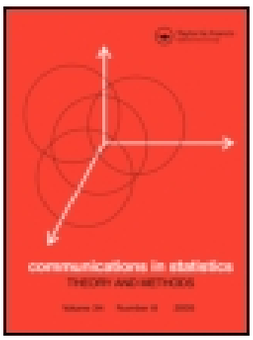

Taylor & Francis
Taylor & Francis Group

Communications in Statistics - Theory and Methods

ISSN: 0361-0926 (Print) 1532-415X (Online) Journal homepage: http://www.tandfonline.com/loi/lsta20

Jump-detection and curve estimation methods for discontinuous regression functions based on the piecewise B-spline function

Guo-Xiang Liu, Meng-Meng Wang, Xiu-Li Du, Jin-Guan Lin & Qi-Bing Gao

To cite this article: Guo-Xiang Liu, Meng-Meng Wang, Xiu-Li Du, Jin-Guan Lin & Qi-Bing Gao (2017): Jump-detection and curve estimation methods for discontinuous regression functions based on the piecewise B-spline function, Communications in Statistics - Theory and Methods, DOI: 10.1080/03610926.2017.1400061

To link to this article: https://doi.org/10.1080/03610926.2017.1400061

Published online: 27 Nov 2017.

COMMUNICATIONS IN STATISTICS—THEORY AND METHODS
2017, VOL. 0, NO. 0, 1–21
https://doi.org/10.1080/03610926.2017.1400061

Jump-detection and curve estimation methods for discontinuous regression functions based on the piecewise B-spline function

Guo-Xiang Liu[b], Meng-Meng Wang[b], Xiu-Li Du [a,b], Jin-Guan Lin[c], and Qi-Bing Gao[b]

[a]Department of Mathematics, Southeast University, Nanjing, China; [b]College of Mathematical Sciences, Nanjing Normal University, Nanjing, China; [c]Institute of Statistics and Data Science, Nanjing Audit University, Nanjing, China

ABSTRACT

Jump-detection and curve estimation methods for the discontinuous regression function are proposed in this article. First, two estimators of the regression function based on B-splines are considered. The first estimator is obtained when the knot sequence is quasi-uniform; by adding a knot with multiplicity $p+1$ at a fixed point x_0 on support $[a, b]$, we can obtain the second estimator. Then, the jump locations are detected by the performance of the difference of the residual sum of squares $DRSS(x_0)$ $(x_0 \in (a, b))$; subsequently the regression function with jumps can be fitted based on piecewise B-spline function. Asymptotic properties are established under some mild conditions. Several numerical examples using both simulated and real data are presented to evaluate the performance of the proposed method.

ARTICLE HISTORY
Received 31 July 2017
Accepted 26 October 2017

KEYWORDS
B-spline; Discontinuous regression function; Jump detection; Local linear kernel smoothing; Piecewise B-spline.

MATHEMATICS SUBJECT CLASSIFICATION

1. Introduction

Non parametric regression is an important branch in statistics. In the past several decades, statistical inference about the continuous non parametric regression functions has been the focus of attention. But in many practical applications, the discontinuous regression functions, namely, the regression functions with jumps seem to be more appropriate to describe the related phenomena. For example, the stock price series, the oil price series, the gold price series, and the price index series of stock market often contain jumps (see the real-data example in Sec. 6); the equi-temperature surfaces of the high sky and the deep ocean also involve discontinuity (see Wahba 1986). Therefore, the research on regression models with jumps is very necessary.

A non parametric regression model with jumps can be described as follows

$$y = f(x) + \varepsilon, \tag{1}$$

where x is the explanatory variable, ε is the stochastic disturbance with $E(\varepsilon|x) = 0$ and $Var(\varepsilon|x) = \sigma^2$, and $f(x)$ is continuous in the design interval $[a, b]$ except for q positions $a < s_1 < s_2 < \cdots < s_q < b$, called jumps, with magnitudes $d_j \neq 0$ for $j = 1, 2, \ldots, q$.

CONTACT Xiu-Li Du duxiuli@njnu.edu.cn College of Mathematical Sciences, Nanjing Normal University, 210023, China.
Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/lsta.

Without loss of generality, $f(x)$ is assumed to be right-continuous at all jump positions. Throughout this article, the number and positions of jumps are all unknown in advance.

Statistical inference about regression models with jumps may be first studied by McDonald and Owen (1986). Under the assumption of piecewise smoothing, at any given point, various linear fits are calculated under different window sizes and orientations. The final estimator is obtained by weighted average of these linear fits where the weights are determined by their pseudo-mean squared errors. Afterward, Hall and Titterington (1992) suggested an alternative method by establishing some relations among three local linear smoothers to detect jumps; the regression curve is then fitted as usual in design subintervals separated by the detected jumps. The idea of constructing three estimates of the regression function on any given point based on the observations on the left side, the right side, and both sides of the given point, respectively, laid the foundation for estimation methods of discontinuous non parametric regression models.

Besides the piecewise estimation methods above, some scholars proposed the local polynomial and kernel-type methods for jump-detection and jump-preserving estimation. Qiu (1994) proposed an estimator of the number of jumps of discontinuous regression functions. They calculated the difference of left and right one-sided kernel functions which took non zero values apart from the original point for a distance. A test statistic is presented based on the difference. Qiu and Yandell (1998) suggested a jump-detection algorithm based on the local polynomial fitting to detect jumps in regression function and its derivatives. Qiu (2003) proposed a jump-preserving curve fitting procedure based on the local linear kernel estimation. For each point, two one-sided local linear estimates are considered, and based on the comparison of the weighted residual mean squares of the two one-sided fits, the curve estimate at each point is obtained by one of the two estimates or their average. The resulting estimates preserve jumps quite well, but show a rough behavior in continuity regions of the underlying regression curve. To overcome this disadvantage, Gijbels, Lambert, and Qiu (2007) proposed a compromise local linear jump-preserving method (denoted as CLLJP in our article). The two one-sided local linear estimates mentioned above along with the two-sided local linear estimate are used to fit the regression curve by comparing the weighted residual mean squares of them. The resulting estimates preserve the jumps well and also give smooth estimates of the continuity parts of the curve. But a threshold parameter is introduced in the procedure, and the choice for it may increase the computation burden. For this reason, Qiu (2009) presented another method to distinguish smooth regions and discontinuous regions based on the fact that the variance of the two-sided estimator is about twice as that of the one-sided estimators. The method does not need to compute the threshold parameter, hence is easier to implement. At the same time, Joo and Qiu (2009) presented a jump-detection procedure by combining all useful information about the jumps in both the first-order and the second-order derivatives of the regression curve. Xia and Qiu (2015) suggested a jump information criterion to estimate the discontinuous curve when the number of jumps is unknown. By minimizing the criterion function which consists of a term measuring the fidelity of the estimated regression curve to the observed data and a penalty with respect to the number of jumps and the jump magnitudes, the number of jumps is obtained. Recently, Zhao et al. (2016) presented an adaptive jump-preserving estimation procedure to estimate varying-coefficient models with discontinuous regression coefficients. They proposed a new estimator based on three types of local linear kernel estimation as in Gijbels, Lambert, and Qiu (2007), and the resulting estimator can automatically preserve the possible jumps of the coefficient functions without knowing the number and locations of jump points.

The local polynomial estimation procedure is a pointwise estimation procedure, which inevitably has the huge computation burden. However, as pointed out by Jupp (1978), the

spline method developed in recent decades has the advantage of fast computation, less oscillation, and more smoothness. Therefore, related research on spline fit for regression functions with jumps has also been carried out by many scholars. Koo (1997) pointed out that if we fit a regression function by a continuous linear spline around a discontinuous point, local bias of the estimate could be unacceptably large and thus there could be an overshoot near the jump. Therefore, a discontinuous spline is appropriate to detect jumps. Moreover, an automatic procedure involving the least-squares method, stepwise knot addition, stepwise basis deletion, knot-merging, and Bayesian information criterion (BIC) is proposed to select the final model. However, the procedure above which selects knots through deterministic optimization techniques could not achieve a global optimal knots. Miyata and Shen (2003) proposed a function estimation procedure using free-knot B-splines with variable multiple knots, where locating the optimal knots is implemented by an adaptive model selection scheme. The proposed procedure flexibly adapts to many complicated structures, such as the discontinuous regression function. Evolutionary algorithms are also considered which can yield the global optimal knots. However, the disadvantage of evolutionary algorithms is the huge computation. Yang and Song (2014) proposed to use B-spline to estimate locations and sizes of jumps in non parametric time-series models. A test statistic based on the smoothing B-spline fit is constructed to detect jumps.

In this article, motivated by Gijbels, Lambert, and Qiu (2007) and the advantage of B-spline on computation, we propose a new jump-detecting and curve estimation method based on piecewise B-spline. First, regression curve is fitted by the B-spline (denoted by CB) function. Second, the whole support interval $[a, b]$ is divided into two subintervals by any given point $x_0 \in (a, b)$, namely $[a, x_0)$ and $[x_0, b]$, then we fit the regression function by the B-spline on the two intervals, respectively. Third, jump points are detected by comparing the residual sum of squares of the two B-spline fits. Subsequently, regression functions with jumps can be fitted by the piecewise B-spline.

The rest of the article is organized as follows. In Section 2, we introduce the definition and basic properties of B-spline. In Section 3, the jump-detection and curve estimation algorithm based on B-spline is proposed. Section 4 gives some theoretical results about our estimator. Numerical simulations are conducted in Section 5. A real data analysis is given in Section 6. Section 7 gives a brief discussion of our method. Proofs can be found in the Appendix.

2. Preliminaries

Assume that $\mathbf{U} = (u_1, \ldots, u_k)$ $(u_0 = a < u_1 < \cdots < u_k < b = u_{k+1})$ is the internal knot sequence on support $[a, b]$, then a space of B-spline functions with degree p can be described as follows:

$$S(p, \mathbf{U}) = \{s \in C^{p-1}[a, b] : s \text{ is a polynomial of degree } p \text{ on each subinterval } [u_i, u_{i+1}] \text{ and has } (p-1)\text{th-ordered continuous derivative at the knot } u_i\},$$

where $C^{p-1}[a, b]$ is the set of functions with $(p-1)$th-ordered continuous derivatives on interval $[a, b]$.

