsymbol{\beta}, \qquad \text{(A6)}$$

where $\boldsymbol{A} = \boldsymbol{Q}(\boldsymbol{Q}^T\boldsymbol{Q})^{-1}\boldsymbol{R}^{1/2}$ is a $n \times (n-2)$ matrix, $\mathbf{1}_n$ is a length-n vector of ones and $\boldsymbol{x} = (x_1, \ldots, x_n)^T$.

Appendix 2

The linear mixed model representation is used for efficient computation of NCS. Meanwhile, the piecewise polynomial representation is used for formulating shape constraint on NCS for the same problem. The connection between linear mixed model representation and piecewise polynomial representation are stated as follows.

A.3 Specifying the NCS function f from x and g

Given $\boldsymbol{x}$, matrice $\boldsymbol{Q}$ and $\boldsymbol{R}$ can be constructed as shown in Appendix A.2. The second derivative vector $\boldsymbol{\gamma}$, can be obtained by Theorem A.1 as follows,

$$\boldsymbol{\gamma} = \boldsymbol{R}^{-1}\boldsymbol{Q}^T\boldsymbol{g}, \qquad \text{(A7)}$$

since $\boldsymbol{R}$ is of full rank by construction. From Section 2.4.1 in Green and Silverman (1993), the derivate of $f(.)$ at knot x_1 and x_n are

$$g_1' = \frac{g_2 - g_1}{x_2 - x_1} - \frac{1}{6}(x_2 - x_1)\gamma_2$$
$$g_n' = \frac{g_n - g_{n-1}}{x_n - x_{n-1}} + \frac{1}{6}(x_n - x_{n-1})\gamma_{n-1},$$

respectively. Finally, with $h_i = x_{i+1} - x_i$, the following formula summarised from Section 2.4.2 in Green and Silverman (1993) gives the entire NCS function f:

$$f(t) = \begin{cases} g_1 - (x_1 - t)g_1', & \text{if } t \le x_1, \\ \frac{(t-x_i)g_{i+1} + (x_{i+1}-t)g_i}{h_i} \\ \quad - \frac{(t-x_i)(x_{i+1}-t)}{6} \\ \quad \left\{\left(1 + \frac{t-x_i}{h_i}\right)\gamma_{i+1} \right. \\ \quad \left. + \left(1 + \frac{x_{i+1}-t}{h_i}\right)\gamma_i\right\}, & \text{if} x_i < t < x_{i+1} \\ & \quad \text{for } i \in (1, \ldots, n-1), \\ g_n + (t - x_n)g_n', & \text{if } t > x_n. \end{cases}$$

Hence the NCS function f is fully specified. That is, we can reconstruct NCS function f from $\boldsymbol{x}$ and $\boldsymbol{g}$.

A.4 Specifying the piecewise polynomial representation given f

The resulting function f can be used to estimate all coefficients a_i, b_i, c_i and d_i under the piecewise polynomial representation of f. The steps are as follows:

(1) Given $\boldsymbol{x}$ and the function of f, get $\boldsymbol{\gamma}$ from (A7).
(2) Calculate $(f'(x_1), \ldots, f'(x_n))^T$ by Equation (2.20) and (2.21) in Green and Silverman (1993) as:

$$f'(x_i) = \frac{g_{i+1} - g_i}{x_{i+1} - x_i} - \frac{(x_{i+1} - x_i)(2\gamma_i + \gamma_{i+1})}{6}, i = 1, \ldots, n-1;$$
$$f'(x_n) = \frac{g_n - g_{n-1}}{x_n - x_{n-1}} + \frac{\gamma_{n-1}(x_n - x_{n-1})}{6}.$$

(3) Obtain $c_0 = f_0'(x_1)$ and $c_n = f_n'(x_n)$ based on the piecewise polynomial representation.
(4) Using the fact $f''(x_{i+1}) = 6a_ix_{i+1} + 2b_i$ and $f''(x_i) = 6a_ix_i + 2b_i$, we get $a_i = \frac{1}{6}((\gamma_{i+1} - \gamma_i)/(x_{i+1} - x_i))$ for $i = 1, \ldots, n-1$.
(5) With a_i's from previous step, obtain $b_i = \frac{1}{2}(\gamma_i - 6a_ix_i)$ for $i = 1, \ldots, n-1$.
(6) From previous steps with a_i, b_i and $f'(x_i)$, obtain $c_i = f'(x_i) - 3a_ix_i^2 - 2b_ix_i$ for $i = 1, \ldots, n-1$.
(7) Obtain $d_n = f(x_n) - c_nx_n$, $d_0 = f(x_1) - c_0x_1$, and $d_i = f(x_i) - a_ix_i^3 - b_ix_i^2 - c_ix_i$ for $i = 1, \ldots, n-1$.

