Fast and flexible methods for monotone polynomial fitting

K. Murray, S. Müller & B. A. Turlach

JOURNAL OF STATISTICAL COMPUTATION AND SIMULATION, 2016
http://dx.doi.org/10.1080/00949655.2016.1139582

Fast and flexible methods for monotone polynomial fitting

K. Murray[a,b,c], S. Müller[b] and B. A. Turlach[a,c]

[a]Centre for Applied Statistics (M019), University of Western Australia, Crawley, WA, Australia; [b]School of Mathematics and Statistics, University of Sydney, NSW, Australia; [c]School of Mathematics and Statistics (M019), University of Western Australia, Crawley, WA, Australia

ABSTRACT

We investigate an isotonic parameterization for monotone polynomials previously unconsidered in the statistical literature. We show that this parameterization is more flexible than its alternatives through enabling polynomials to be constrained to be monotone over either a compact interval or a semi-compact interval of the form $[a, \infty)$, in addition to over the whole real line. Furthermore, algorithms based on our new parameterization estimate the fitted monotone polynomials much faster than algorithms based on previous isotonic parameterizations which in turn makes the use of standard bootstrap methodology feasible. We investigate the use of the bootstrap under monotonicity constraints to obtain confidence bands for the fitted curves and show that an adjustment by using either the '*m* out of *n*' bootstrap or a post hoc symmetrization of the confidence bands is necessary to achieve more uniform coverage probabilities. We illustrate our new methodology with two real world examples which demonstrate not only the need for such techniques, but how restricting the monotonicity constraints to be over either a compact or semi-compact interval allows the fitting of even degree monotone polynomials. We also describe methods for using the '*m* out of *n*' bootstrap to select the degree of the fitted monotone polynomial. All algorithms discussed in this paper are available in the R package `MonoPoly` (version 0.3-6 or later).

ARTICLE HISTORY
Received 2 September 2015
Accepted 5 January 2016

KEYWORDS
Monotone polynomials; monotone regression; semi-compact interval; confidence bands; shape constraints; bootstrap

1. Introduction

This article presents an isotonic parameterization of polynomials that allows the fitting of polynomials that can be constrained to be monotone over a compact interval, semi-compact interval, or over the entire real line. The proposed sums-of-squares parameterization is not only key to generalizing the fitting of monotone polynomials over (semi-)compact regions but also allows the fitting of monotone polynomials of even degree over such regions. We examine the impact of the sums-of-squares parameterization and its effectiveness and show that this new approach leads to algorithms for fitting monotone polynomials that are a noticeable improvement over the algorithms available until now.

The outline of this paper is as follows: Section 1 continues with the introduction and highlights the context, need and applications of using the sums-of-squares parameterization for estimating monotone polynomials. Section 2 describes this new methodology proposed for fitting monotone polynomials to data. This alternative parameterization enables the fitting of monotone polynomials, not only over the whole real line but also over compact or semi-compact intervals, thus allowing polynomials of even degree, that are monotone over such regions, to be considered. Section 3 describes

CONTACT K. Murray kevin.murray@uwa.edu.au

through numerical examples bootstrap methodology for estimating confidence bands for the monotone fits and makes two suggestions for the tuning of these bands to enable desired and/or uniform coverage probabilities. In this context we also discuss some of the asymptotic properties of monotone polynomials and prediction intervals. Section 4 illustrates the proposed methodologies on two real world examples. A brief discussion and conclusions are given in Section 5.

In many regression settings there is a need to place constraints on the shape of the regression function, as it is known, for example, from some underlying neurological, psychological or physical theory, that the regression curve is monotone. However, in applications that benefit from the implementation of monotonicity constraints, it is sometimes not necessary to constrain the response curve over the whole real line, rather the constraints may only be necessary over a compact or semi-compact interval. One prime example of this is human growth curve modelling (see, e.g. [1,2]) in which it is desirable to have the height of an individual to be monotone increasing over time, but for obvious reasons such a constraint is not needed at negative time points and outside the range of data. Intuitively, this example extends to any monotonic growth or decay situation that is restricted by a starting point (i.e. time zero or equivalent). For example, studying the vocabulary (measured in number of words) of children which increases monotonically from birth [3]; survival analyses, in which the survival probability can be modelled as a monotone decreasing function from a fixed time point. Numerous other examples arise from physical applications in which such a relaxation of the monotonicity constraint may be sensible. For example, in forensic science, estimation of the age of a child from their dental records, a process that is frequently carried out by calculating a dental maturity score, which itself by definition is over a finite compact interval (see, e.g. [4]). In this instance, using a monotone curve to predict age would only require the curve to be monotone increasing (i.e. higher scores would predict higher ages) over the range of possible values of the maturity score (typically 0–100). Another example, from neuroscience, involves the modelling of stimulus amplitude measurements at each delay extension step as a monotone decreasing function.[5,6] That is with increasing delay, more and more action potentials escape from the collision protocol and, consequently, are not reflected by the stimulus amplitude anymore causing a lower response. Again in this instance there is no need for the monotonicity constraint over the whole real line. These latter two examples, the dental age data and the brain data, are discussed in more detail later in this paper but are amongst a long list of potential applications in which the underlying function needs only to be monotonic over a (semi-)compact region.

A researcher who needs to fit a monotone regression curve typically has to make the choice between a parametric nonlinear regression model,[7] in particular those models developed in the growth curve literature (see, e.g. [8,9]), or a shape constrained smoothing technique. Popular methods for the latter approach involve the use of either spline smoothing or kernel smoothing techniques. Incorporating shape constraints into spline smoothing has been well studied and, for reviews of the existing literature, we refer to the introductory sections of Turlach [10], Hazelton and Turlach [11], and Meyer [12,13]. By way of contrast, kernel smoothing techniques for (general) shape constraints [14,15] are somewhat less often used, but there is some notable work on monotone kernel smoothers.[16–20]

Unfortunately approaches to monotone regression that use parametric nonlinear models may suffer from the problem that many parametric monotone regression models depend only on a few parameters and, hence, such models may miss important features such as the number and location of roots, extrema and inflection points of (higher order) derivatives of a regression function. On the other hand, most approaches that impose monotonicity on non-parametric smoothing techniques can be problematic due to the estimated regression curve having flat stretches, and monotone regression 'has been criticized because practitioners do not believe in all those flat spots'.[19] Having to contend with estimated regression functions with many (spurious) flat spots is highly undesirable, especially in situations where the estimation of derivatives (and features thereof) is important, as these would translate to increased variability of such estimates. While previous approaches (see [11,19,21]) lead to estimated regression functions that are strictly monotone, the functional form of the estimated regression function does not lend itself readily to a subsequent analysis to aspects of its derivatives.

This suggests that there is a need for some methodology that occupies the middle ground between the two approaches discussed, that is, there is a need for models that are more flexible than parametric monotone regression curves but which lend themselves more easily to post-processing and further calculations than monotone non-parametric smoothing techniques. We suggest that monotone polynomials provide such a methodology and note that these polynomials also exhibit many useful properties, for example, that they are naturally strictly monotone and have easily identifiable derivatives.

Previous work has revisited the idea of using the monotone polynomials [22] and provided algorithms for the fitting of monotone polynomials to data based on established isotonic parameterizations [23] and a semi-infinite programming algorithm.[24] This paper discusses the sum-of-squares parameterization of monotone polynomials which provides at least two advantages. First, it allows, when fitting polynomials to data, to impose the monotonicity constraints only over compact or semi-compact regions, respectively. A relaxation to the monotonicity constraints for monotone polynomials not proposed or implemented by any other algorithm to date and to our knowledge.

Secondly, the speed with which monotone polynomials can be fitted using this parameterization makes it feasible to explore the use of bootstrap techniques (see, e.g. [25–30]) for statistical inference purposes. We propose, and investigate, the use of bootstrap methodology for model selection, and for estimation of confidence/prediction bands. Although there is a large body of literature on bootstrap methodology for estimating confidence bands in a non-parametric, or semi-parametric setting (see, for example, the summary provided in [31], and references therein), to date and to our knowledge there is a lack of available bootstrap methodology for our specific problem involving monotone polynomials. However, there are a couple of exceptions that address some of the issues arising when calculating such bands under shape constraints. In particular Hall and Horowitz [31] describe the construction of bootstrap confidence bands, taking into account that the standard bootstrap bias estimators suffer from relatively high frequency stochastic error. They combine this with a technique based on quantiles, leading to simple to construct confidence bands. Another exception is the construction of confidence bands for a given function with guaranteed coverage probability, under the assumption that the function is isotonic or convex.[32] Furthermore, a comparison of techniques for bootstrap confidence intervals for monotone non-parametric regression is given in [33]. Here, we consider through numerical examples the non-parametric and the residual bootstrap, examine their performance under monotonicity constraints, and provide some general advice when considering the calculation of confidence bands for monotone polynomials. We describe some of the issues associated with estimating confidence bands using monotone polynomials and discuss consistency. Furthermore, we illustrate our methodology with examples from Neuroscience and Forensic Science, utilizing standard bootstrap methods to select the ‘best’ degree of polynomial to fit.

2. Methodology

The usual parameterization for a polynomial regression function is

$$p(x) = p(x; \boldsymbol{\beta}) = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_q x^q, \tag{1}$$

where q is the degree of the polynomial and $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_q)^{\mathrm{T}}$, with the β_js being the regression parameters in the linear regression model

$$Y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_q x^q + \epsilon, \tag{2}$$

where ϵ is the random error term assumed to have mean zero and constant variance. To fit a polynomial regression curve to given data (x_i, y_i), $i = 1, \ldots, n$, using least squares, the residual sum of squares (RSS) is minimized, which under the assumption of Gaussian errors in Equation (2) achieves the same as maximizing the likelihood. That is, the fit is determined by minimizing

$$\mathrm{RSS}(\boldsymbol{\beta}) = \sum_{i=1}^{n} (y_i - p(x_i))^2. \tag{3}$$

Note that this objective function is strictly convex in $\boldsymbol{\beta}$ if the number of distinct x-values exceeds q.

