A Local Polynomial Jump-Detection Algorithm in Nonparametric Regression

Peihua Qiu; Brian Yandell

Technometrics, Vol. 40, No. 2. (May, 1998), pp. 141-152.

Stable URL:
http://links.jstor.org/sici?sici=0040-1706%28199805%2940%3A2%3C141%3AALPJAI%3E2.0.CO%3B2-J

Technometrics is currently published by American Statistical Association.

Your use of the JSTOR archive indicates your acceptance of JSTOR's Terms and Conditions of Use, available at http://www.jstor.org/about/terms.html. JSTOR's Terms and Conditions of Use provides, in part, that unless you have obtained prior permission, you may not download an entire issue of a journal or multiple copies of articles, and you may use content in the JSTOR archive only for your personal, non-commercial use.

Please contact the publisher regarding any further use of this work. Publisher contact information may be obtained at http://www.jstor.org/journals/astata.html.

Each copy of any part of a JSTOR transmission must contain the same copyright notice that appears on the screen or printed page of such transmission.

JSTOR is an independent not-for-profit organization dedicated to creating and preserving a digital archive of scholarly journals. For more information regarding JSTOR, please contact support@jstor.org.

http://www.jstor.org/
Tue Sep 26 11:58:59 2006

A Local Polynomial Jump-Detection Algorithm in Nonparametric Regression

Peihua QIU

Biostatistics Program
The Ohio State University
Columbus, OH 43210-1240
(qiu@biostat3.med.ohio-state.edu)

Brian YANDELL

Department of Statistics
University of Wisconsin
Madison, WI 53706
(yandell@stat.wisc.edu)

We suggest a one-dimensional jump-detection algorithm based on local polynomial fitting for jumps in regression functions (zero-order jumps) or jumps in derivatives (first-order or higher-order jumps). If jumps exist in the mth-order derivative of the underlying regression function, then an $(m+1)$-order polynomial is fitted in a neighborhood of each design point. We then characterize the jump information in the coefficients of the highest-order terms of the fitted polynomials and suggest an algorithm for jump detection. This method is introduced briefly for the general setup and then presented in detail for zero-order and first-order jumps. Several simulation examples are discussed. We apply this method to the Bombay (India) sea-level pressure data.

KEY WORDS: Edge detection; Image processing; Jump-detection algorithm; Least squares line; Modification procedure; Nonparametric jump regression model; Threshold value.

1. INTRODUCTION

Stock-market prices often jump up or down under the influence of some important random events. Physiological responses to stimuli can likewise jump after physical or chemical shocks. Regression functions with jumps may be more appropriate than continuous regression models for such data. The one-dimensional (1-D) nonparametric jump regression model (NJRM) with jumps in the mth derivative can be expressed as

$$Y_i = f(t_i) + \varepsilon_i, \quad i = 1, 2, \ldots, n, \tag{1.1}$$

and

$$f^{(m)}(t) = g(t) + \sum_{i=1}^{p} d_i I_{[s_i, s_{i+1})}(t), \tag{1.2}$$

with design points $0 \leq t_1 < t_2 < \cdots < t_n \leq 1$ and iid errors $\{\varepsilon_i\}$ having mean 0 and unknown variance σ^2. The mth-order derivative $f^{(m)}(t)$ of the regression function $f(t)$ consists of a continuous part $g(t)$ and p jumps at positions $\{s_i, i = 1, 2, \ldots, p\}$ with magnitudes $\{d_i - d_{i-1}, i = 1, 2, \ldots, p\}$. For convenience, let $d_0 = 0$ and $s_{p+1} = 1$. Zero-order jumps ($m = 0$) in the regression function itself correspond to the step edge in image processing. First-order jumps ($m = 1$) may exist in the first derivative of $f(t)$, related to the roof edge in image processing. For $m > 1$, the jumps in (1.2) are called higher-order. The objective of this article is to develop an algorithm to detect the jumps of 1-D NJRM (1.1)–(1.2) from the noisy observations $\{Y_i, i = 1, 2, \ldots, n\}$.

Example 1.1: Bombay Sea-Level Pressure Data. Figure 1 shows sea-level pressure data that were provided by Dr. Wilbur Spangler at the National Center for Atmospheric Research, Boulder, Colorado. The small dots represent the December sea-level pressures during 1921–1992 in Bombay, India. Shea, Worley, Stern, and Hoar (1994) pointed out that "a discontinuity is clearly evident around 1960. . . . Some procedure should be used to adjust for the continuity" (p. 14). By using the procedure introduced in this article, a jump is detected. The fitted model with this detected jump accommodated is shown in the plot as a solid curve. For comparison, we plot the fitted model with the usual kernel-smoothing method as a dotted curve. More explanation about this example is given in Section 4.3.

McDonald and Owen (1986) proposed an algorithm based on three smoothed estimates of the regression function, corresponding to the observations on the right, left, and both sides of a point in question, respectively. They then constructed a "split linear smoother" as a weighted average of these three estimates, with weights determined by the goodness-of-fit values of the estimates. If there is a jump near the given point, then only some of these three estimates are likely to provide good fits, accommodating the discontinuities of the regression functions. Hall and Titterington (1992) suggested an alternative method by establishing some relations among three local linear smoothers and using them to detect the jumps. This latter method is easier to implement.