Appendix 3

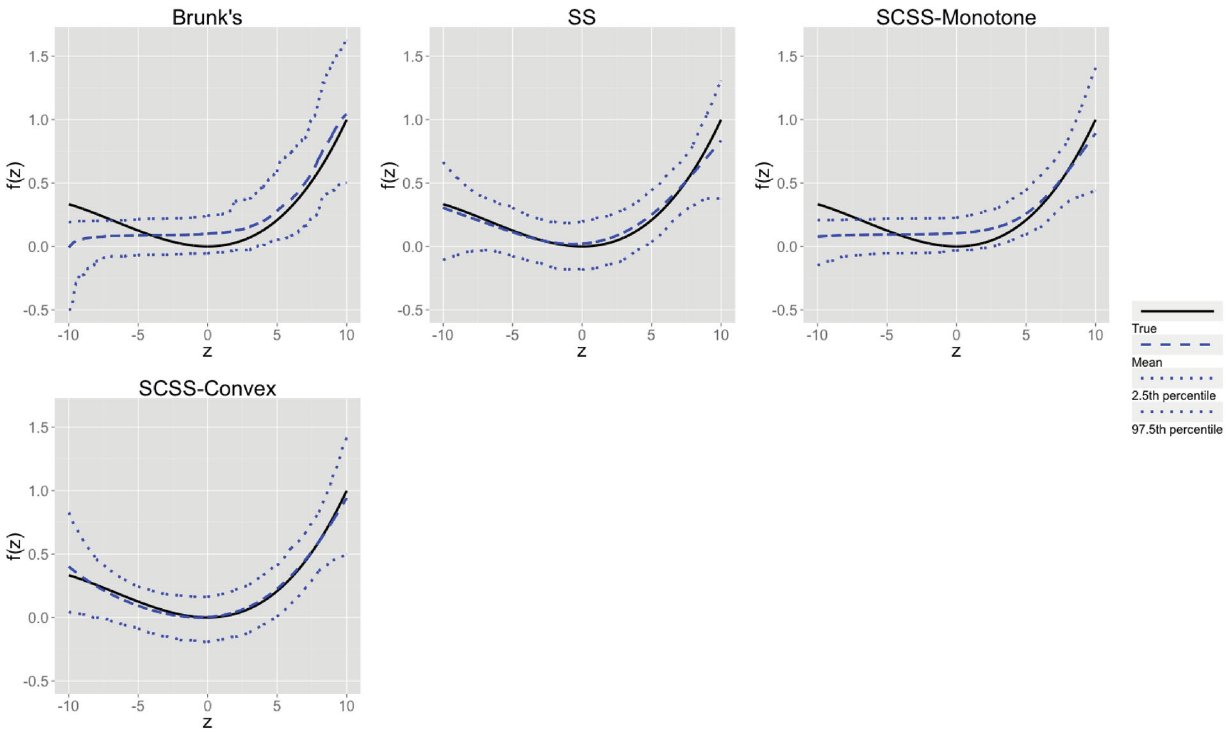

Figure A1. The estimator percentiles (2.5% and 97.5%) of SCSS for function $f(x) = (20x^2 + x^3)/3000$ in Example 4 under normal error.