However, this parameterization is not convenient to use if there are monotonicity constraints on the polynomial over a set $\mathcal{R} \subseteq \mathbb{R}$. Previously published work on monotone polynomials only considers the case $\mathcal{R} = (-\infty, \infty)$. In this paper we propose to use another isotonic parameterization which will allow us to specify, in addition to $(-\infty, \infty)$, more general regions $\mathcal{R}$, namely semi-compact intervals $[a, \infty)$ and compact intervals $[a, b]$ for finite $a, b \in \mathbb{R}$, on which the fitted polynomial satisfies a monotonicity constraint.

2.1. Isotonic parameterization

As an alternative approach to overcome the problem that Equation (3) cannot be easily minimized under monotonicity constraints when the parameterization (1) is used, we consider here parameterization of isotonic polynomials of the form

$$p(x) = \delta + \alpha \int_0^x \check{p}(u)\,\mathrm{d}u, \tag{4}$$

where $\check{p}(u)$ is required to be non-negative on $\mathcal{R}$. From Equation (4) it follows immediately that the first derivative of the polynomial $p'(x) = \alpha\check{p}(x)$, ensuring that $p(x)$ is monotone increasing or monotone decreasing on $\mathcal{R}$ depending on whether α is positive or negative, respectively.

The isotonic parameterizations considered previously, based on established formulations,[23] wrote $\check{p}(x)$ as a product of quadratic polynomials, where each of these quadratics had either conjugate complex roots or real roots of even multiplicity. Finding the parameter vector that minimizes RSS is typically slow when using these parameterizations, although they allow the fitting of monotone polynomials in some situations in which the semi-infinite programming approach [24] failed (see [22], for a more detailed discussion). However, without awkward case distinctions, these approaches cannot be readily extended to more general forms of $\mathcal{R}$.

In this paper we propose another isotonic parameterization for $\check{p}(u)$ which is based on the following proposition.

Proposition 1: *A polynomial $\check{p}(x)$ of degree q is non-negative*

(1) *on $\mathcal{R} = (-\infty, \infty)$ if and only if $q = 2K$ and it can be written as the sum of two squared polynomials*

$$\check{p}(x) = p_1(x)^2 + p_2(x)^2, \quad \forall x \in \mathbb{R}, \tag{5}$$

where $p_1(x)$ and $p_2(x)$ are polynomials whose degrees are at most K.

(2) *on $\mathcal{R} = [a, \infty)$ if and only if it can be written as*

$$\check{p}(x) = p_1(x)^2 + (x - a)p_2(x)^2, \quad \forall x \in \mathbb{R}, \tag{6}$$

where, if $q = 2K$, $p_1(x)$ and $p_2(x)$ are polynomials whose degrees are at most K and $K - 1$, respectively, and, if $q = 2K + 1$, both degrees are at most K.

(3) *on $\mathcal{R} = [a, b]$ if and only if it can be written as*

(a) *if $q = 2K$:*

$$\check{p}(x) = p_1(x)^2 + (x - a)(b - x)p_2(x)^2, \quad \forall x \in \mathbb{R}, \tag{7}$$

where $p_1(x)$ and $p_2(x)$ are polynomials whose degrees are at most K and $K - 1$, respectively.

(b) *if* $q = 2K + 1$:

$$\check{p}(x) = (x - a)p_1(x)^2 + (b - x)p_2(x)^2, \quad \forall x \in \mathbb{R}, \tag{8}$$

where $p_1(x)$ *and* $p_2(x)$ *are polynomials with their degree at most* K.

Proposition 1 can be proved using the theory of Tchebycheff systems [34] or the theory of canonical moments.[35] A proof that does not utilize such deep mathematical theories can be found in [36].

We use Proposition 1, by fixing α in Equation (4) to be either -1 or 1, depending on whether the polynomial should be monotone decreasing or monotone increasing respectively over $\mathcal{R}$. We denote the coefficients of $p_1(x)$ and $p_2(x)$ in Equations (5)–(8) by $\boldsymbol{\beta}_1 = (\beta_{01}, \beta_{11}, \dots, \beta_{q_1 1})^{\mathrm{T}}$ and $\boldsymbol{\beta}_2 = (\beta_{02}, \beta_{12}, \dots, \beta_{q_2 2})^{\mathrm{T}}$, respectively, with $q_1, q_2 \in \{K-1, K\}$. This allows us to write the vector of parameters generically by $\boldsymbol{\theta} = (\delta, \boldsymbol{\beta}_1^{\mathrm{T}}, \boldsymbol{\beta}_2^{\mathrm{T}})^{\mathrm{T}}$. Finally, for the remainder of the paper, it will be clear from the context whether we regard the RSS as a function of $\boldsymbol{\beta}$ or as a function of $\boldsymbol{\theta}$.

2.2. Evaluating the objective function and its derivatives

To evaluate the RSS we first calculate the $\boldsymbol{\beta}$ which corresponds to the given $\boldsymbol{\theta}$. This allows an easy evaluation of $p(x)$ for arbitrary values of x, using for example the Horner scheme for numerical stability,[37] and to use Equation (3) for the calculation of the RSS.

We illustrate the necessary calculations for a polynomial that is monotone on $\mathcal{R} = (-\infty, \infty)$; other choices of $\mathcal{R}$ can be handled analogously. To calculate $\boldsymbol{\beta}$ for a given $\boldsymbol{\theta}$, we calculate the convolution of $\boldsymbol{\beta}_1$ with itself and the convolution of $\boldsymbol{\beta}_2$ with itself to obtain the coefficients of the polynomials $p_1(x)^2$ and $p_2(x)^2$, respectively. By adding the corresponding coefficients of these two polynomials, we obtain the coefficients $\boldsymbol{\gamma} = (\gamma_0, \dots, \gamma_{q-1})^{\mathrm{T}}$ of the polynomial

$$\check{p}(t) = \gamma_0 + \gamma_1 t + \cdots + \gamma_{q-1} t^{q-1}.$$

From $\boldsymbol{\gamma}$ we can readily calculate $\boldsymbol{\beta}$ as

$$\boldsymbol{\beta} = \left(\delta, \alpha\gamma_0, \alpha\frac{\gamma_1}{2}, \dots, \alpha\frac{\gamma_{q-1}}{q}\right)^{\mathrm{T}}.$$

To minimize RSS numerically, using a derivative-based optimization algorithm, requires first and second derivatives. These derivatives can easily be calculated in a similar manner to how the objective function is evaluated. Using Equation (3) we see

$$\frac{\partial}{\partial\boldsymbol{\theta}}\mathrm{RSS} = -2\sum_{i=1}^{n}(y - p(x_i))\frac{\partial}{\partial\boldsymbol{\theta}}p(x_i). \tag{9}$$

Clearly, $\partial p(x_i)/\partial\delta \equiv 1$. For other components of $\boldsymbol{\theta}$, say β_{jl}, for $l \in \{1, 2\}$ and some $j \in \{0, 1, \dots, q_l\}$, we find

$$\frac{\partial}{\partial\beta_{jl}}p(x) = \frac{\partial}{\partial\beta_{jl}}\alpha\int_0^x \check{p}(u)\,\mathrm{d}u = \alpha\int_0^x \frac{\partial}{\partial\beta_{jl}}\check{p}(u)\,\mathrm{d}u = \alpha\int_0^x 2p_l(u)u^j\,\mathrm{d}u \tag{10a}$$

$$= \alpha\int_0^x 2\left(\sum_{k=0}^{q_l}\beta_{kl}u^k\right)u^j\,\mathrm{d}u = 2\alpha\left(\sum_{k=0}^{q_l}\frac{\beta_{kl}x^{k+j+1}}{k+j+1}\right). \tag{10b}$$

The equations in (10) show that the coefficients of the polynomials appearing in Equation (9) are easily determined.

Furthermore, from Equation (9) it follows that the Hessian matrix of RSS is

$$\frac{\partial^2}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^{\mathrm{T}}}\mathrm{RSS}(\boldsymbol{\theta}) = 2\sum_{i=1}^{n}\left(\frac{\partial}{\partial\boldsymbol{\theta}}p(x_i)\right)\left(\frac{\partial}{\partial\boldsymbol{\theta}}p(x_i)\right)^{\mathrm{T}} - 2\sum_{i=1}^{n}(y - p(x_i))\frac{\partial^2}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^{\mathrm{T}}}p(x_i). \tag{11}$$

And from Equation (10) it follows that

$$\frac{\partial^2}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^{\mathrm{T}}}p(x) = \begin{pmatrix} 0 & \mathbf{0}^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} \\ \mathbf{0} & \mathbf{H} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{H} \end{pmatrix}, \tag{12}$$

where $\mathbf{H}$ is a $(q_l + 1) \times (q_l + 1)$ matrix with (j, k)th entry being $2\alpha/(j + k + 1)x^{j+k+1}$, with $j, k = 0, \ldots, q_l$.

2.3. Algorithms for optimizing the objective function

As shown in Section 2.2, the gradient and the Hessian of the RSS can be readily calculated, which motivates the use of a derivative-based optimization algorithm for minimizing the RSS. We found a Levenberg–Marquardt modification of the Newton–Raphson algorithm [38] proved to be effective for minimizing $\mathrm{RSS}(\boldsymbol{\theta})$. The iterative part of this algorithm has been described previously [22] and we discuss here only the initialization and the stopping criteria.

2.3.1. Starting values

To start the optimization routine, we fit a polynomial using a constant, linear and cubic term to those points (x_i, y_i) for which x_i falls into $\mathcal{R}$ if the monotone polynomial to be fitted is of odd degree, otherwise we use an initial polynomial fit up to and including the quadratic term.

From this initial fit starting values are determined. If the initial fit is monotone, one can calculate its representation using the appropriate isotonic prarameterization (5) – (7) or (8). Otherwise, it is possible to determine the isotonic parameterization of a polynomial whose coefficients equals the coefficient of the initial fit in absolute values. Detailed formulae for calculating these starting values are given in [39].

Finally, α is automatically chosen as either -1 or 1 based on the sign of the correlation of those points (x_i, y_i) for which x_i falls into $\mathcal{R}$. However, the implementation in our R package `MonoPoly` allows the user to override this choice and specify whether α should be -1 or 1.