Definition 1 (de Boor (1978)). Assume that $\mathbf{U} = (u_1, \ldots, u_k)$ $(u_0 = a < u_1 < \cdots < u_k < b = u_{k+1})$ is the internal knot sequence on support $[a, b]$, $U^* = (u_0^*, u_1^*, \ldots, u_{k+2p+1}^*)$ $(where\, u_0^* = \cdots = u_p^* = a, u_{p+1}^* = u_1, \ldots, u_{p+k}^* = u_k, u_{k+p+1}^* = \cdots = u_{k+2p+1}^* = b)$ is called the extended knot sequence, then the normalized B-spline basis functions of degree p on

$S(p, \mathbf{U})$ are defined as follows:

$$B_{i,p}(x) = \begin{cases} (-1)^{p+1}(u^*_{i+p+1} - u^*_i)[u^*_i, \ldots, u^*_{i+p+1}](x-u)^p_+, & \text{if } u^*_i < u^*_{i+p} \\ 0, & \text{otherwise} \end{cases}$$

where $(x-u)^p_+ = \max\{0, (x-u)^p\}$ and $[u^*_i, \ldots, u^*_{i+p+1}](x-u)^p_+$ is the pth-order divided difference of $(x-u)^p_+$, $i = 0, 1, \ldots, p+k$.

The recursive form of B-spline basis functions can be easily obtained from the above definition, i.e.,

$$B_{i,0}(x) = \begin{cases} 1, & x \in [u^*_i, u^*_{i+1}) \\ 0, & \text{otherwise} \end{cases}$$

$$B_{i,p}(x) = \frac{x - u^*_i}{u^*_{i+p} - u^*_i} B_{i,p-1}(x) + \frac{u^*_{i+p+1} - x}{u^*_{i+p+1} - u^*_{i+1}} B_{i+1,p-1}(x),$$

where $[u^*_i, u^*_{i+1})$ is the ith knot interval, $i = 0, 1, \ldots, p+k$, see Schumaker (2007).

Here we list some important properties of B-spline basis functions.

1. $0 < B_{i,p}(x) \le 1$ for $x \in [u^*_i, u^*_{i+p+1})$, otherwise $B_{i,p}(x) = 0$;
2. The number of non zero B-spline basis functions is at most $p+1$ in the interval $[u_i, u_{i+1})$ for any $0 \le i \le k$, and these non zero B-spline basis functions are $B_{i,p}(x), \ldots, B_{i+p,p}(x)$;
3. $\sum_{j=i}^{i+p} B_{j,p}(x) = 1$, where $x \in [u_i, u_{i+1})$ for any $0 \le i \le k$;
4. $B_{i,p}(x)$ has $(p-r)$-ordered derivative at a knot with multiplicity r;
5. $\int_a^b B_{i,p}(x)\mathrm{d}x = \dfrac{u^*_{i+p+1} - u^*_i}{p+1}$.

Remark 1. From Property 4, we know that increasing degree will improve the continuity of B-spline functions, while increasing multiplicity will reduce the continuity.

From Definition 1, for any $s(x) \in S(p, \mathbf{U})$, there exists $\boldsymbol{\alpha} \in R^{p+k+1}$ such that

$$s(x) = \mathbf{B}^T(x)\boldsymbol{\alpha},$$

where $\mathbf{B}(x) = (B_{0,p}(x), \ldots, B_{p+k,p}(x))^T$.

The following theorem shows the bias of B-spline estimator which will be used in the proof of the subsequent theorems.

Theorem 1 (Barrow and Smith (1978)). *Let $f(x)$ be a pth-ordered smooth function on $[a, b]$, then*

$$\inf_{s(x)\in S(p,\mathbf{U})} \| f(x) + b(x) - s(x) \|_{L_\infty} = o(h^{p+1}), \tag{2}$$

where $\|\cdot\|_{L_\infty}$ is the infinity norm, $h_i = u_i - u_{i-1}$, and $h = max_{1\le i\le k+1} h_i$. And $b(x) = -\frac{f^{(p+1)}(u_i)h_i^{p+1}}{(p+1)!} B_{p+1}\left(\frac{x-u_i}{h_i}\right)$, in which $B_{p+1}(\cdot)$ is the $(p+1)$th Bernoulli polynomial which is inductively defined as,

$$B_0(x) = 1, \quad B_i(x) = \int_0^x iB_{i-1}(z)dz + b_i,$$

where $b_i = -i\int_0^1 \int_0^x B_{i-1}(z)dzdx$ is the ith Bernoulli number.

Remark 2. We can see from Theorem 1 that $b(x) = O(h^{p+1})$.

3. Jump-detection and curve estimation method based on piecewise B-splines

3.1. Basic idea

Assume that $\{(x_i, y_i), i = 1, \ldots, n\}$ are observed data from the non parametric regression model (1). To explain our basic idea and theorems below clearly, we first introduce the following conditions.

C1. k denotes the number of internal knots. Let $h_i = u_i - u_{i-1}$, $h = \max_{1\le i\le k+1} h_i$, and assume

$$\max_{1\le i\le k} |h_{i+1} - h_i| = o\left(\frac{1}{k}\right) \quad \text{and} \quad \frac{h}{\min_{1\le i\le k+1} h_i} \le M_1, \tag{3}$$

where $M_1 > 0$ is a constant. The assumption means that the knot sequence approximately meets the quasi-uniform condition.

C2.

$$\sup_{x\in[a,b]} |Q_n(x) - Q(x)| = o\left(\frac{1}{k}\right), \tag{4}$$

where $Q_n(x)$ and $Q(x)$ represent the empirical distribution function and theoretical distribution function with a positive continuous density $q(x)$ of explanatory variable x, respectively.

C3. The distance of adjacent jump points must be larger than $(2p + 1)h$.

C4. The number of interior knots $k = O(n^{1/(2p+1)})$, i.e., $c_k n^{1/(2p+1)} \le k \le C_k n^{1/(2p+1)}$ for some positive constants c_k and C_k.

When the knot sequence is quasi-uniform, the B-spline estimator with degree p for $f(x)$ can be obtained by minimizing the following objective function

$$\sum_{i=1}^{n} \left(y_i - \sum_{j=0}^{k+p} \alpha_j B_{j,p}(x_i) \right)^2 \tag{5}$$

with respect to α_j, $j = 0, 1, \ldots, k + p$. Denote the minimizer as $\hat{f}(x; U^*)$, then

$$\hat{f}(x; U^*) = \mathbf{B}^T(x)(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{Y},$$

where $\mathbf{M} = (\mathbf{B}(x_1), \ldots, \mathbf{B}(x_n))^T$, $\mathbf{Y} = (y_1, y_2, \cdots, y_n)^T$, $\mathbf{B}(x)$ is as in Section 2. Denote the residual sum of square as RSS_0, namely $RSS_0 = \sum_{i=1}^{n}(y_i - \hat{f}(x_i; U^*))^2$, obviously $RSS_0 = \mathbf{Y}^T(\mathbf{I} - \mathbf{M}(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T)\mathbf{Y}$.

On the other hand, if we add a new knot $x_0 \in (a, b)$ with multiplicity $p + 1$ in U^*, without loss of generalization, we assume that $x_0 \in (u_i, u_{i+1})$, then the new knot sequence can be denoted by $\tilde{U}^*$, namely,

$$\tilde{U}^* = (\underbrace{a, \ldots, a}_{p+1}, u_1, \ldots, u_i, \underbrace{x_0, \ldots, x_0}_{p+1}, u_{i+1}, \ldots, \underbrace{b, \ldots, b}_{p+1}).$$

Then, $\hat{f}(x; \tilde{U}^*)$ can be computed by minimizing the following objective function

$$\sum_{x_i<x_0} \left(y_i - \sum_{j=0}^{i+p} \alpha_j \tilde{B}_{j,p}(x_i) \right)^2 + \sum_{x_i\ge x_0} \left(y_i - \sum_{j=i+p+1}^{k+2p+1} \alpha_j \tilde{B}_{j,p}(x_i) \right)^2, \tag{6}$$

with respect to α_j, $j = 0, 1, \ldots, k + 2p + 1$, where $\tilde{B}_{j,p}(x)$ is the B-spline basis function under the knot sequence $\tilde{U}^*$. Let $\tilde{\mathbf{M}} = (\tilde{\mathbf{B}}(x_1), \ldots, \tilde{\mathbf{B}}(x_n))^{\mathbf{T}}$, where $\tilde{\mathbf{B}}(x) = (\tilde{B}_{0,p}(x), \ldots, \tilde{B}_{2p+k+1,p}(x))^T$, then $\hat{f}(x; \tilde{U}^*) = \tilde{\mathbf{B}}^T(x)(\tilde{\mathbf{M}}^T\tilde{\mathbf{M}})^{-1}\tilde{\mathbf{M}}^T\mathbf{Y}$.

If we check (6) carefully, we can see that the residual sum of square has been separated into two parts after inserting a new knot x_0 with multiplicity $p + 1$, i.e.,

$$RSS_l(x_0) = \sum_{x_i \in [a, x_0)} (y_i - \hat{f}(x_i; \tilde{U}^*))^2;$$
$$RSS_r(x_0) = \sum_{x_i \in [x_0, b]} (y_i - \hat{f}(x_i; \tilde{U}^*))^2.$$

Define the difference function of residual sum of square as follows:

$$DRSS(x_0) = RSS_0 - (RSS_l(x_0) + RSS_r(x_0)),\ x_0 \in (a, b).$$

After some calculation, we can easily get $DRSS(x_0) = \mathbf{Y}^T(P_{\tilde{\mathbf{M}}} - P_{\mathbf{M}})\mathbf{Y}$, where $P_{\mathbf{M}} = \mathbf{M}(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T$ is the projection matrix of $\mathbf{M}$, $P_{\tilde{\mathbf{M}}} = \tilde{\mathbf{M}}(\tilde{\mathbf{M}}^T\tilde{\mathbf{M}})^{-1}\tilde{\mathbf{M}}^T$. Obviously, $P_{\mathbf{M}}$ and $P_{\tilde{\mathbf{M}}}$ are both the symmetric idempotent matrices.