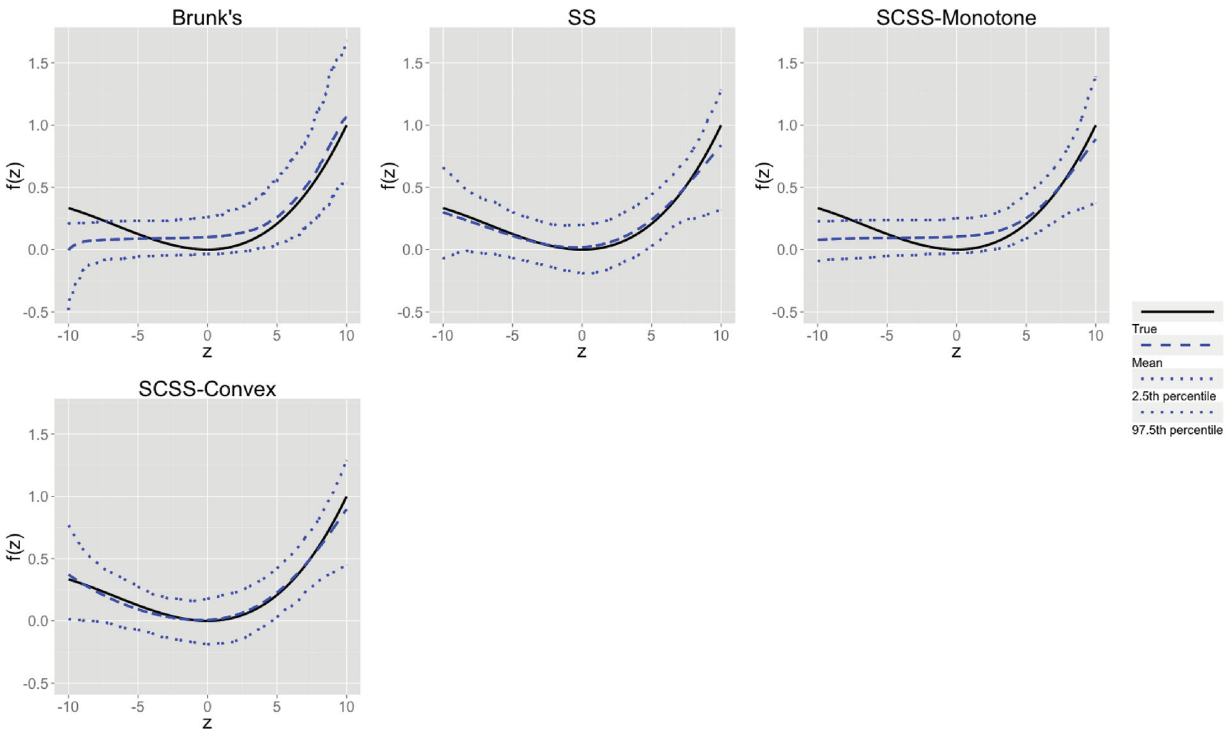

Figure A2. The estimator percentiles (2.5% and 97.5%) of SCSS for function $f(x) = (20x^2 + x^3)/3000$ in Example 4 under *t* error.