2.3.2. Stopping criteria

We determine convergence by monitoring the gradient vector ∇RSS, with a suitable default stopping criteria being that the absolute value of each entry in ∇RSS is smaller than 10^{-5}.

2.4. Timing comparison with previous approaches

In our experience fitting a monotone polynomial to data using the sum-of-squares parameterization is markedly faster than using any of the isotonic parameterizations considered previously and is comparable with the semi-infinite programming approach of Hawkins [24]. The latter could be extended to allow fitting of polynomials that are monotone only over a specified compact or semi-compact region. However, we do not investigate this further here since previous work [22] noted that earlier isotonic parameterizations [23] allow the fitting of monotone polynomials in some situations in which the semi-infinite programming approach fails; this also being true for the sum-of-squares parameterization. Furthermore, it was noted that the semi-infinite programming approach might require the careful choice of numerical precision parameters to allow the various numerical algorithms used in its implementation to successfully work together.[22] We would expect these numerical precision

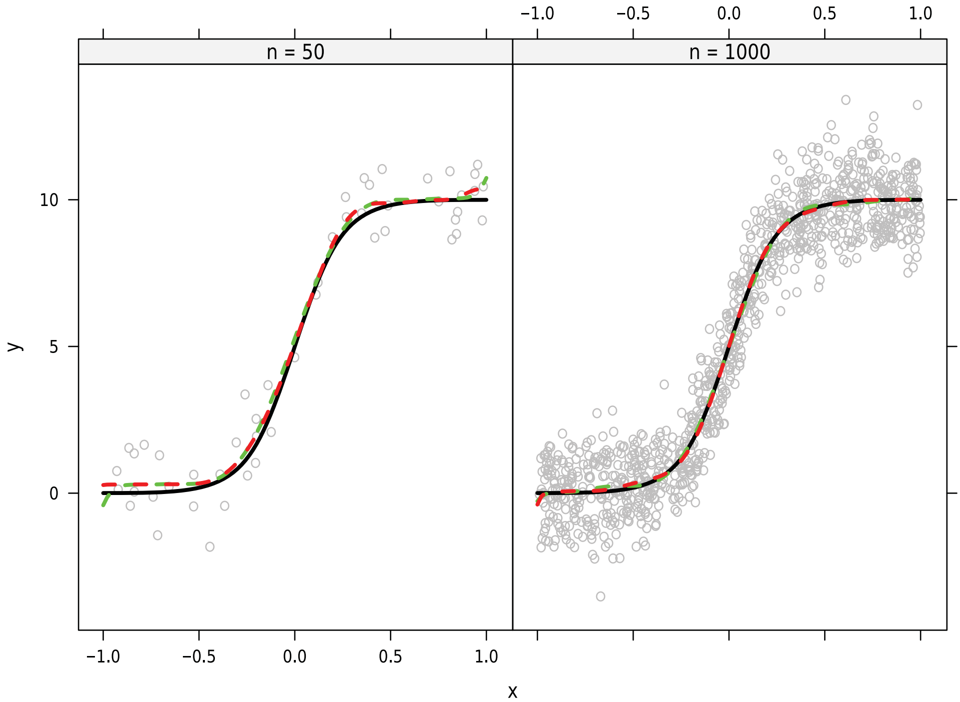

Figure 1. Sigmoidal curve (black) with $n = 50$ (left panel) and $n = 1000$ (right panel) observations generated from this regression curve. Monotone polynomial fit of polynomial of degree 9 (green dashed lines) and degree 15 (red dashed lines) are shown.

problems to increase when a polynomial is restricted to be monotone over a (semi-)compact interval and, a decision would have to be made on which side of the boundary a 'horizontal inflection point' [24] falls.

We report results from one of our timing experiments that we performed using a sigmoidal function as the true underlying regression function. This target function is selected as it provides a suitable target for monotone polynomials of increasing degree. In fact, it has been shown that monotone functions over compact intervals can be approximated uniformly to an arbitrary precision by monotone polynomials of sufficiently high degree q.[40] Figure 1 depicts the sigmoidal function used as a solid black line together with a typical data set of size $n = 50$ (left panel) and $n = 1000$, respectively. Fitted monotone polynomials of degrees $q = 9$ and $q = 15$, respectively, are overlaid in both panels and illustrate that monotone polynomials of degree 9 approximate the underlying regression curve fairly well.

In our timing experiment, we used sample sizes of $n = 50$, 100, 200, 250, 400, 500, 800, 1000, 1600 and 2500, respectively, and monotone polynomial of degrees 5, 7, . . . , 19 and 21. For each sample size, 10 replicates were generated, that is, we sampled x_i, $i = 1, \dots, n$, from a uniform distribution, and generated corresponding y_is by evaluating the sigmoidal regression function at the x_is and adding standard normally distributed noise. For each simulated data set the time needed to fit a monotone polynomial using the sum-of-squares parameterization, Hawkins' semi-indefinite programming approach [24] and the Elphinstone type isotonic parameterization recommended previously [22] were recorded and, for each method, averaged over the 10 replications. The resulting average times are shown in Figures 2 and 3. These figures show that the sum-of-squares parameterization is markedly faster than using the preferred Elphinstone type isotonic parameterization [23] and is comparable with the semi-infinite programming approach of Hawkins [24].

During our timing experiments, we also noticed that the sum-of-squares isotonic parameterization approach and Hawkins' semi-infinite programming [24] approach always resulted in the same

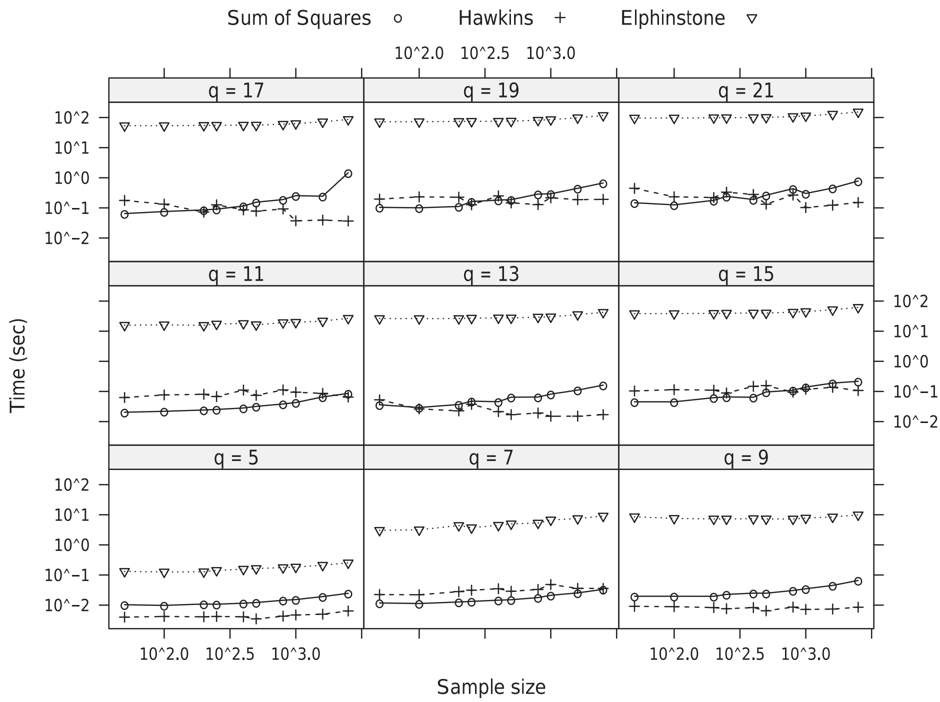

Figure 2. Monotone polynomials of degree $q = 5, 7, \ldots, 21$ were fitted to simulated data with various sample sizes from a sigmoidal curve (Figure 1). The plot shows for three approaches to fitting monotone polynomials the average time, over 10 replications, needed to fit the model against the sample size. Both axes are on a log scale.

fitted monotone polynomial, while the fitted monotone polynomial using the Elphinstone type isotonic parameterization [23] occasionally differed if the true regression function was a polynomial and somewhat more often with the sigmoidal target function. This suggests that fitting monotone polynomials using the sum-of-squares parameterization, instead of an Elphinstone type isotonic parameterization,[23] is not only beneficial in terms of run-time but also reduces the potential risk of the algorithm converging to a local minima.

2.5. Miscellaneous comments

Previous applications of fitting monotone polynomials to data involved fitting monotone polynomials of increasing degree.[23,41] The earlier suggestion was to use the fitted parameters from a monotone polynomial of degree $q = 2K - 1$ as starting values for fitting a monotone polynomial of degree $q = 2K + 1$, initializing the two new additional parameters to zero.[23] However, it was demonstrated that, with the earlier isotonic parameterizations,[23,41] this strategy is problematic if a gradient-based optimization algorithm is used for minimizing the objective function, as one of the new parameters would remain fixed at 0.[22]

However, we note that using the new sum of squares formulation, implementing this method of choosing starting values when iteratively fitting monotone polynomials of increasing degree does not suffer from such problems. Moreover, for monotone polynomials over compact or semi-compact intervals, it is also possible to use such a strategy in instances when the degree is incremented by 1. As monotone polynomials over semi-compact intervals have a single parameterization, given by Equation (6), it would be easy to implement such a strategy. By way of contrast, taking the same incremental approach for fitting monotone polynomials over compact intervals, would necessitate alternating between parameterizations (7) and (8).