Theorem 2 gives the distribution of $DRSS(x_0)$.

Theorem 2. *Under the assumptions C1–C2, for any $x_0 \in (a, b)$, we have the following conclusions.*

(i) If x_0 is not any internal knots, then

$$DRSS(x_0)/\sigma^2 \sim \chi^2_{p+1}(\lambda_{x_0}), \tag{7}$$

where $\lambda_{x_0} = f(\mathbf{X})^T(P_{\tilde{\mathbf{M}}} - P_{\mathbf{M}})f(\mathbf{X})/\sigma^2$ is the non centrality parameter of the Chi-squared distribution, $\mathbf{X} = (x_1, \ldots, x_n)^T$.

(ii) If x_0 is one of the internal knots, then

$$DRSS(x_0)/\sigma^2 \sim \chi^2_p(\lambda_{x_0}), \tag{8}$$

where λ_{x_0} is the same as in (i).

From Theorem 1, Koo (1997), and Zhou and Shen (2001), we can see that if $f(x)$ is smooth on support $[a, b]$, then at each design point $\hat{f}(x; U^*)$ is a consistent estimator of $f(x)$; if $f(x)$ has jump points on $[a, b]$, then $\hat{f}(x; U^*)$ is not a consistent estimator in the neighborhood of jump points. $\hat{f}(x; \tilde{U}^*)$ has the same property on intervals $[a, x_0)$ and $[x_0, b]$, see also Figure 1 for a demonstration. From Figure 1, we can see intuitively that when x_0 lies in a continuous region away from jumps of regression curve, for example $x_0 = 0.3$, $\hat{f}(x; \tilde{U}^*)$ makes little difference from $\hat{f}(x; U^*)$ in the intervals both $[a, x_0)$ and $[x_0, b]$, including at the neighborhood of jump points, so $RSS_l(x_0) + RSS_r(x_0)$ is close to RSS_0; thus $DRSS(x_0)$ is small. However, if x_0 is in the neighborhood of a jump point, for example $x_0 = 0.49$, $\hat{f}(x; \tilde{U}^*)$ is consistent only on left side of the neighborhood of x_0 and inconsistent on the right neighborhood, while $\hat{f}(x; U^*)$ is inconsistent on both sides of the neighborhood of x_0; therefore, $DRSS(x_0)$ becomes large when x_0 is close to a jump point. Specifically, if x_0 coincides with one of jump points, $DRSS(x_0)$ should get its local maxima.

In summary, the $DRSS(x_0)$ is large and has a local maxima when there exists a jump point in the neighborhood of x_0, otherwise it is small. Hence, the statistic $\{DRSS(x_0)\}$ can be used to detect the possible jumps in the regression function. From Theorem 2, $DRSS(x_0)/\sigma^2 \sim$

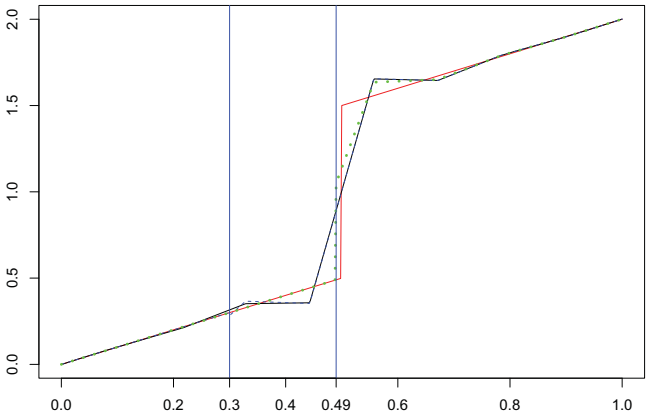

Figure 1. A function curve with a jump at 0.5 (red solid line) and three fitted curves. The black solid line denotes the fitted curve of continuous B-spline. The blue dash line and green dash line represent the fitted curves of B-spline estimator inserting 0.3 and 0.49 into the knot sequence U^*, respectively.

$\chi^2_{p+1}(\lambda_{x_0})$ for $x_0 \in (a, b)$ except internal knots, hence when $DRSS(x_0)/\sigma^2 > \chi^2_{1-\alpha,p+1}(\lambda_{x_0})$ (where α is the significance level), x_0 can be thought as a jump point.

In fact, the non centrality parameter $\lambda_{x_0} = f(\mathbf{X})^T(P_{\tilde{\mathbf{M}}} - P_{\mathbf{M}})f(\mathbf{X})/\sigma^2$ depends on $f(x)$, hence is unknown. Therefore, in order to detect the jumps, we first need to estimate λ_{x_0}. Under the null hypothesis that $f(x)$ is continuous, from the definition of $DRSS(x_0)$, Theorem 2, and Theorem 1 in Zhou and Shen (2001), we can infer that in the continuous region except internal knots, $\{DRSS(x_0)\}$ are almost the same, and hence have the same Chi-squared distribution with the non centrality parameter λ; at the same time, we know that $E\left(\frac{DRSS(x_0)}{\sigma^2}\right) = p + 1 + \lambda$. Therefore, the non centrality parameter can be estimated by

$$\lambda = \frac{\text{mean of } \{DRSS(x_0), x_0 \in (a, b)\backslash\{u_1, u_2, \ldots, u_k\}\}}{\sigma^2} - p - 1.$$

Based on this, we present a jump-detecting procedure one by one based on B-spline. If $DRSS(x_0) \le \chi^2_{1-\alpha,p+1}(\lambda)$, $f(x)$ has no jumps. Otherwise, we choose the maximum point of $DRSS(x_0)$ as a new jump and insert it into the knot sequence with multiplicity $p + 1$. With the new knot sequence, we again calculate new $\{DRSS(x_0)\}$ and new non centrality parameter and repeat the hypothesis tests until $DRSS(x_0) \le \chi^2_{1-\alpha,p+1}(\lambda)$ or the new jump is in the neighborhood of one of the detected jumps.

3.2. Jump-detection and curve estimation algorithms I

Based on the above analysis, we can find all the jumps step by step by hypothetical test, then obtain the final estimator of $f(x)$ with the final knot sequence based on the piecewise B-spline. The jump-detection and curve estimation procedure based on B-splines (Denote as JDBS-I) is summarized in the following algorithm.

Step 1. Fit model (1) by the B-spine function with the knot sequence U^* on the whole interval $[a, b]$, and calculate the residual sum of squares RSS_0.

Step 2. Divide the interval $[a, b]$ by any point x_0 into $[a, x_0)$ and $[x_0, b]$, then fit the regression function by the B-spline function on the two intervals, respectively, to obtain $RSS_l(x_0)$ and $RSS_r(x_0)$, and finally calculate out $DRSS(x_0)$ for any $x_0 \in (a, b)$.

Step 3. Calculate the mean of $\{DRSS(x_0), x_0 \in (a, b)\backslash\{u_1, \ldots, u_k\}\}$ and denote as $\overline{DRSS}$.

Step 4. Compare the values of $DRSS(x_0)$ with the threshold value $\chi^2_{1-\alpha,p+1}(\lambda)$, where $\lambda = \overline{DRSS}/\hat{\sigma}^2 - p - 1$:
  (i) if $DRSS(x_0)/\hat{\sigma}^2 \le \chi^2_{1-\alpha,p+1}(\lambda)$, no more jumps could be detected;
  (ii) if $DRSS(x_0)/\hat{\sigma}^2 > \chi^2_{1-\alpha,p+1}(\lambda)$, then $f(x)$ contains at least one jump. Choose the maximum point of $DRSS(x_0)$ as the new jump. If the new jump is in the neighborhood of any detected jumps, stop checking; otherwise, insert the new jump point into the knot sequence U^* with multiplicity $p+1$. Denote the new knot sequence by U^* without confusion.

Step 5. Repeat Steps 1–4, until no more jumps are detected.

Step 6. Fit model(1) by the piecewise B-spline function with the final knot sequence.

Remark 3. In Step 4, we need to estimate σ^2. When $f(x)$ is smooth, a used consistent estimator of σ^2 is (see Rice 1984)

$$\hat{\sigma}^2 = \frac{1}{n-1}\sum_{i=1}^{n-1}(y_{i+1} - y_i)^2/2.$$

Since the number of jump points is very sparse in our model, which will not have much impact on estimating σ^2; therefore we still adopt the form above which is consistent to estimate σ^2.

3.3. Jump-detection and curve estimation algorithms II

In the above algorithm, the parameter λ is estimated by averaging $DRSS(x_0)$ on (a, b) except the internal knots but including the jump points. As we know, λ_{x_0}'s in the vicinity of the jump points are large, hence the estimated λ can be larger than true λ in continuous region, which will lead to the low power in detecting jumps. For this reason, we present another estimation method for λ.

For any $x_0 \in (a, b)$, let A_{x_0} denote a neighborhood of x_0, then $DRSS(x_0)$ can be divided into two parts: the difference of the residual sum of square on A_{x_0}, denote as $DRSS_{A_{x_0}}(x_0)$, and this one on $(a, b)\backslash A_{x_0}$, denote as $DRSS(x_0) - DRSS_{A_{x_0}}(x_0)$. When x_0 lies on a continuous region, both $DRSS_{A_{x_0}}(x_0)$ and $DRSS(x_0) - DRSS_{A_{x_0}}(x)$ are approximately similar to $DRSS(x_0)$; hence $DRSS(x_0)$ can be estimated by $\frac{n}{n-n_{A_{x_0}}}(DRSS(x_0) - DRSS_{A_{x_0}}(x_0))$, where $n_{A_{x_0}}$ represents the sample size on A_{x_0}. When x_0 lies on a neighborhood of some jump point, $DRSS_{A_{x_0}}(x_0)$ contains the information caused by the jump point; therefore if we cut A_{x_0} out of (a, b), then $\frac{n}{n-n_{A_{x_0}}}(DRSS(x_0) - DRSS_{A_{x_0}}(x_0))$ can be seen as the estimator of $DRSS(x_0)$ under the assumption that x_0 lies on a continuous region.