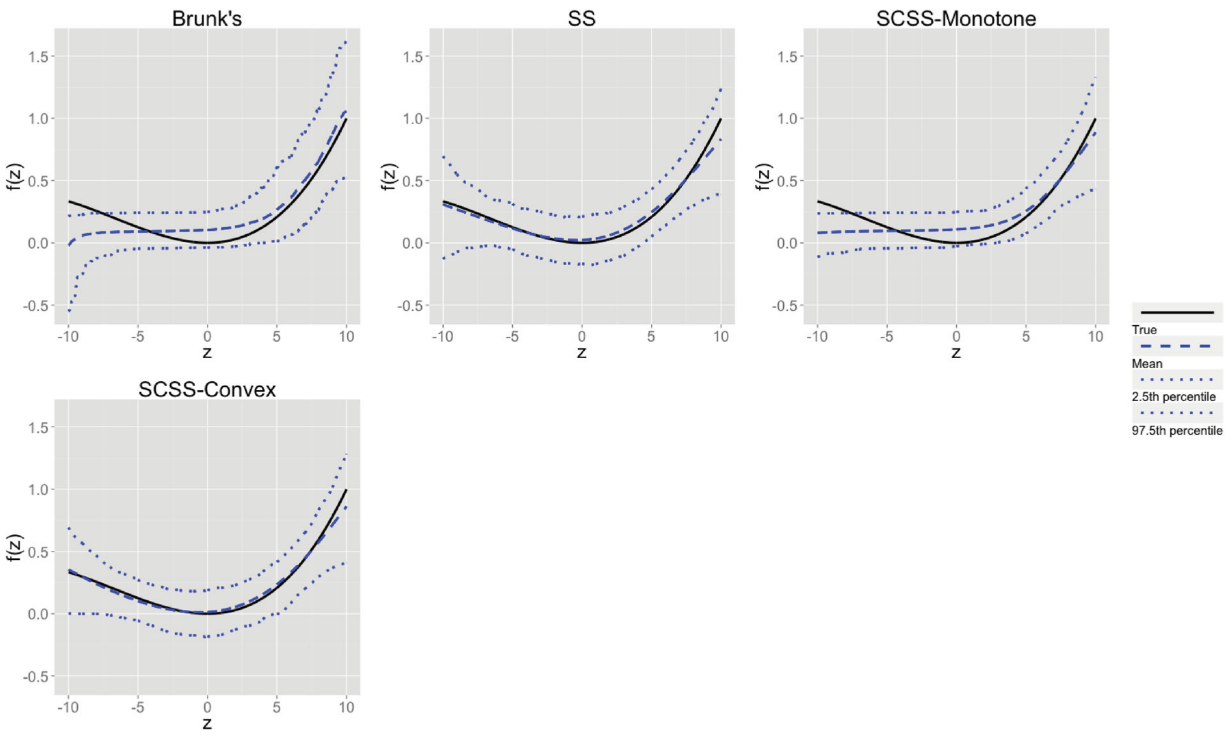

Figure A3. The estimator percentiles (2.5% and 97.5%) of SCSS for function $f(x) = (20x^2 + x^3)/3000$ in Example 4 under beta error.

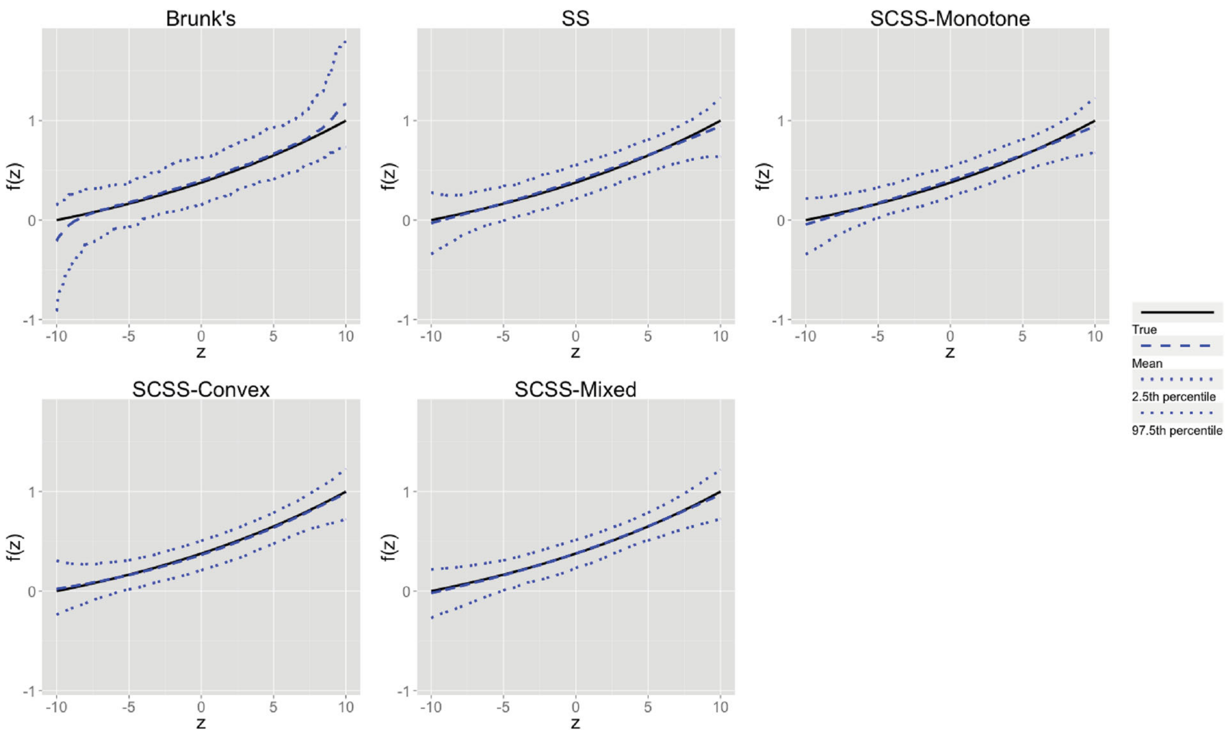

Figure A4. The estimator percentiles (2.5% and 97.5%) of SCSS for function $f(x) = (e^{x/20} - e^{-10/20})/(e^{10/20} - e^{-10/20})$ in Example 5 under normal error.