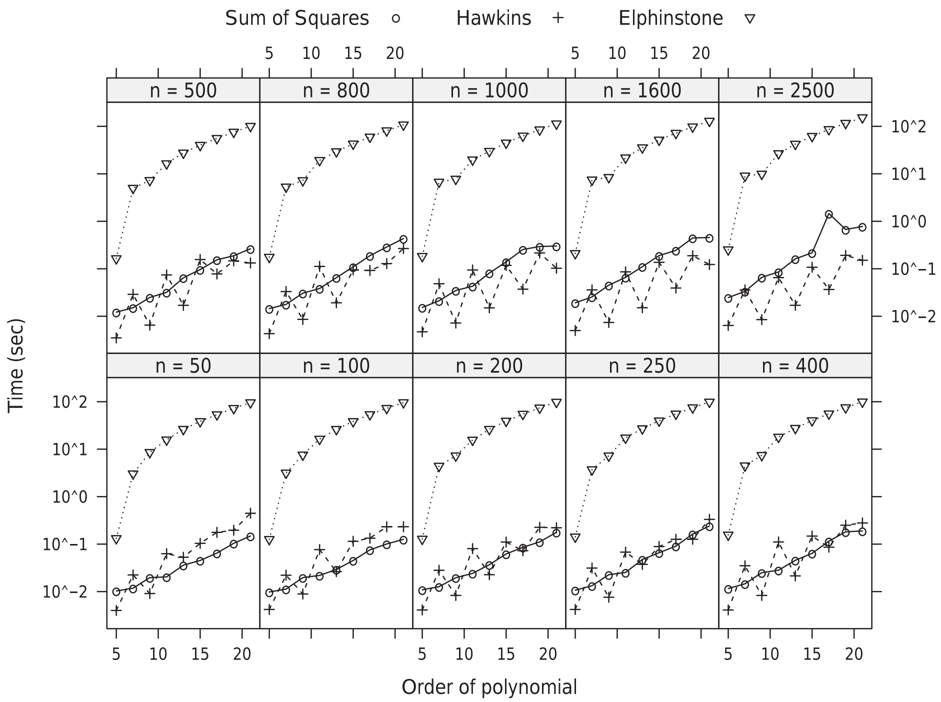

Figure 3. Monotone polynomials of various degrees were fitted to simulated data from a sigmoidal curve (Figure 1). Sample sizes used in this timing study are indicated in the strip of each panel. The plot shows for three approaches to fitting monotone polynomials the average time, over 10 replications, needed to fit the model against the degree of the polynomial.

Table 1. Expected proportion of monotone fits for a range of sample sizes, when the data generating polynomial is $p_k(x)$, $k = 1,2$, and the standard least squares estimator is used to fit unconstrained polynomials.

n	$p_1(x) = 3x^3 + x + 1$	$p_2(x) = x^5$
50	0.727	0.086
100	0.858	0.111
200	0.946	0.141
500	0.995	0.194
1000	1.000	0.240
10,000	1.000	0.356

The RSS as a function of $\boldsymbol{\beta}$ is a convex function, and strictly convex if the number of unique design points is larger than the degree of the polynomial fitted to the data, with no local extrema. Conversely, the RSS as a function of $\boldsymbol{\theta}$ is not convex, and consequently may have local extrema. Table 1 of [22] showed that for some of the isotonic parameterizations used in that paper the iterative optimization algorithm had convergence problems or converged to a local minima. In our extensive numerical experience, when using the sums-of-squares parameterization to fit polynomials that are monotone over the whole real line, and the starting values described in the previous section, there is no evidence of such problems arising, that is, we have not observed any convergence issues or any convergence to local minima. However, when fitting polynomials that are constrained to be monotone over a (semi-)compact region the potential of the algorithm converging to a local minima seems to be greater. Thus, as in all optimization problems in which local minima could be a potential issue, we recommend the use of several random starting values. Additionally, when fitting polynomials (of odd degree) constrained to be monotone over a (semi-)compact interval, we recommend comparing the fit with the polynomial that is constrained to be monotone over the whole real line to ensure that one has obtained at least as good a fit.

3. Properties of monotone polynomial and bootstrap confidence bands

In this section we informally discuss and demonstrate, via numerical examples, properties of monotone polynomials and the use of bootstrap methods for statistical inference, in particular for the calculation of confidence intervals for monotone polynomials. Denote by $\mathcal{C}^q_{\mathcal{R}}$ the set of all $\boldsymbol{\beta} \in \mathbb{R}^{q+1}$ for which $p(x; \boldsymbol{\beta})$, a polynomial of degree at most q, is monotone for all $x \in \mathcal{R} \subseteq \mathbb{R}$. Note that $\mathcal{C}^q_{\mathcal{R}}$ is a (non-pointed) closed convex cone in $\mathbb{R}^{q+1}$. For the sake of brevity, we will refer below to polynomials lying within $\mathcal{C}^q_{\mathcal{R}}$ or on its boundary instead of the vector of coefficients of a polynomial lying within $\mathcal{C}^q_{\mathcal{R}}$ or on its boundary. We will demonstrate that standard bootstrap methodology typically leads to under coverage in certain regions and show that these areas of under coverage appear to be related to whether the polynomial lies within $\mathcal{C}^q_{\mathcal{R}}$ or on its boundary. To address this problem of under coverage we consider two solutions. First, we investigate an 'm out of n' bootstrap which results in similar patterns of coverage probabilities to standard bootstrap methodology, but yields more conservative confidence bands by choosing m smaller than n. Second, by making confidence bands symmetric we show we can ensure the coverage probabilities to be more uniform over the range of the regressor variable.

In our numerical examples, we consider two monotone polynomials,

$$p_1(x) = 3x^3 + x + 1, \quad p_2(x) = x^5,$$

and simulate responses for n equidistant design points over $[-1, 1]$ from the linear regression models

$$Y_i = p_k(x_i) + \epsilon_i, \quad k = 1, 2, \quad x_i = -1 + 2\frac{i-1}{n-1}, i = 1, \ldots, n. \tag{13}$$

For the cubic regression model the errors are independent $N(0, 2^2)$ and for the quintic the errors are independent $N(0, 0.3^2)$.

These two monotone polynomials are specifically chosen to demonstrate how the different bootstrapping techniques work under monotonicity constraints. In particular, $p_1(x)$ lies within $\mathcal{C}^q_{\mathbb{R}}$ for $q = 3$, but on the boundary of $\mathcal{C}^q_{\mathbb{R}}$ for $q \geq 5$, and $p_2(x)$ lies on the boundary of $\mathcal{C}^q_{\mathbb{R}}$ for $q \geq 5$.

Standard least squares regression theory states that when fitting unconstrained polynomials of degree q, the distribution of the $q+1$ vector of parameter estimators, $\hat{\boldsymbol{\beta}}_{\text{unc}}$, has a multivariate normal distribution (MVN) when the errors are independent and identically normal. Specifically $\hat{\boldsymbol{\beta}}_{\text{unc}} \sim \text{MVN}(\boldsymbol{\beta}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ is the variance covariance matrix given by $\sigma^2(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}$ and $\boldsymbol{X}$ is the design matrix with ith row $\mathbf{x}_i = (1, x_i, x_i^2, \ldots, x_i^q)$. To provide an idea of the necessity of constraining fitted polynomials to be monotone, we carry out Monte-Carlo integration of the density of $\hat{\boldsymbol{\beta}}_{\text{unc}}$ over $\mathcal{C}^q_{\mathbb{R}}$, $q = 3$ for $p_1(x)$ and $q = 5$ for $p_2(x)$. The observed probabilities of monotone fits, using standard unconstrained polynomials and least squares estimation, are shown in Table 1. As one would expect, for $p_1(x)$ the probability of a fit being monotone quickly converges to 1. By way of contrast, for $p_2(x)$, which lies on the boundary of $\mathcal{C}^q_{\mathbb{R}}$, the probability of a fit being monotone does not converge to 1 as $n \to \infty$. Consequently, a monotone polynomial estimator of a polynomial lying within $\mathcal{C}^q_{\mathcal{R}}$ (for some q and some $\mathcal{R}$) is asymptotically equivalent to the unconstrained polynomial estimator, and therefore inherits all the properties of the latter, such as consistency and multivariate normality of the estimated vector of coefficients. Conversely, for a polynomial lying on the boundary, the monotone polynomial estimator is, by the triangle inequality, closer to the 'true' polynomial than the unconstrained polynomial estimator, as the former is the orthogonal projection (with respect to some norm) of the latter onto $\mathcal{C}^q_{\mathcal{R}}$. Consistency of the constrained polynomial follows from this observation, although in this case the asymptotic distribution of the constrained polynomial estimator is more complicated; see also [42] or [43].

A slightly more careful argument for consistency of the monotone polynomial estimator would be necessary if q is also estimated, for example, via 'm out of n' bootstrap, as we propose and discuss in Section 4. However, the theory of the 'm out of n' bootstrap indicates that it provides a consistent estimator $\hat{q}$ of the degree of an underlying monotone polynomial [44] thus, the monotone polynomial fit from $\mathcal{C}_{\hat{q}}^{\mathcal{R}}$ would be consistent for the underlying monotone function.

We do not regard monotone polynomials (of high degrees) as a monotone smoothing technique, or as competitor of such techniques. It seems possible from Shevchuk's result,[40] that a consistency result similar to the one by Mammen [17] for monotone kernel smoothers can be established for monotone polynomials in a non-parametric smoothing setting. Establishing such a result more rigorously, in particular if it is coupled with an investigation of the rate of consistency (which would require that $q = q(n) \to \infty$ as $n \to \infty$) might be of some theoretical interest, and is left for future research. Finally, extensive Monte-Carlo simulations examining the accuracy of the point estimates through calculation of bias and mean-squared error have been carried out in [39].

3.1. Bootstrap confidence bands for monotone polynomials

Due to the complex nature of the isotonic parameterizations, we propose to calculate (point wise) confidence bands using bootstrap methodology. We mainly focus on the results for the model $p_2(x) + \epsilon$, with $n = 50$ and $n = 1000$ and use both a standard residual bootstrap algorithm and a non-parametric (paired) bootstrap approach, with 1000 bootstrap replications. Our numerical experiments (not reported here) demonstrate little difference between these two approaches. An example of the resulting confidence bands using the paired bootstrap is shown in Figure 4 for both true data generating functions $p_1(x)$ and $p_2(x)$. The top panel depicts the fitted 95% (and 80%) confidence bands and it is noticeable, for both functions, that these have an asymmetric nature. We show the deviations from the fitted curve to the upper and lower limits in the bottom panels of Figure 4 to demonstrate more clearly this asymmetry. We note that these deviations are more pronounced in certain regions and this is dependent on the underlying functions. Using either the residual or the non-parametric bootstrap produces confidence bands that themselves are monotonic increasing. However, we will demonstrate in the next subsection that these exhibit, in some instances, poor coverage probabilities for certain ranges of the regressor variable.