In summary, given $x_0 \in (a, b)$, we first calculate the difference of the residual sum of square away from its neighborhood, i.e., $DRSS(x_0) - DRSS_{A_{x_0}}(x_0)$, then obtain the estimate of $DRSS(x_0)$ by $DRSS'(x_0) = \frac{n}{n-n_{A_{x_0}}}(DRSS(x_0) - DRSS_{A_{x_0}}(x_0))$, subsequently calculate the mean of $\{DRSS'(x_0), x_0 \in (a, b)\}$, denoted by $\overline{DRSS'}$. Finally, the critical value is determined by $\chi^2_{1-\alpha,p+1}(\lambda')$, where $\lambda' = \overline{DRSS'}/\hat{\sigma}^2 - p - 1$.

From the analysis above, we can see that the new non centrality parameter λ' is obviously smaller than the λ in JDBS-I method. Hence, it can improve the power in detecting jumps. However, it can produce more fault jumps. In the following, we denote the jump-detection and curve estimation procedure based on λ' as JDBS-II method.

3.4. Parameter selection

During the process of fitting regression function by B-spline function, there are two parameters to be chosen: the number of internal knots k and degree p. Degree p reflects the smoothing performance of the B-spline function, in practice $p \leq 3$ is enough. The optimal order of k, by condition C4, is $O(n^{1/(2p+1)})$. In practice, we select number of knots $C_p n^{1/(2p+1)}$ with a relatively large positive constant C_p to deliberately undersmooth the data to capture possible jump points. In our numerical analysis, we take C_p in a range of [5, 10] and we optimize k from $[5n^{1/(2p+1)}, \min(10n^{1/(2p+1)}, n/2 - p - 1)]$ through the following BIC:

$$\mathrm{BIC}(k, p) = \log(RSS_0/n) + (k + p) * \log(n)/n. \tag{9}$$

4. Asymptotic properties

The asymptotic properties of the estimators are given by the following theorem.

Theorem 3. *Under the conditions C1–C3 and $k = o(n^r)$, for some $r \in (0, 1/2]$, then for any $x \in [a, b]$,*

$$\mathrm{E}(\hat{f}(x; \tilde{U}^*)) - f(x) = b(x) + o_p(h^{p+1}), \tag{10}$$

$$\mathrm{Var}(\hat{f}(x; \tilde{U}^*)) = \frac{\sigma^2}{n}\tilde{\mathbf{B}}^T(x)G^{-1}(q)\tilde{\mathbf{B}}(x) + o_p((nh)^{-1}), \tag{11}$$

where $b(x)$ is the same as in Theorem 1 and $G(q) = \int_a^b \tilde{\mathbf{B}}(x)\tilde{\mathbf{B}}^T(x)q(x)dx$.

Remark 4. From Theorem 3 and Remark 2, we know that $\hat{f}(x; \tilde{U}^*)$ is the consistent estimator of $f(x)$ for any $x \in [a, b]$, which means the proposed estimator can preserve the jumps well and perform good on the smoothing area as well.

Theorem 4. *In addition to the assumptions in Theorem 3, suppose that $k \geq Cn^{1/(2p+1)}$ for some constant $C > 0$. Then for any fixed $x \in [a, b]$,*

$$\frac{\hat{f}(x; \tilde{U}^*) - (f(x) + b(x))}{\sqrt{\mathrm{Var}(\hat{f}(x; \tilde{U}^*))}} \xrightarrow{D} N(0, 1), \tag{12}$$

From Theorem 3 and Theorem 4, we can construct the confidence regions for $f(x)$ when the variance σ^2 is known.

Theorem 5. *In addition to the assumptions in Theorem 4, suppose that the $kn^{-1/(2p+1)} \to \infty$. Then for any fixed $x \in [a, b]$, the $100(1 - \alpha)\%$ asymptotic confidence interval for $f(x)$ is*

$$\hat{f}(x; \tilde{U}^*) \pm z_{\alpha/2}\sqrt{\mathrm{Var}(\hat{f}(x; \tilde{U}^*))},$$

where $z_{\alpha/2}$ is the $(1 - \alpha/2)$th normal percentile and

$$\mathrm{Var}(\hat{f}(x; \tilde{U}^*)) = \sigma^2\tilde{\mathbf{B}}^T(x)G^{-1}\tilde{\mathbf{B}}(x)/n$$

with $G = n\tilde{\mathbf{X}}\tilde{\mathbf{X}}'$ and $\tilde{\mathbf{X}} = n^{-1}(\tilde{\mathbf{B}}(x_1), \ldots, \tilde{\mathbf{B}}(x_n))$.

In our procedure, we estimate $f(x)$ by the piecewise B-spline function, namely in subintervals $[a, s_1), \ldots, [s_q, b]$ (which are constructed by jump points), $f(x)$ is estimated by the ordinary B-spline function. Therefore, proofs of Theorems 3–5 are similar to the case of B-spline estimator, so we omit them here.

5. Numerical simulations

In this section, we will carry out numerical simulations to investigate the performance of our JDBS procedures. The following three examples are considered.

$$\text{Example 1.}\quad f_1(x) = \begin{cases} -3x + 2, & 0 \le x < 0.3; \\ -3x + 3 - \sin(\frac{(x-0.3)\pi}{0.2}), & 0.3 \le x < 0.7; \\ x/2 + 1.55, & 0.7 \le x \le 1. \end{cases}$$

$$\text{Example 2.}\quad f_2(x) = \begin{cases} 0, & 0 \le x < 0.3; \\ 3x^2 + 0.93, & 0.3 \le x < 0.7; \\ 4x^2 + 1.24, & 0.7 \le x \le 1. \end{cases}$$

$$\text{Example 3.}\quad f_3(x) = \begin{cases} \cos(8\pi(0.5 - x)), & 0 \le x < 0.5; \\ -\cos(8\pi(0.5 - x)), & 0.5 \le x \le 1. \end{cases}$$

In order to evaluate the asymptotic properties of the estimators, three sample sizes $n = 200$, 300, and 500 are considered in each example. Furthermore, to show the influence of noise, three different $\sigma = 0.1, 0.2$, and 0.4 are considered. Data are generated from model (1) when $\varepsilon \sim N(0, \sigma^2)$. The design points $\{x_i, i = 1, \ldots, n\}$ are sampled from a uniform distribution on [0, 1]. And we perform 1000 replicated simulations in each case. In order to show the performance of JDBS procedures on the estimation of discontinuous function, we also give the related results based on the B-spline procedure (denoted as CB) and results based on the CLLJP procedure.

In order to evaluate the performance of our methods, the mean integrated squared error (MISE) is introduced, which is given by

$$\text{MISE} = \frac{1}{N}\sum_{l=1}^{N} \text{ISE}_l, \quad \text{with} \quad \text{ISE}_l = \frac{1}{n}\sum_{i=1}^{n} (\hat{f}_l(x_i) - f(x_i))^2,$$

where N denotes simulation times. $\{\hat{f}_l(x_i), i = 1, \ldots, n\}$ are the estimate results of the lth simulated sample.

In the neighborhood of a jump point, the approximated local MISE (LMISE) is given by

$$\text{MISE}_{\underline{u}_j} = \frac{1}{N}\sum_{l=1}^{N} \left(\frac{1}{n}\sum_{i=1}^{n} (\hat{f}_l(x_i) - f(x_i))^2 \text{I}[\underline{u}_j - 0.05 < x_i < \underline{u}_j + 0.05] \right),$$

where $\underline{u}_j$ denotes the jth jump point and $\text{I}(\cdot)$ is an indicator function.

In Example 1, $f_1(x)$ has two jump points at $x = 0.3$ and $x = 0.7$ with the same jump magnitude 1. The results of jump detection using JDBS-I and JDBS-II procedures can be seen from Tables 1 and 2. We can learn from these tables that for fixed noise level, the bias and standard deviation decline with the increase of sample size n; the power of detecting jumps and the frequency of detecting exactly two jumps also increase when n becomes large. On the other hand, if n is fixed, the performance of JDBS-I and JDBS-II procedures becomes less effective with the increase of σ. When σ is small, the frequency of detecting exactly two jumps is large, otherwise it decreases, which implies that the noise has an important effect on jump detecting. From Tables 1 and 2, we can also see that more than two jumps are possibly detected by the JDBS methods, and JDBS-II procedure usually detects more jumps than JDBS-I procedure, but the MISE and LMISE values are almost the same (see Table 4).

In Xia and Qiu (2015), they also carried out the simulations of Example 1 in three different cases. But none of them is exactly the same as our case because the design points in our

Table 1. Bias, standard deviation, detecting power, and frequency of jump points based on JDBS-I procedure when $f = f_1, N = 1000$, and $\varepsilon \sim N(0, \sigma^2)$.

						Frequency				
σ	n	Jumps	Bias	sd	Power	0	1	2	3	>3
0.1	200	0.3	0.0001	0.0050	0.898	1	210	763	25	1
		0.7	0.0002	0.0049	0.887					
	300	0.3	−0.0003	0.0034	0.993	0	24	895	70	11
		0.7	0.0002	0.0035	0.982					
	500	0.3	0.0000	0.0018	1	0	0	789	165	46
		0.7	0.0000	0.0020	1					
0.2	200	0.3	−0.0015	0.0087	0.740	67	355	465	96	17
		0.7	0.0009	0.0066	0.696					
	300	0.3	−0.0005	0.0044	0.889	8	183	626	144	39
		0.7	0.0004	0.0052	0.886					
	500	0.3	−0.0002	0.0025	0.984	0	28	721	208	43
		0.7	0.0002	0.0028	0.988					
0.4	200	0.3	−0.0055	0.0149	0.313	354	377	204	55	10
		0.7	0.0039	0.0162	0.267					
	300	0.3	−0.0022	0.0123	0.486	188	404	273	105	30
		0.7	0.0027	0.0134	0.467					
	500	0.3	−0.0020	0.0095	0.696	65	323	432	150	30
		0.7	0.0021	0.0092	0.674					

simulations are random. The results of Xia and Qiu (2015) and our JDBS methods are summarized in Table 3. The percentage of $\hat{m} = m_0$ in our procedures is smaller than the results in JIC method of Xia and Qiu (2015) and BIC method. It may be partly due to the fact of random design. On the other hand, our methods tend to detect more jumps. Maybe we could add a penalty like in Xia and Qiu (2015) to balance with the number of detected jumps and curve estimation in our future work.