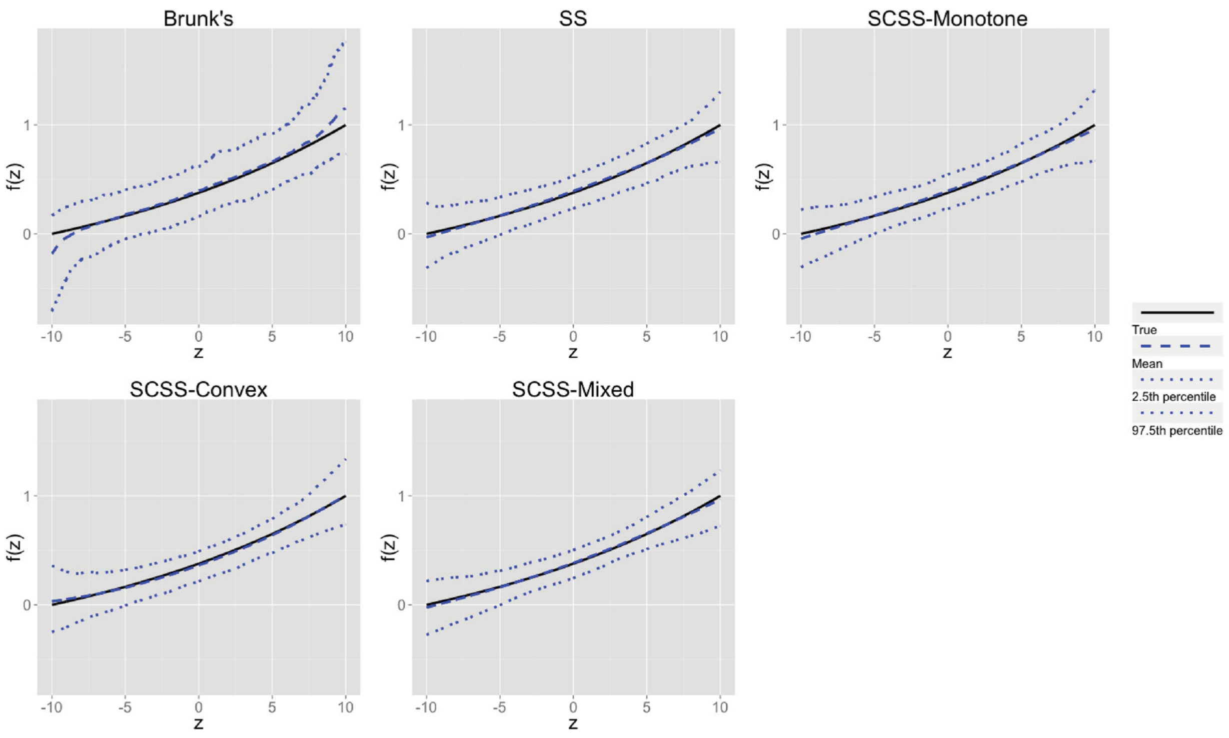

Figure A5. The estimator percentiles (2.5% and 97.5%) of SCSS for function $f(x) = (e^{x/20} - e^{-10/20})/(e^{10/20} - e^{-10/20})$ in Example 5 under t error.