3.1.1. Coverage probabilities

To examine the effectiveness of the bootstrap, we estimate coverage probabilities extracting the 10th and 90th percentiles of the 1000 bootstrap replications for each of 5000 simulation runs. Figure 5 shows the coverage probabilities for $p_1(x)$ (top panel) and $p_2(x)$ (bottom panel), using both standard unconstrained least squares estimation (left panel) and monotone polynomial fitting (right panel). We note that with standard least squares fitting the coverage probabilities are relatively uniform over the range of the regressor variable and there is under-coverage in the smaller sample size ($n = 50$, solid blue line) as opposed to the near nominal coverage for ($n = 1000$ orange dashed line), as expected. However, when these models are fit using monotone polynomials we note two things. First, for $p_1(x)$, the coverage probabilities are near the nominal level for both sample sizes. However, when we look at the coverage probabilities for $p_2(x)$ we see a distinct pattern with undulations around $x = -0.5$ and $x = 0.5$ depicting areas of extreme under-coverage (sometimes as low as 0.65). Second, we note that increasing the sample size does not impact on this to any great extent. From this it is evident that straight forward off the shelf bootstrap methodology alone is not suitable for monotone polynomials in all situations, and is particularly poor when the underlying regression function is on the boundary of $\mathcal{C}_{\mathcal{R}}^{q}$. Furthermore, and upon examination of alternative regression functions, we suggest that this observed under coverage, using both bootstrap methodologies, is not restricted to the specific degree of polynomial nor the sample size.

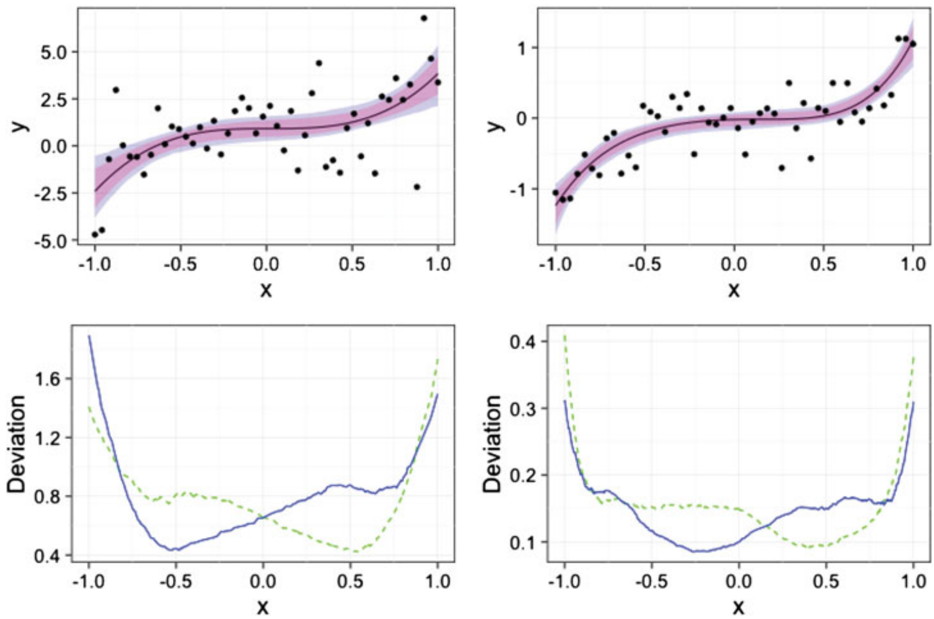

Figure 4. Top row: Example 80% and 95% bootstrap confidence bands for each of the two functions $p_1(x)$ (top left), $p_2(x)$ (top right). Bottom row: deviation from the fitted curve to the upper and lower 80% confidence limits; deviation to lower limit (green dashed line) and deviation to upper limit (blue solid line) for each of the two functions respectively demonstrating the asymmetric nature of the confidence bands.

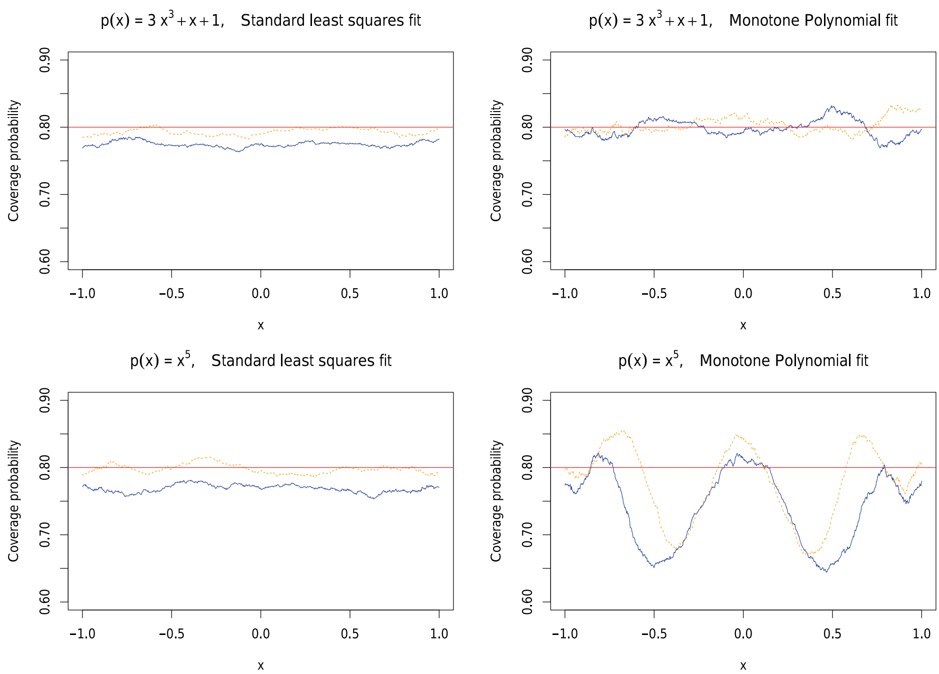

Figure 5. Coverage probabilities for 80 % confidence intervals for the cubic $p_1(x) = 3x^3 + x + 1$ and the quintic $p_2(x) = x^5$, for sample sizes $n = 50$ (blue solid line) and $n = 1{,}000$ (orange dashed line). Left panel – standard least squares fits using 'n out of n' residuals bootstrap. Right panel – monotone polynomial fits using 'n out of n' residuals bootstrap.

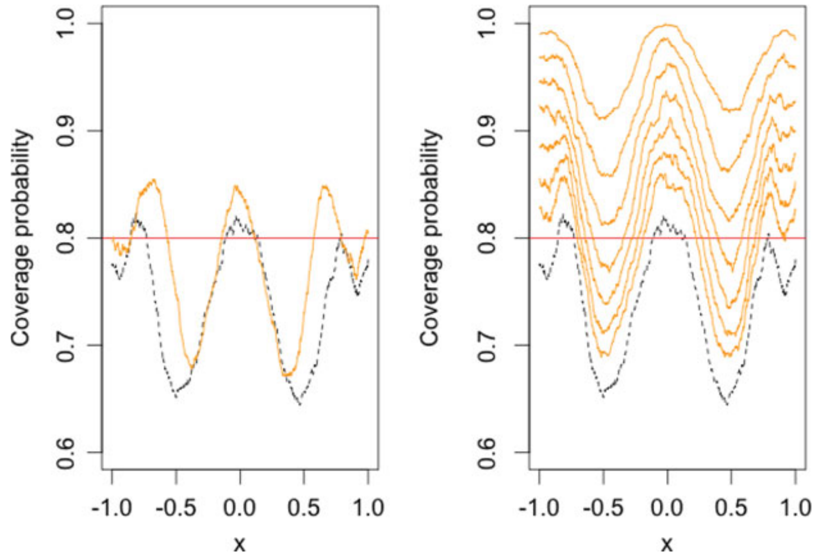

Figure 6. Coverage probabilities for 80 % confidence intervals for the quintic monotone polynomial $p_2(x) = x^5$. Left panel – '*n* out of *n*' residuals bootstrap: Black dashed line $n = 50$; orange solid line $n = 1{,}000$. Right panel – '*m* out of *n*' bootstrap: decreasing values of *m* from bottom (black dashed) line $m = n$ through $m = 40, 35, 30, 25, 20, 15, 10$ for $n = 50$ design points.

3.1.2. Using the 'm out of n' bootstrap.

In our numerical examples, adopting an '*m* out of *n*' bootstrap, as described in [45], rectifies this under coverage. The right panel of Figure 6 shows coverage probabilities for 80% confidence intervals based on the '*m* out of *n*' bootstrap, varying *m* over 10, 15, 20, 25, 30, 35, 40 and 50 for $n = 50$, whilst the left panel shows the coverage probabilities for the '*n* out of *n*' bootstrap for $n = 50$ and $n = 1000$. We note that using an '*m* out of *n*' bootstrap gives more acceptable results in terms of coverage probabilities for an astute choice of *m*. In this example we observe that $m = 35$, which is relatively close to *n*, achieves an average coverage probability that is close to the nominal level, whereas for small choices of *m* the resulting confidence bands are very conservative. We found this to be true for a range of examples (not reported here) and depending on *m* there are three possibilities in general: we either choose *m* too large and get under coverage; we can choose *m* too small so that the coverage probabilities on the whole exceed the nominated confidence level, hence we get substantial over coverage in some areas; or we select *m* approximately optimal to obtain average coverage over the range to be at the nominated confidence level, which results in some areas with under coverage and some areas with over coverage. However, we note that a data driven choice of *m* remains an open research question. Table 2 summarizes coverage probabilities for varying size *m*. In particular we note that using $m = 35$ gives the best overall mean coverage probabilities. However, using a criteria such as minimizing mean or median absolute deviation may suggest an alternative choice for *m*.

We further observe from the right panel of Figure 6 that regardless of the magnitude of *m* the coverage probabilities are not uniform over *x*. To examine this in more detail we propose the idea of a post hoc adjustment. Our conjecture is that this under coverage results from the asymmetric nature of the confidence bands arising from the monotonicity constraint, and that the resulting under coverage is particularly pronounced in underlying models that are on or close to the boundary of the cone of monotone polynomials.