In order to further study the performance of our procedure, we compute the MISE and LMISE ($= \text{MISE}_{0.3} + \text{MISE}_{0.7}$). We also give the related results based on CB procedure and CLLJP procedure, which are presented in Table 4. From Table 4, it can be seen that for a fixed sample size n and a small σ, CB has the biggest MISE and LMISE, while the MISE and LMISE

Table 2. Bias, standard deviation, detecting power, and frequency of jump points based on JDBS-II procedure when $f = f_1, N = 1000$, and $\varepsilon \sim N(0, \sigma^2)$.

						Frequency				
σ	n	Jumps	Bias	sd	Power	0	1	2	3	>3
0.1	200	0.3	−0.0005	0.0061	0.983	0	49	879	70	2
		0.7	0.0004	0.0056	0.963					
	300	0.3	−0.0001	0.0033	0.994	0	10	926	64	0
		0.7	0.0002	0.0036	0.993					
	500	0.3	−0.0001	0.0019	1	0	0	968	32	0
		0.7	0.0001	0.0017	1					
0.2	200	0.3	−0.0013	0.0089	0.936	0	38	514	343	105
		0.7	0.0013	0.0091	0.919					
	300	0.3	−0.0004	0.0055	0.978	0	5	472	412	111
		0.7	0.0007	0.0059	0.977					
	500	0.3	−0.0001	0.0022	0.982	0	25	775	185	15
		0.7	0.0003	0.0033	0.981					
0.4	200	0.3	−0.0024	0.0166	0.613	8	150	375	340	127
		0.7	0.0027	0.0183	0.603					
	300	0.3	−0.0032	0.0141	0.783	2	66	331	384	217
		0.7	0.0021	0.0136	0.725					
	500	0.3	−0.0022	0.0104	0.907	0	26	373	398	203
		0.7	0.0013	0.0098	0.889					

Table 3. Percentages of $\hat{m}$ values[1] obtained by different methods which are equal to, smaller than, or lager than m_0 value[2] in 1000 replicated simulations.

Method	$\hat{m}$	$n = 200$	$n = 500$	Method	$n = 200$	$n = 500$
JIC (m)	$> m_0$	2.8	0.0	BIC(m)	31.8	10.8
Small penalty	$= m_0$	96.4	100.0		68.2	89.
	$< m_0$	0.8	0.0		0.0	0.0
JIC (m)	$> m_0$	0.0	0	Wavelets	0.0	0.0
Moderate penalty	$= m_0$	96.6	100		16.0	88.0
	$< m_0$	3.4	0		84.0	12.0
JIC (m)	$> m_0$	0.0	0.0			
Large penalty	$= m_0$	54.0	94.4			
	$< m_0$	46.0	5.6			
JDBS-I	$> m_0$	11.3	25.1	JDBS-II	44.8	20.0
	$= m_0$	46.5	72.1		51.4	77.5
	$< m_0$	42.2	2.8		3.8	2.5

[1] $\hat{m}$ denotes the number of detected jumps.
[2] m_0 is the real number of jumps, i.e., $m_0 = 2$ in Example 1.

based on the CLLJP and JDBS procedures are much smaller than the CB procedure, and the MISE–LMISE values are very close to each other under three different methods, which shows that the JDBS and CLLJP procedures can preserve jumps well and perform as well as the CB procedure on smoothing areas. The MISE and LMISE of three kinds of procedures are quite close when $\sigma = 0.4$, which shows that the CLLJP and JDBS procedures are easily influenced by noise level. All the MISE and LMISE values based on three kinds of methods decline with the increasing sample size n, which illustrates that the effect of fitted curves becomes more efficient with the increasing sample size n. Further, the MISE and LMISE values of the JDBS procedure are smaller than that of the CLLJP procedure for the same sample size n, which means that the JDBS procedure can preserve jumps better than CLLJP procedure and its global fitting is also superior. Figure 2 shows the fitted results based on CB, CLLJP, and JDBS-I procedures and their 95% pointwise confidence intervals for $n = 200$, 300, and 500. Because MISE and LMISE values of the JDBS-II procedure are close to those of the JDBS-I procedure, in the following, we will not consider the plots of the JDBS-II procedure any more. We can further see that the JDBS procedure works better than CLLJP procedure whether at the jump point or on the smoothing areas.

Besides, we also record the time consuming of each method on an ordinary PC with Intel Core 2 Duo E7500 CPU and 2GB RAM. From Table 5, the computation complexity of the

Table 4. MISE and LMISE when $f = f_1$, $N = 1000$, and $\varepsilon \sim N(0, \sigma^2)$.

		$n = 200$		$n = 300$		$n = 500$	
σ	Method	MISE	LMISE	MISE	LMISE	MISE	LMISE
0.1	CB	0.0074	0.0062	0.0069	0.0060	0.0055	0.0050
	CLLJP	0.0031	0.0019	0.0014	0.0006	0.0009	0.0004
	JDBS-I	0.0019	0.0010	0.0010	0.0004	0.0006	0.0002
	JDBS-II	0.0016	0.0007	0.0010	0.0003	0.0006	0.0002
0.2	CB	0.0130	0.0093	0.0113	0.0087	0.0096	0.0078
	CLLJP	0.0087	0.0046	0.0061	0.0032	0.0037	0.0022
	JDBS-I	0.0067	0.0034	0.0043	0.0020	0.0023	0.0008
	JDBS-II	0.0060	0.0024	0.0040	0.0015	0.0023	0.0008
0.4	CB	0.0289	0.0167	0.0232	0.0140	0.0154	0.0104
	CLLJP	0.0241	0.0123	0.0215	0.0119	0.0133	0.0065
	JDBS-I	0.0226	0.0103	0.0166	0.0082	0.0109	0.0055
	JDBS-II	0.0237	0.0096	0.0167	0.0070	0.0102	0.0044

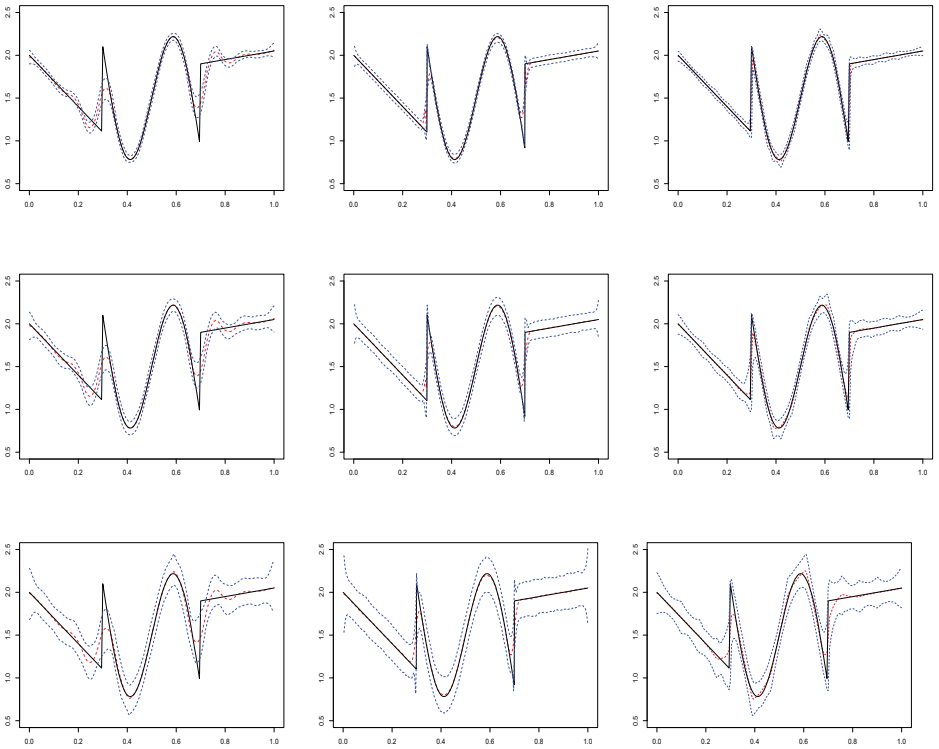

Figure 2. Plot of curve estimate when $f = f_1$, $n = 300$, and $N = 1000$. The three rows of panels, respectively, show the fitted curves when $\sigma = 0.1$, $\sigma = 0.2$, and $\sigma = 0.4$. The three columns of panels, respectively, show fitted curves based on CB, CLLJP, and JDBS-I methods. The solid curve represents the true regression model. The fitted curves and the 95% pointwise confidence intervals are denoted by dotted curves.

CLLJP procedure increases rapidly with increasing sample size n, while our procedure can always maintain a relatively lower level. The time consuming does not change much with different sample size n and noise level σ. The similar results can also be got in the other two examples.

In order to study the effect of the magnitude of jumps on jump detecting, Example 2 is considered here. In Example 2, $f_2(x)$ also has two jump points at $x = 0.3$ and $x = 0.7$, but they have different jump magnitudes 1.2 and 0.8. The simulation results can be seen from Table 6 and 7. Similar conclusions could be obtained as those in Example 1. Furthermore, we can also see that a larger jump magnitude always corresponds to a higher jump-detecting

Table 5. The average computing time (in seconds) per replication of CLLJP, JDBS-I, and JDBS-II procedures when $f = f_1$, $N = 1000$, and $\varepsilon \sim N(0, \sigma^2)$.