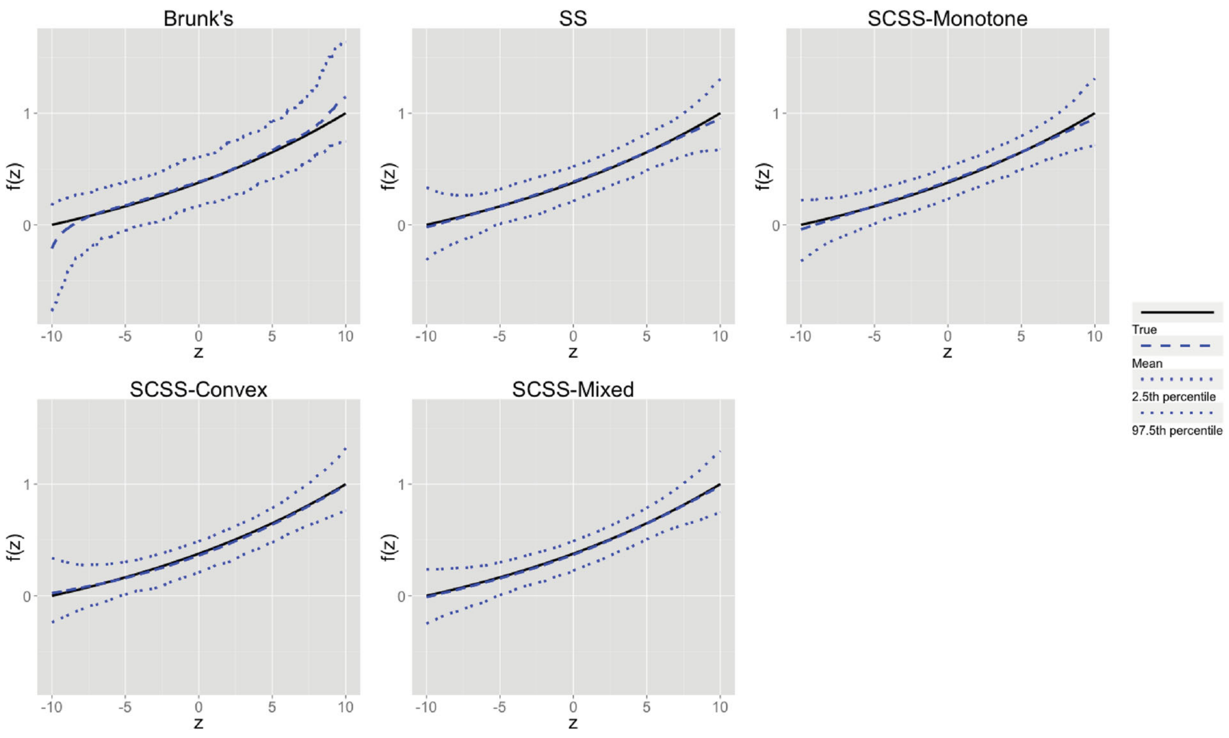

Figure A6. The estimator percentiles (2.5% and 97.5%) of SCSS for function $f(x) = (e^{x/20} - e^{-10/20})/(e^{10/20} - e^{-10/20})$ in Example 5 under beta error.