3.1.3. Post hoc adjustments to confidence bands

We consider a post hoc adjustment and illustrate it for $p_2(x)$. We propose a simple adjustment to the confidence bands estimated using an '*n* out of *n*' bootstrap; specifically we make these bands

Table 2. Summaries of coverage probabilities from different '*m* out of *n*' bootstrapping algorithms and post hoc adjustments.

Algorithm	Mean	Median	Min	Max	Mean deviation	Median deviation	SABC	% above 0.8
$m = 10$	0.961	0.966	0.910	0.999	0.161	0.166	0.321	1.000
$m = 15$	0.924	0.929	0.857	0.986	0.124	0.129	0.249	1.000
$m = 20$	0.890	0.900	0.812	0.963	0.090	0.100	0.180	1.000
$m = 25$	0.859	0.872	0.768	0.937	0.066	0.072	0.118	0.784
$m = 30$	0.831	0.849	0.734	0.912	0.057	0.060	0.061	0.660
$m = 35$	0.804	0.820	0.708	0.880	0.052	0.055	0.008	0.564
$m = 40$	0.782	0.798	0.689	0.860	0.049	0.045	−0.037	0.490
$m = n$ (No Adj.)	0.741	0.757	0.644	0.822	0.063	0.043	−0.118	0.185
Post hoc Adj.	0.840	0.840	0.759	0.887	0.043	0.040	0.081	0.909

Note: Signed area between curves (SABC) denotes the signed area of the difference of the estimated coverage probability curve and the nominal coverage line.

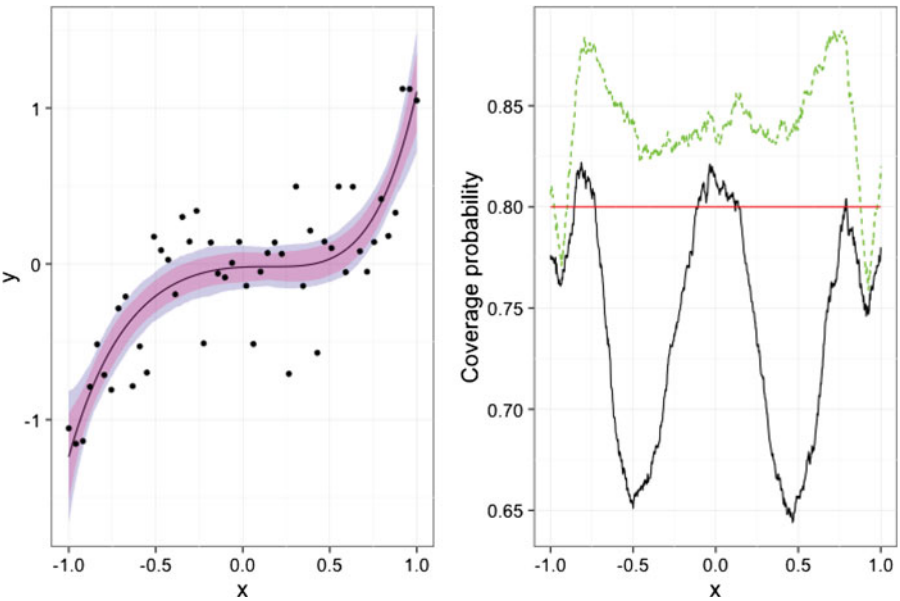

Figure 7. Left panel: Fitted monotone polynomial based on simulated data from underlying function $p_2(x) = x^5$ with adjusted 80% and 95% bootstrap confidence bands. Right panel: Coverage probabilities for adjusted 80% bootstrap confidence bands on monotone polynomials based on the underlying function $p_2(x) = x^5$; adjusted probabilities (green), unadjusted (black).

symmetric around $\hat{p}(x)$. Let $\delta_{u,x}$ and $\delta_{l,x}$ be the upper and lower deviations from $\hat{p}(x)$ for a given x. We define an adjusted confidence band for a given x to be $[\hat{p}(x) - \max\{\delta_{u,x}, \delta_{l,x}\}, \hat{p}(x) + \max\{\delta_{u,x}, \delta_{l,x}\}]$. Such an adjustment is illustrated in Figure 7 with the left panel showing the fitted confidence band for an example simulation realization, and the right panel showing the new coverage probabilities after this symmetrization of the confidence bands over the 1000 simulation runs. We observe that the apparent under coverage initially observed has now been removed by this adjustment and the coverage is much more uniform over the whole range of x. However, we acknowledge that these intervals may now be somewhat conservative with marginally higher coverage than the nominal.

Numerical experiments (not reported here) using other functions, with varying degree polynomials, show similar positive results, to suggest such a post hoc adjustment could be beneficial for monotone polynomial bootstrap confidence bands.

3.2. Bootstrap prediction bands for monotone polynomials

In addition to the investigation of confidence bands, we also studied and adapted a previous approach to use the bootstrap to calculate prediction intervals [46] for monotone polynomials. In our extensive numerical experiments we found that, in contrast to the confidence bands described earlier, these prediction intervals work equally well for polynomials lying inside the cone of monotone polynomials of order at most q as for polynomials on the boundary of the cone, resulting in actual prediction coverage probabilities close to nominal levels. Presumably the extra variance term that appears in the calculation of prediction intervals dominates the width of these intervals and, consequently, close to nominal coverage is also obtained for polynomials on the boundaries, for which the usual asymptotics with respect to asymptotic distribution and/or confidence intervals do not apply.

4. Real world examples

We demonstrate the effectiveness of our techniques through two real world examples.

4.1. Brain function data

We describe our first example data set adapting the description from Firmin et al.[5,6] The Latency of Motor evoked potentials (MepL) distribution for each of 50 subjects was derived from the triple stimulation technique (TST) amplitude measurements, where the expected TST amplitude $E(Y_{ijk})$, which is a function of δt, is an unscaled survival function of the MepL distribution. The maximal value of the expected TST amplitude served as baseline for 100% survival and proportions thereof resulted in $1 - F_i(x)$, where F denotes the true but unknown MepL distribution, $x = \delta t$, and i is the index for the ith subject. TST amplitude measurements were available at stimulus delays in steps of 1 ms and for most stimuli with smaller delay extension, $j < 9$, there were at least $n_{ij} = 3$ repeated measurements per subject. For example in Figure 8 we have $\{5, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 1\}$ observations at each of the increasing design points from 0 through 11 in increments of 1. The particular interest of Firmin et al. [5,6] was to determine the inflection points of the estimated survival function, that is, the modes (peaks) of the estimated MepL density within the support of the density which coincide with the inflection points of the estimated mean curves. These were estimated previously using monotone smoothing splines.[5,6] We will consider the same problem using monotone polynomials, which have the advantage of well defined inflection points.

We consider using the following polynomials:

$$p(x; \boldsymbol{\beta}) = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_q x^q, \tag{14}$$

where in this instance q is not restricted to be odd, as with previously described monotone polynomial fitting, but $\boldsymbol{\beta}$ such that $p(x)$ is monotone over $x \in [0, \infty)$ reflecting the nature of the brain function data.

An initial examination of the 50 individual data sets involved fitting cubic, quintic and septic unconstrained polynomials using standard least squares. Of the 150 unconstrained models fitted we observed only 13 which provided monotone fits, further strengthening the argument of the need for the constrained version of polynomials. We consider for demonstrative purposes one of these patients and describe the differences between the three different polynomial methodologies in Figure 8, where we show the data and three fits of degree 7 polynomials (the unconstrained fit, the monotone over the whole real line fit, and the monotone over the semi-compact interval fit). The horizontal axis has been extended outside the range of the observed x data values, for illustrative purposes only, to demonstrate where the curves become non-monotone. We note that, over the range of the data, there are noticeable differences between the unconstrained polynomial fit and the two constrained polynomial fits, with little difference between the latter two.

For model selection purposes, as suggested previously,[44] we consider using the 'm out of n' paired bootstrap, implemented over a range of values for m. For each bootstrap sample we calculate

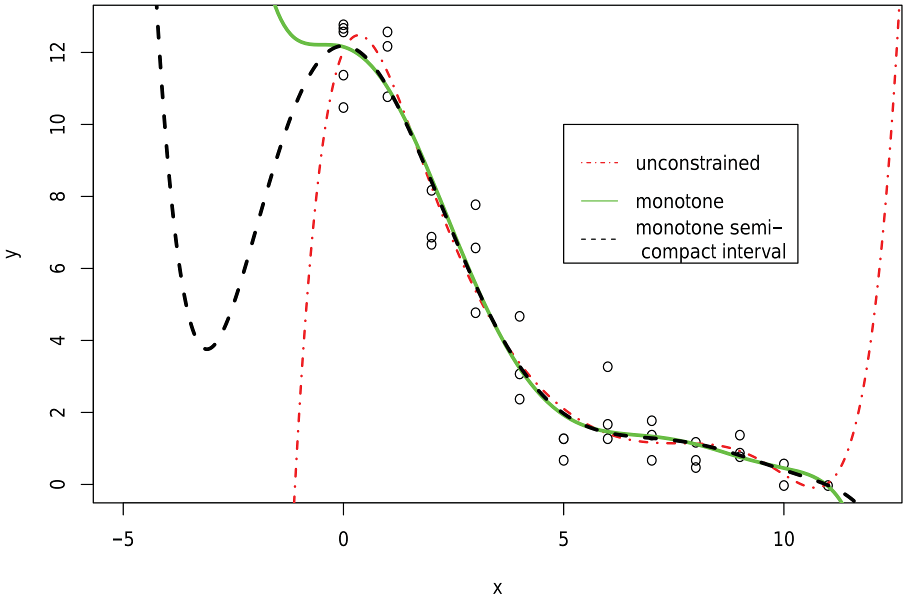

Figure 8. Different degree 7 polynomial fits to brain function data of an individual.

the prediction errors for fitted monotone polynomials of degrees $q = 1, \ldots, 11$, using:

$$PE(q) = u^{-1} \sum_{j=1}^{u} (y_{i_j} - p(x_{i_j}; \hat{\boldsymbol{\beta}}_q))^2, \tag{15}$$

where $\hat{\boldsymbol{\beta}}_q$ is the estimated vector of coefficients for the monotone polynomial of degree q, based on the bootstrap sample; and $(x_{i_j}, y_{i_j}), j = 1, \ldots, u \geq n - m$, are the (unselected) out-of-bag (x, y) pairs. For each bootstrap iteration the polynomial with degree $q = \arg\min \mathrm{PE}(q)$, is selected as the 'best' model.