σ	Method	$n = 200$	$n = 300$	$n = 500$
0.1	CLLJP	11.54	44.84	364.44
	JDBS-I	4.77	6.33	9.24
	JDBS-II	1.73	2.04	2.65
0.2	CLLJP	11.49	43.96	360.61
	JDBS-I	4.07	5.69	7.97
	JDBS-II	1.66	1.97	2.59
0.4	CLLJP	11.36	44.12	363.86
	JDBS-I	3.05	5.43	8.57
	JDBS-II	1.68	1.92	2.45

Table 6. Bias, standard deviation, detecting power, and frequency of jump point based on JDBS-I procedure when $f = f_2, N = 1000$, and $\varepsilon \sim N(0, \sigma^2)$.

						Frequency				
σ	n	Jumps	Bias	sd	Power	0	1	2	3	>3
0.1	200	0.3	0.0000	0.0046	0.998	0	154	835	8	3
		0.7	0.0002	0.0047	0.846					
	300	0.3	0.0000	0.0031	1	0	9	943	45	3
		0.7	0.0002	0.0031	0.991					
	500	0.3	0.0000	0.0016	1	0	0	903	89	8
		0.7	0.0000	0.0021	1					
0.2	200	0.3	−0.0002	0.0063	0.936	19	428	490	54	9
		0.7	−0.0002	0.0079	0.554					
	300	0.3	−0.0001	0.0032	0.988	1	204	647	135	13
		0.7	0.0001	0.0048	0.788					
	500	0.3	0.0000	0.0016	0.998	0	41	774	162	23
		0.7	0.0000	0.0021	0.959					
0.4	200	0.3	−0.0016	0.0118	0.623	186	508	253	48	5
		0.7	0.0003	0.0160	0.265					
	300	0.3	−0.0005	0.0087	0.758	114	469	313	95	9
		0.7	−0.0001	0.0125	0.369					
	500	0.3	−0.0001	0.0054	0.908	33	416	409	124	18
		0.7	−0.0005	0.0099	0.510					

power, which means that the jump magnitude has a significant effect on the jump-detecting power. From Table 8 and Figure 3, the fitted effect can also be seen to be good.

In order to further study the effect of the smoothness of regression curve on jump detecting, Example 3 is considered here. The simulation results can be seen from Tables 9 to 11 and Figure 4. Similar conclusion could be obtained as in Example 1 and Example 2.

6. Real data analysis

The stock market as the barometer of national economy is highly valued by government and investors. The stock market is full of uncertainty, abundant opportunities, and risks.

Table 7. Bias, standard deviation, detecting power, and frequency of jump point based on JDBS-II procedure when $f = f_2, N = 1000$, and $\varepsilon \sim N(0, \sigma^2)$.

						Frequency				
σ	n	Jumps	Bias	sd	Power	0	1	2	3	>3
0.1	200	0.3	0.0001	0.0046	0.998	0	19	935	46	0
		0.7	0.0002	0.0050	0.981					
	300	0.3	−0.0001	0.0028	1	0	19	973	8	0
		0.7	0.0001	0.0031	0.978					
	500	0.3	0.0000	0.0017	1	0	0	990	10	0
		0.7	0.0000	0.0019	1					
0.2	200	0.3	0.0002	0.0057	0.981	1	84	666	227	22
		0.7	−0.0005	0.0092	0.863					
	300	0.3	0.0000	0.0032	0.996	0	85	771	137	7
		0.7	0.0000	0.0059	0.888					
	500	0.3	0.0000	0.0018	1	0	36	876	82	6
		0.7	0.0001	0.0029	0.953					
0.4	200	0.3	−0.0005	0.0120	0.856	2	142	442	308	106
		0.7	0.0007	0.0172	0.558					
	300	0.3	0.0001	0.0090	0.923	1	65	391	398	145
		0.7	0.0007	0.0155	0.709					
	500	0.3	0.0001	0.0053	0.976	0	63	557	310	70
		0.7	−0.0001	0.0104	0.806					

Table 8. MISE and LMISE when $f = f_2$, $N = 1000$, and $\varepsilon \sim N(0, \sigma^2)$.

		$n = 200$		$n = 300$		$n = 500$	
σ	Method	MISE	LMISE	MISE	LMISE	MISE	LMISE
0.1	CB	0.0092	0.0082	0.0065	0.0058	0.0057	0.0051
	CLLJP	0.0015	0.0006	0.0012	0.0005	0.0007	0.0003
	JDBS-I	0.0012	0.0007	0.0006	0.0002	0.0003	0.0001
	JDBS-II	0.0009	0.0004	0.0006	0.0002	0.0003	0.0001
0.2	CB	0.0136	0.0105	0.0117	0.0094	0.0092	0.0077
	CLLJP	0.0047	0.0019	0.0035	0.0014	0.0019	0.0009
	JDBS-I	0.0059	0.0035	0.0033	0.0017	0.0015	0.0006
	JDBS-II	0.0051	0.0025	0.0030	0.0015	0.0015	0.0006
0.4	CB	0.0243	0.0145	0.0195	0.0130	0.0161	0.0109
	CLLJP	0.0216	0.0090	0.0155	0.0073	0.0127	0.0074
	JDBS-I	0.0219	0.0110	0.0161	0.0087	0.0099	0.0056
	JDBS-II	0.0219	0.0094	0.0159	0.0074	0.0094	0.0047

Therefore, if we could discover some useful information, it would help people better seize the opportunities and avoid risks. The stock price index is an effective tool to measure the market information which reflects the activity level of the market. Hence, it is important to analyze the stock price index statistically to seize more information.

As we know from statistical view, the stock price index can be seen as a function of time which is often discontinuous at some time points. As an example, we have collected a set of data of closing price about daily Shanghai securities composite index from January 2, 2014, to

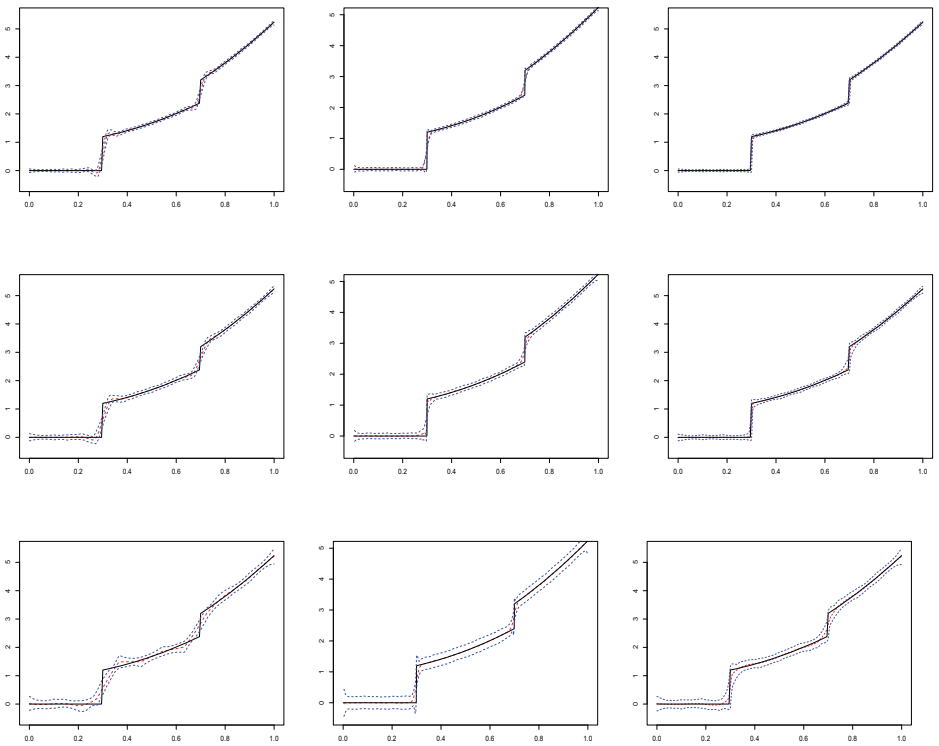

Figure 3. Plot of curve estimate when $f = f_2$, $n = 300$, and $N = 1000$. The three rows of panels, respectively, show the fitted curves when $\sigma = 0.1$, $\sigma = 0.2$, and $\sigma = 0.4$. The three columns of panels, respectively, show fitted curves based on CB, CLLJP, and JDBS-I methods. The solid curve represents the true regression model. The fitted curves and the 95% pointwise confidence intervals are denoted by dotted curves.

Table 9. Bias, standard deviation, detecting power, and frequency of jump point based on JDBS-I procedure when $f = f_3, N = 1000$, and $\varepsilon \sim N(0, \sigma^2)$.

						Frequency				
σ	n	Jumps	Bias	sd	Power	0	1	2	3	>3
0.1	200	0.5	0.0000	0.0082	0.793	213	780	7	0	0
	300	0.5	0.0000	0.0054	0.994	8	990	2	0	0
	500	0.5	0.0001	0.0045	1	0	982	17	0	1
0.2	200	0.5	0.0001	0.0093	0.762	242	740	16	2	0
	300	0.5	0.0000	0.0059	0.951	53	917	25	4	1
	500	0.5	0.0000	0.0044	1	5	929	45	17	4
0.4	200	0.5	0.0005	0.0091	0.460	483	405	80	28	4
	300	0.5	0.0001	0.0073	0.714	260	560	130	44	6
	500	0.5	0.0000	0.0039	0.955	45	732	163	40	20

Table 10. Bias, standard deviation, detecting power, and frequency of jump point based on JDBS-II procedure when $f = f_3, N = 1000$, and $\varepsilon \sim N(0, \sigma^2)$.