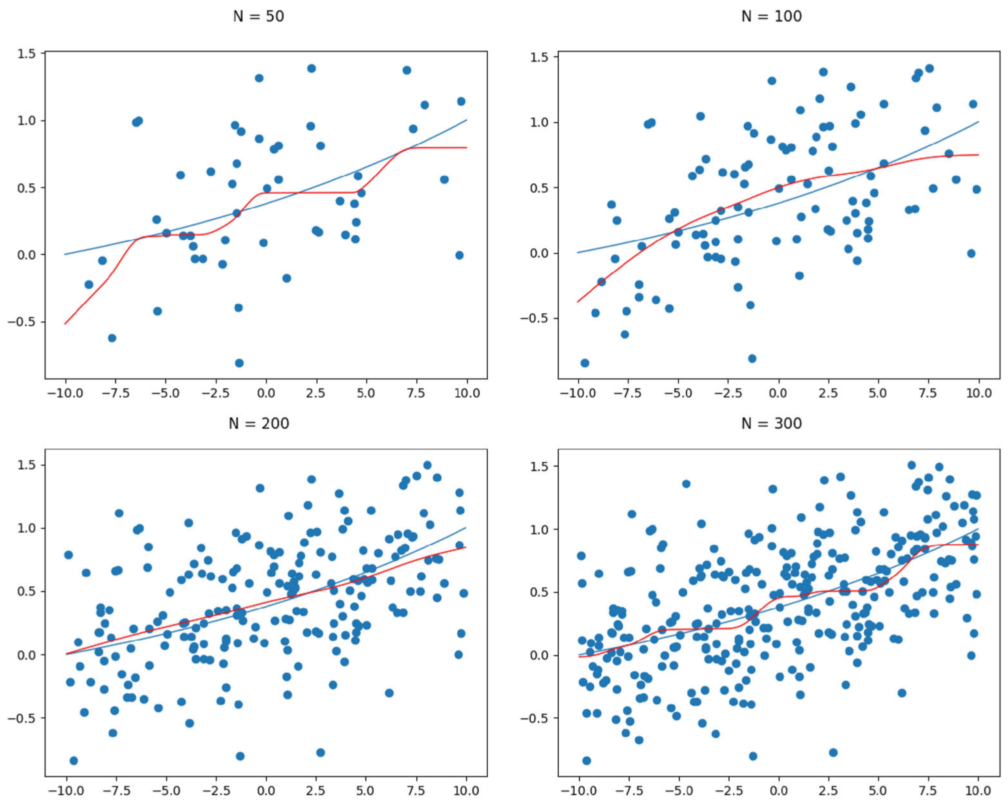

Figure A7. SCSS convergence for function $f(x) = (e^{x/20} - e^{-10/20})/(e^{10/20} - e^{-10/20})$ in Example 5 under normal error and monotone constraint.

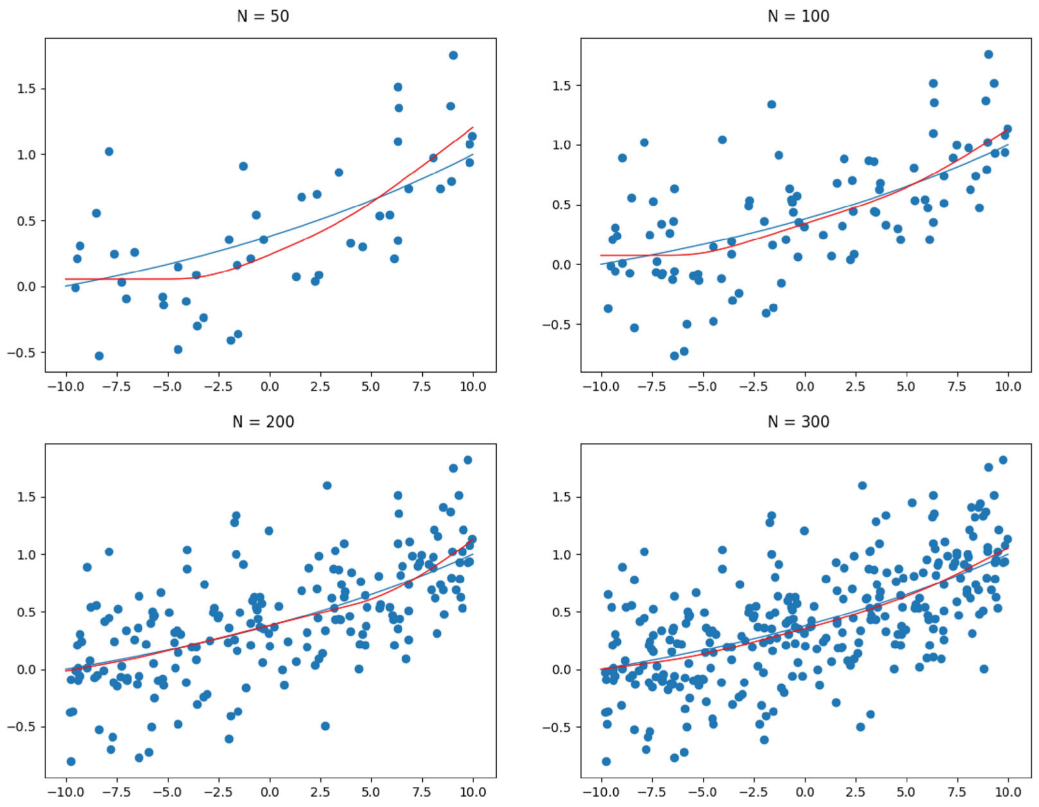

Figure A8. SCSS convergence for function $f(x) = (e^{x/20} - e^{-10/20})/(e^{10/20} - e^{-10/20})$ in Example 5 under normal error and mixed constraint.