To reflect the nature of the data and the design matrix (i.e. a relatively small number of unique design points), we use a stratified version of the paired bootstrap.[47,48] Stratification facilitates that individual bootstrap samples are not too different to the full data, for example ensuring that samples have sufficient design spread to enable reasonable polynomial fits. We let $D = \{d_1, d_2, \ldots, d_{n_d}\}$ be the set of values at which we have design points, and $\#D = n_d$ be the number of unique design points. We also let $\#d_i$ be the number of values at each design point d_i, $i = 1, \ldots, n_d$. If $n_d \leq m$ we sample one value from each unique design point, and randomly select with replacement and with probability proportional to $\#d_i$ another $m - n_d$ design points to sample additional observations from. If $n_d > m$ we simply sample without replacement m design points and select one observation from each. Minimizing prediction error as the objective function, as described previously, we examine the selection probabilities of the differing degree polynomials. We consider only polynomials up to degree 8 (restricted by the number of design points) that are constrained to be monotone over a semi-compact region and values of m ranging over 10, 15, 20, 25, 30 and 35. Results are described in Table 3 and Figure 9. One can immediately determine that the best fitting degree of constrained monotone polynomial, irrespective of the size of m, is degree 6, thus highlighting the benefits of our improved algorithm over previous algorithms which were restricted to fitting monotone polynomials of only odd degree.

Table 3. Brain function data: proportion of models selected using '*m* out of *n*' bootstrap by degree *q* and *m*.

	Degree of polynomial *p*							
m	1	2	3	4	5	6	7	8
10	0.038	0.034	0.256	0.166	0.052	**0.374**	0.052	0.028
15	0.000	0.000	0.060	0.077	0.046	**0.563**	0.219	0.035
20	0.000	0.000	0.082	0.083	0.048	**0.553**	0.193	0.041
25	0.001	0.000	0.098	0.055	0.055	**0.562**	0.190	0.039
30	0.000	0.001	0.111	0.040	0.051	**0.569**	0.203	0.025
35	0.001	0.003	0.131	0.036	0.041	**0.546**	0.216	0.026

Note: The significance of the bold values is that these are the largest proportions in each row.

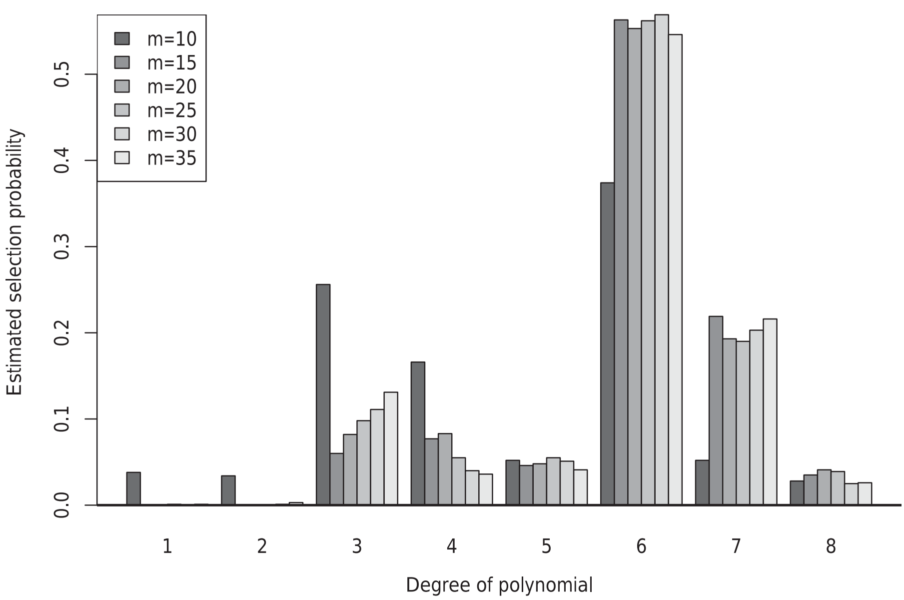

Figure 9. Selection probabilities for the '*m* out of *n*' bootstrap by degree of polynomial and *m* for brain function data.

For completeness (not described here) we carried out the bootstrapping without stratification and results were comparable to the results obtained from the stratified bootstrap. However, in some simulations runs higher degree polynomials could not be fitted or were highly variable, mainly due to attempting to have more parameters estimated than unique design points.

4.2. Estimation of dental age

In forensic science, methods for estimating dental maturity or dental age by reference to a scoring system created from the radiological appearance of teeth is commonly used. [49]Using population-based data a simple plotting method with subsequent extraction of percentiles for age estimation for a given maturity score was described. Subsequently regression-based methods were mainly used for smaller data sets and comparisons using polynomials to model the relationship between age and the dental maturity score with the original percentile method [49] have been made.[50] Using data described in [4] in which $n = 284$ adolescent teenagers were examined, with the objective of fitting models to describe the relationship between a dental maturity score (x) and age of child (y), we consider modelling this relationship using polynomials, and incorporate the underlying widely accepted view in the forensic science literature that the relationship be monotone increasing. Due to the nature

Table 4. Estimation of dental age data: proportion of models selected using '*m* out of *n*' bootstrap by degree *q* and *m*.

	Degree of polynomial *q*										
m	1	2	3	4	5	6	7	8	9	10	11
20	0.005	0.176	0.080	0.136	**0.219**	0.098	0.085	0.051	0.067	0.043	0.040
25	0.000	0.128	0.070	0.143	**0.236**	0.104	0.078	0.066	0.074	0.047	0.054
30	0.000	0.106	0.052	0.131	**0.249**	0.107	0.126	0.075	0.076	0.022	0.056
35	0.000	0.078	0.037	0.146	**0.308**	0.108	0.089	0.050	0.084	0.038	0.062
40	0.000	0.050	0.025	0.123	**0.318**	0.115	0.109	0.071	0.084	0.042	0.063
50	0.000	0.043	0.022	0.123	**0.326**	0.105	0.088	0.075	0.113	0.045	0.060
100	0.000	0.007	0.003	0.121	**0.487**	0.085	0.047	0.048	0.130	0.030	0.042
n	0.000	0.000	0.016	0.106	**0.474**	0.088	0.032	0.032	0.141	0.056	0.055

Note: The significance of the bold values is that these are the largest proportions in each row.

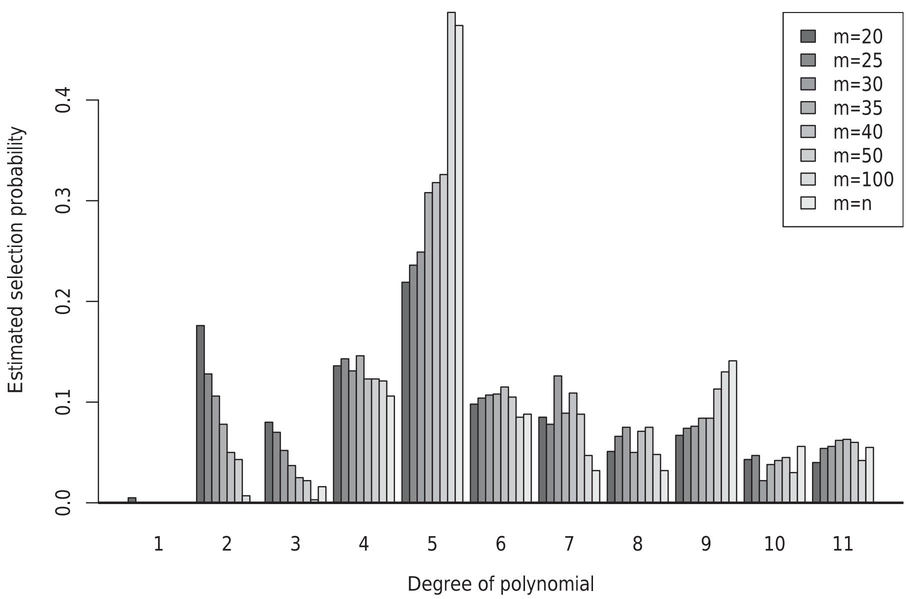

Figure 10. Selection probabilities for the '*m* out of *n*' bootstrap for dental maturity example by degree *q* and a range of *m* values.

of how the score (x variable) is generated, we also constrain the polynomial to be monotone over the compact interval [0, 100]. Hence we consider using the polynomials described in Equation (14).

For model selection purposes, we again consider using the 'm out of n' paired bootstrap. Here the sample size is $n = 284$ and we consider values of m ranging over 20, 25, 30, 35, 40, 50, 100 and n. For each bootstrap sample we calculate the prediction errors for fitted monotone polynomials of degrees $q = 1, \ldots, 11$, again using Equation (15).

Table 4 shows the proportion of times the polynomial of degree q is selected. Consistent results are obtained regardless of the magnitude of m, with a distinct mode in the distribution of the proportions at $q = 5$. This is visualized in Figure 10. These results indicate a higher degree polynomial than that usually used when polynomials are fit to this type of data in forensic science, which is usually $q = 3$. However, we suspect that the main reason for this is that with increasing degree polynomials the chance of obtaining a monotone fit, as needed for this type of data, is smaller and cubics are generally the only polynomials which will give a monotone fitted curve. Our methodology now provides more flexibility in the degree of polynomials that can be used for these types of examples.

5. Conclusions

Fitting unconstrained polynomials to data has historically been an easy task and is popular mainly due to the simplicity of the functions and the readily available theoretical results on the properties of the estimators. The constrained version of such polynomials has not received as much publicity mainly due to the lack of available software to implement such methods and the difficulty in providing effective optimization algorithms. We have shown through our simple real world examples that there is not only a need for such methodology but a necessity in some instances where standard polynomials do not provide intuitively correct estimated functions, for example ensuring monotonicity. Previously some ideas were postulated on how such models can be fitted with various different formulations and parameterizations.[22] These methods were made available through the `MonoPoly` package in R.