						Frequency				
σ	n	Jumps	Bias	sd	Power	0	1	2	3	>3
0.1	200	0.5	0.0001	0.0081	0.986	14	978	8	0	0
	300	0.5	−0.0011	0.0051	0.996	4	956	37	3	0
	500	0.5	0.0000	0.0030	1	0	984	16	0	0
0.2	200	0.5	−0.0006	0.0085	0.957	26	780	176	18	0
	300	0.5	−0.0003	0.0057	0.987	8	754	221	17	0
	500	0.5	0.0001	0.0034	0.997	1	830	155	14	0
0.4	200	0.5	0.0000	0.0097	0.869	19	318	401	197	65
	300	0.5	0.0001	0.0065	0.949	7	281	403	248	61
	500	0.5	−0.0001	0.0043	0.995	0	240	430	245	85

December 30, 2016 (see website http://q.stock.sohu.com). The real data are shown in Figure 5 (the black solid line). During this 3-year period, the stock market experienced several crises, which is known as Chinese stock market turbulence. The daily Shanghai securities composite index was unstable in this period. The turbulence began on June 15, 2015, and ended in early February 2016. Three steep falls can be seen in June 2015, August 2015, and January 2016. However, due to the impact of noise, the position and magnitude of jump points are unknown to us. Therefore, detecting jumps and recovering closing price curve are needed to

Table 11. MISE and LMISE when $f = f_3, N = 1000$, and $\varepsilon \sim N(0, \sigma^2)$.

		$n = 200$		$n = 300$		$n = 500$	
σ	Method	MISE	LMISE	MISE	LMISE	MISE	LMISE
0.1	CB	0.0129	0.0105	0.0121	0.0101	0.0110	0.0093
	CLLJP	0.0081	0.0064	0.0073	0.0060	0.0045	0.0031
	JDBS-I	0.0076	0.0054	0.0057	0.0039	0.0046	0.0032
	JDBS-II	0.0068	0.0045	0.0060	0.0042	0.0047	0.0033
0.2	CB	0.0196	0.0137	0.0173	0.0128	0.0142	0.0109
	CLLJP	0.0131	0.0087	0.0117	0.0070	0.0059	0.0032
	JDBS-I	0.0130	0.0073	0.0096	0.0052	0.0069	0.0038
	JDBS-II	0.0112	0.0051	0.0088	0.0043	0.0066	0.0034
0.4	CB	0.0380	0.0206	0.0322	0.0195	0.0258	0.0172
	CLLJP	0.0298	0.0103	0.0201	0.0070	0.0123	0.0037
	JDBS-I	0.0293	0.0114	0.0213	0.0079	0.0137	0.0049
	JDBS-II	0.0274	0.0055	0.0199	0.0044	0.0138	0.0035

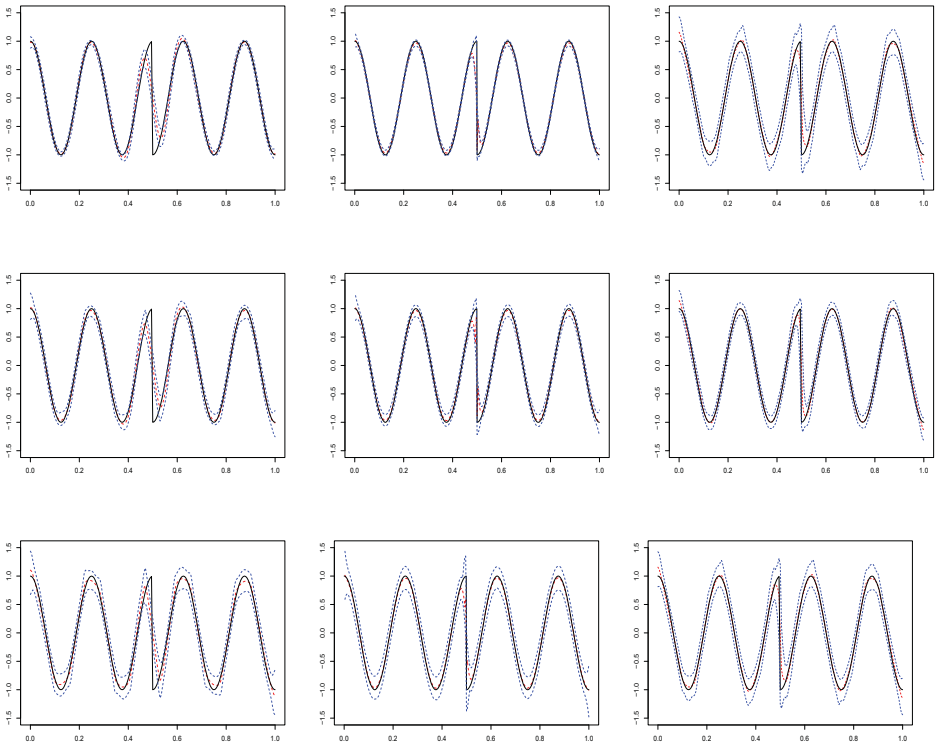

Figure 4. Plot of curve estimate when $f = f_3$, $n = 300$, and $N = 1000$. The three rows of panels, respectively, show the fitted curve when $\sigma = 0.1, \sigma = 0.2$, and $\sigma = 0.4$. The three columns of panels, respectively, show fitted curves based on CB, CLLJP, and JDBS-I methods. The solid curve represents the true regression model. The fitted curve and the 95% pointwise confidence interval are denoted by dotted curves.

concern more. In the following, we deal with this example by JDBS-I and CLLJP procedures, respectively.

First, we normalize the design points such that the normalized design interval is [0, 1]. Then, we choose procedure parameter by the BIC. Following the jump-detection procedure

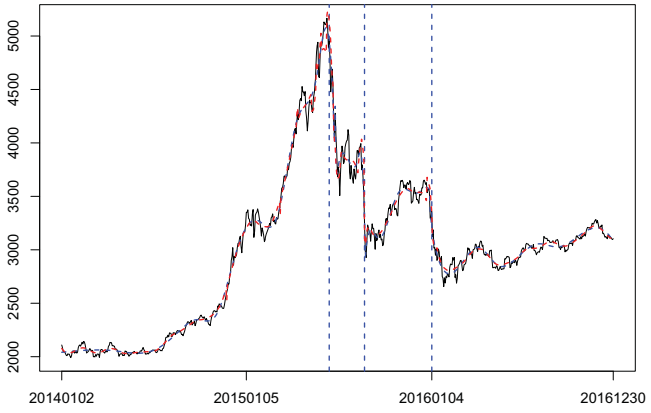

Figure 5. Plot of the daily Shanghai securities composite index from January 2, 2014, to December 30, 2016, with the sample size $n = 733$. The black solid line shows the real data. The red dotted line shows the fitted curve based on CLLJP method. The blue dashed line shows the fitted curve based on JDBS-I method. The blue vertical dashed lines show the detected jump points based on JDBS-I method.

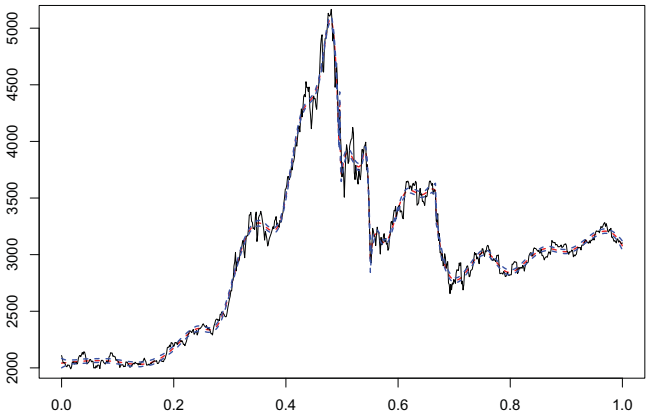

Figure 6. The fitted curve based on JDBS-I procedure and the 95% pointwise confidence interval.

in Section 3, three jump points at 0.484, 0.551, and 0.667 which could be seen from Figure 5 (the blue vertical lines, corresponding to the data June 16, 2015, August 25, 2015, and January 4, 2016), are identified. The performance of JDBS-I and CLLJP procedures is evaluated by their residual sums of square. The results show that JDBS-I procedure has smaller residual sum of square than that of CLLJP procedure (3092386 vs 4177232). From Figure 5, it can also be seen that the fitted curve of JDBS-I procedure is closer to the real data than that of the CLLJP procedure whether at the jump points or in the continuous regions. In addition, the elapsed time of our procedure is much less than CLLJP (12.03 vs. 948.32 seconds). From the above analysis and related results, our procedure works a bit better than the CLLJP procedure.

During the past few years, China is experiencing economic transformation. In 2015, many individual investors inflated the stock market with borrowed money through mass amounts of investments in stocks, which exceeds the rate of economic growth and profits of the companies. The stock market bubble burst after June 12, 2015, which corresponds to the first jump. Some commentators called August 24 and 25, 2015 (which corresponds to the second jump) Black Monday and Tuesday, because Shanghai main share index lost 15% of its value in the 2 days. At the beginning of January 2016 (which corresponds to the third jump), the Chinese stock market experienced a sharp sell-off of about 7% that quickly sent stocks tumbling globally.

Since the crash that started on June 15, 2015, the Chinese government took many unusual steps to stop it. For instance, on July 8, 2015, the central bank pledged to help maintain market stability, and Chinese stock regulators banned company officers and major shareholders from selling shares in listed companies. Therefore, if we could find new method to detect jumps and recover curve in the future stock market, our government would take some actions in advance, which could avoid some unnecessary risks and losses.

7. Conclusion and discussion

In this article, we have presented the jump-detection and curve estimation procedures based on B-splines. Simulations and real data analysis show that they work quite well. Our procedures have the advantage of fast computation, especially compared with the CLLJP procedure. Furthermore, compared with the CLLJP procedure, our procedures have smaller MISE and LMISE, which implies that our procedures perform better on jump preserving and smooth preserving.

However, our procedure still needs further study. First, the procedure presented in our article works well when only some significant jumps exist in regression models. However, many data (especially financial data) often contain abundant of jumps, and our procedure will not suitable any more; for this reason, alternative methods based on pure jump processes seem to be more appropriate to describe them in the real data analysis. The pure jump process has been studied extensively in the literature, see Jing, Kong, and Liu (2012) and Kong, Liu, and Jing (2015) and references therein. Second, our procedure requires that there is no more than one jump within a small range. Therefore, if there are more than one jump within a small range, our procedure will be invalid. For such case, the multipower variation method of diffusion process can be considered (see Woerner 2006; Liu, Wei, and Zhang 2013, and the others). Finally, the local curve and surface approximations have been used to preserve edges in two-dimensional (2D) and 3D image denoizing, see Qiu (2007) and Mukherjee and Qiu (2011) and so on. But in many literature, the computation complexity is rather high; therefore, our method may be extended to 2D or 3D cases to reduce the burden of calculation.