Using the new methodology proposed in this article we have further developed the `MonoPoly` package, specifically with its latest version now including the ability to fit monotone polynomials using the *sum of squared polynomial* formulation. This new formulation itself has led to a dramatic reduction of time to fit such models compared to previously described results. In addition, and as a consequence of the sums-of-squares formulation, we presented methodology in which we can constrain the polynomial to be monotone over the full region of x values, over a semi-compact region, or over compact regions. These latter two methodologies allow even degree polynomials to be considered, something previously unattainable in the fitting of monotone polynomials to data. Our real world examples have demonstrated that using this methodology can describe a relationship in a more efficient way, and, using prediction error as model selection criterion, can select even degree polynomials that are monotone over a compact or a semi-compact region as the best model.

We have further proposed methodology for estimating confidence bands for monotone polynomials and described the difficulties in doing so. In results not described in this article we carried out an investigation of numerous different parametric and non-parametric bootstrap methodologies including some of those described in [33], which have demonstrated varying degrees of success for constrained non-parametric regression. However, results suggest that these methodologies and standard bootstrap methodologies may need adaptation to ensure more uniform coverage probabilities over the range of x values. In light of this we advocate one of two approaches; either using an m out of n bootstrap, or by employing a post hoc curvature adjustment. Initial testing of such approaches has yielded some encouraging results with the m out of n bootstrap providing an average coverage probability over the range of x at the nominated level, if one selects the right choice of tuning parameter. However, the post-hoc adjustment appears to have practical promise, given its ability to ensure the coverage probabilities are more uniform. Details of the mathematical properties of such adjustment remain to the best of our knowledge an open research question.

In our experience in using the m out of n bootstrap, the selection of m through our real world examples has minimal impact on the model selection aspect of the problem, that is, choosing the degree q. However, in our extensive simulations moderate values of m performed best. In some instances a stratified bootstrap proved to be a more efficient and sensible approach which, if employed, would also impact on the selection of m. We have also shown that in some instances, using a constrained polynomial bounded over a semi-compact region, chooses a smaller degree polynomial than that of the corresponding unconstrained or monotone polynomials described previously.[22]

All code to fit the models in this article are available in the most recent version of the R [51] package `MonoPoly` soon to be available on CRAN.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

The work of the second author was supported by the Australian Research Council [grant number DP140101259].

ORCID

B. A. Turlach http://orcid.org/0000-0001-8795-471X

References

[1] Cole TJ, Freeman JV, Preece MA. British 1990 growth reference centiles for weight, height, body mass index and head circumference fitted by maximum penalized likelihood. Stat Med. 1998;17(4):407–429.
[2] Durbán M, Harezlak J, Wand MP, Carroll RJ. Simple fitting of subject-specific curves for longitudinal data. Stat Med. 2005;24(8):1153–1167.
[3] Ganger J, Brent M. Reexamining the vocabulary spurt. Dev Psychol. 2004;40(4):621–632.
[4] Flood SJ, Mitchell WJ, Oxnard CE, Turlach BA, McGeachie J. To evaluate the utility of smaller sample sizes when assessing dental maturity curves for forensic age estimation. J Forensic Sci. 2011;56(6):1604–1609.
[5] Firmin L, Müller S, Rösler KM. A method to measure the distribution of latencies of motor evoked potentials in man. Clin Neurophysiol. 2011;122(1):176–182.
[6] Firmin L, Müller S, Rösler KM. The latency distribution of motor evoked potentials in patients with multiple sclerosis. Clin Neurophysiol. 2012;123(12):2414–2421.
[7] Ratkowsky DA. Handbook of nonlinear regression models. New York: Marcel Dekker; 1990.
[8] Mirman D. Growth curve analysis and visualization using r. The R series. Boca Raton: Chapman & Hall/CRC; 2014.
[9] Panik JM. Growth curve modeling: theory and applications. Hoboken: John Wiley & Sons; 2014.
[10] Turlach BA. Shape constrained smoothing using smoothing splines. Comput Statist. 2005;20(1):81–103.
[11] Hazelton ML, Turlach BA. Semiparametric regression with shape constrained penalized splines. Comput Statist Data Anal. 2011;55(10):2871–2879.
[12] Meyer MC. Inference using shape-restricted regression splines. Ann Appl Stat. 2008;2(3):1013–1033.
[13] Meyer MC. Constrained penalized splines. Canad J Statist. 2012;40(1):190–206. doi:10.1002/cjs.10137.
[14] Marron JS, Turlach BA, Wand MP. Local polynomial smoothing under qualitative constraints. In: Billard L, Fisher NI, editors. Graph-image-vision; computing science and statistics; Vol. 28. Fairfax Station, VA 22039–7460: Interface Foundation of North America, Inc.; 1997. p. 647–652.
[15] Mammen E, Marron JS, Turlach BA, Wand MP. A general projection framework for constrained smoothing. Statist Sci. 2001;16(3):232–248.
[16] Friedman JH, Tibshirani R. The monotone smoothing of scatterplots. Technometrics. 1984;26(3):243–250.
[17] Mammen E. Estimating a smooth monotone regression function. Ann Statist. 1991;19(2):724–740.
[18] Hall P, Huang LS. Nonparametric kernel regression subject to monotonicity constraints. Ann Statist. 2001;29(3):624–647.
[19] Dette H, Neumeyer N, Pilz KF. A simple nonparametric estimator of a strictly monotone regression function. Bernoulli. 2006;12(3):469–490.
[20] Dette H, Pilz KF. A comparative study of monotone nonparametric kernel estimates. J Stat Comput Simul. 2006;76(1):41–56.
[21] Ramsay JO. Estimating smooth monotone functions. J R Stat Soc Ser B. 1998;60(2):365–375.
[22] Murray K, Müller S, Turlach BA. Revisiting fitting monotone polynomials to data. Comput Statist. 2013;28(5):1989–2005.
[23] Elphinstone CD. A target distribution model for non-parametric density estimation. Comm Statist Theory Methods. 1983;12(2):161–198.
[24] Hawkins DM. Fitting monotonic polynomials to data. Comput Statist. 1994;9(3):233–247.
[25] Hall P. The bootstrap and edgeworth expansion. New York: Springer-Verlag; 1992.
[26] Efron B, Tibshirani RJ. An introduction to the bootstrap. Monographs on statistics and applied probability, Vol. 57. London: Chapman and Hall; 1993.
[27] Hjorth JSU. Computer intensive statistical methods: validation, model selection and bootstrap. London: Chapman and Hall; 1994.
[28] Shao J, Tu D. The jackknife and bootstrap. Springer series in statistics. New York: Springer-Verlag; 1995.
[29] Davison AC, Hinkley DV. Bootstrap methods and their application. Cambridge Series in Statistical and Probabilistic Mathematics, Vol. 1. Cambridge: Cambridge University Press; 1997.
[30] Lunneborg CE. Data analysis by resampling: concepts and applications. Pacific Grove (CA): Duxbury Press; 2000.
[31] Hall P, Horowitz J. A simple bootstrap method of constructing nonparametric confidence bands for functions. Ann Statist. 2013;41(1):1982–1921.
[32] Dümbgen L. Optimal confidence bands for shape-restricted curves. Bernoulli. 2003;9(3):423–449.
[33] Strand M, Zhang Y, Swihart B. Monotone nonparametric regression and confidence intervals. Comm Statist Simulation Comput. 2010;39:828–845.
[34] Karlin S, Studden WJ. Tchebycheff systems: with applications in analysis and statistics. New York: Wiley & Sons; 1966.

[35] Dette H, Studden WJ. The theory of canonical moments with applications in statistics, probability, and analysis. Wiley series in probability and statistics. New York: Wiley & Sons; 1997.
[36] Brickman L, Steinberg L. On nonnegative polynomials. Amer Math Monthly. 1962;69(3):218–221.
[37] Fausett LV. Numerical methods: algorithms and applications. Upper Saddle River (NJ): Prentice Hall; 2003.
[38] Osborne MR. Nonlinear least squares – the Levenberg algorithm revisited. J Aust Math Soc. 1976;19(3):343–357.
[39] Murray K. Improved monotone polynomial fitting with applications and variable selection [PhD thesis]. Sydney, NSW, Australia: The University of Sydney; 2015.
[40] Shevchuk IA. Approximation of monotone functions by monotone polynomials. Russ Acad Sci Sb Math. 1993;76(1):51–64. doi:10.1070/SM1993v076n01ABEH003401.
[41] Heinzmann D. A filtered polynomial approach to density estimation. Comput Statist. 2008;23(3):343–360. doi:10.1007/s00180-007-0070-z.
[42] Barlow RE, Bartholomew DJ, Bremner JM, Brunk HD. Statistical inference under order restrictions. London: John Wiley & Sons; 1972.
[43] Silvapulle MJ, Sen PK. Constrained statistical inference: inequality, order and shape restrictions. Wiley series in probability and statistics. Hoboken (NJ): John Wiley & Sons; 2005.
[44] Shao J. Bootstrap model selection. J Amer Statist Assoc. 1996;91(434):655–665.
[45] Shao J. Bootstrap sample size in non-regular cases. Proc Amer Math Soc. 1994;122(4):1251–1262.
[46] Stine RA. Bootstrap prediction intervals for regression. J Amer Statist Assoc. 1985;80(392):1026–1031.
[47] Müller S, Welsh A. Outlier robust model selection in linear regression. J Amer Statist Assoc. 2005;100(472):1297–1310.
[48] Müller S, Welsh AH. Robust model selection in generalized linear models. Statist Sinica. 2009;19(3):1155–1170.
[49] Demirjian A, Goldstein H, Tanner JM. New system of dental age assessment. Hum Biol. 1973;45(2):211–227.
[50] Chaillet N, Nystrom M, Kataja M, Demirjian A. Dental maturity curves in finnish children: Demirjian's method revisited and polynomial functions for age estimation. J Forensic Sci. 2004;49(6):1324–1331.
[51] R Core Team. R: a language and environment for statistical computing. R Foundation for Statistical Computing; Vienna, Austria; 2015. Available from: http://www.R-project.org/.