Funding

This research is supported by the National Natural Science Foundation of China (No. 11471252; No. 11571073), and Postdoctoral Program Foundation of Jiangsu Province of China (No. 1501021C).

ORCID

Xiu-Li Du http://orcid.org/0000-0002-6798-0504

References

Barrow, D. L., and P. W. Smith. 1978. Asymptotic properties of best $L_2[0, 1]$ approximation by splines with variable knots. *Quarterly of Applied Mathematics* 36 (3):293–304.

De Boor, C. 1978. *A practical guide to splines*. New York: Springer-Verlag.

Gijbels, I., A. Lambert, and P. H. Qiu. 2007. Jump-preserving regression and smoothing using local linear fitting: a compromise. *Annals of the Institute of Statistical Mathematics* 59 (2):235–72.

Hall, P., and D. M. Titterington. 1992. Edge-preserving and peak-preserving smoothing. *Technometrics* 34 (4):429–40.

Jing, B. Y., X. B. Kong, and Z. Liu. 2012. Modeling high-frequency financial data by pure jump processes. *Annals of Statistics* 40 (2):759–84.

Joo, J. H., and P. H. Qiu. 2009. Jump detection in a regression curve and its derivative. *Technometrics* 51 (3):289–305.

Jupp, D. L. B. 1978. Approximation to data by splines with free knots. *SIAM Journal on Numerical Analysis* 15 (2):328–43.

Kong, X. B., Z. Liu, and B. Y. Jing. 2015. Testing for pure-jump processes for high-frequency data. *Annals of Statistics* 43 (2):847–77.

Koo, J. Y. 1997. Spline estimation of discontinuous regression functions. *Journal of Computational and Graphical Statistics* 6 (3):266–84.

Liu, G. Y., Z. Y. Wei, and X. S. Zhang. 2013. Asymptotic properties for multipower variation of semimartingales and Gaussian integral processes with jumps. *Journal of Statistical Planning and Inference* 143 (8):1307–19.

McDonald, J. A., and A. B. Owen. 1986. Smoothing with split linear fits. *Technometrics* 28 (3):195–208.

Miyata, S., and X. T. Shen. 2003. Adaptive free-knot splines. *Journal of Computational and Graphical Statistics* 12 (1):197–213.

Mukherjee, P. S., and P. H. Qiu. 2011. 3-D image denoising by local smoothing and nonparametric regression. *Technometrics* 53 (2):196–208.

Qiu, P. H. 1994. Estimation of the number of jumps of the jump regression functions. *Communications in Statistics - Theory and Methods* 23 (8):2141–55.
Qiu, P. H. 2003. A jump-preserving curve fitting procedure based on local piecewise-linear kernel estimation. *Journal of Nonparametric Statistics* 15 (4–5):437–53.
Qiu, P. H. 2007. Jump surface estimation, edge detection, and image restoration. *Journals - American Statistical Association* 102 (478):745–56.
Qiu, P. H. 2009. Jump-preserving surface reconstruction from noisy data. *Annals of the Institute of Statistical Mathematics* 61 (3):15–751.
Qiu, P. H., and B. Yandell. 1998. A local polynomial jump detection algorithm in nonparametric regression. *Technometrics* 40 (2):141–52.
Rice, J. 1984. Bandwidth choice for nonparametric regression. *Annals of Statistics* 12 (4):1215–31.
Schumaker, L. 2007. *Spline functions: Basic theory*. New York: Cambridge University Press.
Wahba, G. 1986. Partial spline modelling of the tropopause and other discontinuities. In *Function estimates, volume 59 of contemporary mathematics*, ed. J. S. Marron, 125–35. Providence, RI: AMS.
Wei, B. C. 2006. *A course in parametric statistics*. Beijing: Higher Education Press.
Woerner, J. H. C. 2006. Power and multipower variation: inference for high frequency data. In *Stochastic finance*, eds. A. N. Shiryaev, M. R. Grossinho, P. E. Oliveira, M. L. Esquvel, 343–64. Berlin: Springer.
Xia, Z. M., and P. H. Qiu. 2015. Jump information criterion for statistical inference in estimating discontinuous curves. *Biometrika* 102 (2):397–408.
Yang, Y. J., and Q. X. Song. 2014. Jump detection in time series nonparametric regression models: a polynomial spline approach. *Annals of the Institute of Statistical Mathematics* 66 (2):325–44.
Zhao, Y. Y., J. G. Lin, X. F. Huang, and H. X. Wang. 2016. Adaptive jump-preserving estimates in varying-coefficient models. *Journal of Multivariate Analysis* 149:65–80.
Zhou, S. G., and X. T. Shen. 2001. Spatially adaptive regression splines and accurate knot selection schemes. *Journal of the American Statistical Association* 96 (453):247–59.

Appendix

Proof of Theorem 2.

(i) We first consider the special case $\varepsilon \sim N(0, \sigma^2)$, i.e., $y_i \sim N(f(x_i), \sigma^2)$, $i = 1, 2, \ldots, n$; therefore, we have $\mathbf{Y} \sim N(f(\mathbf{X}), \sigma^2\mathbf{I}_n)$.
Let $x \in (u_i, u_{i+1})$. Note that before inserting x into U^* with multiplicity $p+1$, the number of B-spline basis functions $\{B_{i,p}(x), i = 0, 1, \ldots, k+p\}$ for B-spline estimate of unknown function $f(\cdot)$ is $p+k+1$; therefore, the rank of matrix $\mathbf{M} = (\mathbf{B}(x_1), \ldots, \mathbf{B}(x_n))^T$ is also $p+k+1$; however after inserting x into U^* with multiplicity $p+1$, the number of B-spline basis functions $\{\tilde{B}_{i,p}(x), i = 0, 1, \ldots, 2p+k+1\}$ becomes $2p+k+2$. However, in the process, only $p+1$ B-spline basis functions $B_{i,p}(x), \ldots, B_{i+p,p}(x)$ in $\{B_{i,p}(x), i = 0, 1, \ldots, k+p\}$ change into $2p+2$ new B-spline basis functions $\tilde{B}_{i,p}(x), \ldots, \tilde{B}_{i+2p+1,p}(x)$ in $\{\tilde{B}_{i,p}(x), i = 0, 1, \ldots, 2p+k+1\}$, the rest does not change. It could be proved by the recursive formula of B-spline basis that the $p+1$ old B-spline basis functions can be linearly expressed by the new $2p+2$ B-spline basis functions. Therefore, there exists a matrix $\mathbf{T}$ such that $\mathbf{M} = \tilde{\mathbf{M}}\mathbf{T}$. Note that

$$\begin{aligned} P_{\tilde{\mathbf{M}}}P_{\mathbf{M}} &= \tilde{\mathbf{M}}(\tilde{\mathbf{M}}^T\tilde{\mathbf{M}})^{-1}\tilde{\mathbf{M}}^T\mathbf{M}(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T \\ &= \tilde{\mathbf{M}}(\tilde{\mathbf{M}}^T\tilde{\mathbf{M}})^{-1}\tilde{\mathbf{M}}^T\tilde{\mathbf{M}}\mathbf{T}(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T \\ &= \tilde{\mathbf{M}}\mathbf{T}(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T = \mathbf{M}(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T = P_{\mathbf{M}}, \end{aligned}$$

similarly, $P_{\mathbf{M}}P_{\tilde{\mathbf{M}}} = P_{\mathbf{M}}$. Hence, we have $(P_{\tilde{\mathbf{M}}} - P_{\mathbf{M}})(\mathbf{I} - P_{\tilde{\mathbf{M}}}) = \mathbf{0}$, which means $DRSS(x) = \sum_{i=1}^{n}(\hat{f}(x_i; U^*) - \hat{f}(x_i; \tilde{U}^*))^2$ and $(P_{\tilde{\mathbf{M}}} - P_{\mathbf{M}})^2 = P_{\tilde{\mathbf{M}}} - P_{\mathbf{M}}$, i.e., $P_{\tilde{\mathbf{M}}} - P_{\mathbf{M}}$

is a symmetric idempotent matrix, hence the rank of $P_{\tilde{\mathbf{M}}} - P_{\mathbf{M}}$ is

$$rank(P_{\tilde{\mathbf{M}}} - P_{\mathbf{M}}) = tr(P_{\tilde{\mathbf{M}}} - P_{\mathbf{M}}) = tr(P_{\tilde{\mathbf{M}}}) - tr(P_{\mathbf{M}}) = rank(P_{\tilde{\mathbf{M}}}) - rank(P_{\mathbf{M}}) = p + 1.$$

By p. 27 of Wei (2006), we have $\mathbf{Y}^T(P_{\tilde{\mathbf{M}}} - P_{\mathbf{M}})\mathbf{Y}/\sigma^2 \sim \chi^2_{p+1}(\lambda_x)$.
If ε_i satisfies unknown distribution with mean 0 and variance σ^2. We can check the Lindeberg–Feller condition and prove that $\mathbf{Y}^T P_{\mathbf{M}}\mathbf{Y}/\sigma^2 \sim \chi^2_{p+k+1}$ and $\mathbf{Y}^T P_{\tilde{\mathbf{M}}}\mathbf{Y}/\sigma^2 \sim \chi^2_{2p+k+2}$. Since $rank(P_{\tilde{\mathbf{M}}} - P_{\mathbf{M}}) = p + 1$, by Cochran's theorem, we have $\mathbf{Y}^T(P_{\tilde{\mathbf{M}}} - P_{\mathbf{M}})\mathbf{Y}/\sigma^2 \sim \chi^2_{p+1}(\lambda_x)$.

(ii) Similar to the above proof, we can obtain the result of (ii). □