Related research on this topic includes the kernel-type methods for jump detection of Müller (1992), Qiu (1991, 1994), Qiu, Asano, and Li (1991), Wu and Chu (1993), Yin (1988), and others. These methods were based on the difference between two one-sided kernel smoothers. Wahba (1986), Shiau (1987), and several others regarded the NJRM's as partial linear regression models and fitted them with partial splines. More recently, Eubank and Speckman (1994) and Speckman (1993) treated the NJRM (1.1)–(1.2) as a semiparametric regression model and proposed estimates of the jump locations and magnitudes. Loader (1996)

© 1998 American Statistical Association
and the American Society for Quality

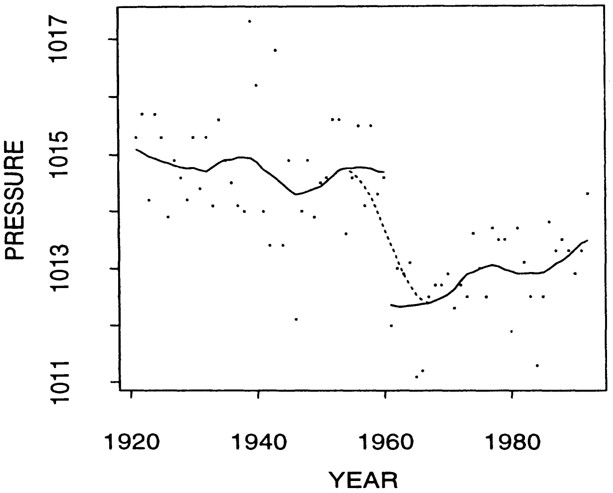


Figure 1. The December Sea-level Pressures During 1921–1992 in Bombay, India. The solid curves represent the fit from our jump-preserving algorithm. The dotted curve is the usual kernel-smoothing fit without considering the jump structure.

suggested a jump detector based on local polynomial kernel estimators.

In this article, we suggest an alternative method. At a given design point t_i, we consider its neighborhood $N(t_i) := \{t_{i-l}, t_{i+1-l}, \ldots, t_i, \ldots, t_{i-1+l}, t_{i+l}\}$ with width $k = 2l + 1 \ll n$, an odd positive integer, for $l + 1 \leq i \leq n - l$. We then fit a local polynomial function of order $(m + 1)$ by the least squares (LS) method in that neighborhood, which can be expressed as

$$\hat{Y}^{(i)}(t) = \hat{\beta}_0^{(i)} + \hat{\beta}_1^{(i)} t + \cdots + \hat{\beta}_{m+1}^{(i)} t^{m+1},$$
$$t \in N(t_i), i = l + 1, \ldots, n - l. \quad (1.3)$$

Intuitively, if $f^{(m)}(t)$ is smooth at t_i, then $\hat{\beta}_{m+1}^{(i)}$ is close to $f^{(m+1)}(t_i)$ for large enough n. If $f^{(m)}(t)$ has a jump at t_i, however, $\{\hat{\beta}_{m+1}^{(j)}\}_{j=l+1}^{n-l}$ has an abrupt change at $\hat{\beta}_{m+1}^{(i)}$. Hence, these coefficients carry information about both the continuous and the jump components of $f^{(m)}(t)$ [given in (1.2)].

A jump-detection criterion can be formed that excludes the information about the continuous part but preserves the jump information. An important requirement of such an algorithm is that it be easy to implement. From the preceding brief description, this algorithm is based on estimated LS coefficients that are available from common statistical software. Its computational complexity is $O(n)$. Another feature of this method is that it does not require the number of jumps to be known beforehand, as do most other existing methods. Jumps are automatically accommodated with our jump-preserving curve-fitting procedure.

We should point out that in our algorithm the window width k must be specified. How should one choose a proper window width in practice? This is a common problem in local smoothing methods. In some applications, a visual intuitive method has been used to adjust the window width. Hastie and Tibshirani (1987) suggested using 10%–50% observations for each running-lines smoother in their local scoring algorithm. Stone (1977) suggested the cross-validation method to choose the window width. For more discussions on the selection of the window width, please refer to Härdle (1991, chap. 6).

In most applications, we are interested in checking for jumps in the regression function itself or in its first-order derivative. Hence, in the following sections, we concentrate on these two special cases. In Section 2, a jump-detection criterion for the case of $m = 0$ is derived, with a corresponding algorithm. Jump detection in slope is discussed in Section 3. In Section 4, some simulation results are presented. We return to the Bombay (India) sea-level pressure data in Section 4.3. Our method is compared with some kernel-type methods in Section 5. We conclude the article with some remarks in Section 6. Some supporting materials are given in the Appendix.

We should point out that the 2-D version of the current problem (namely, jump detection in surfaces) is closely related to edge detection in image processing. We refer interested readers to Besag, Green, Higdon, and Mengerson (1995), Gonzalez and Woods (1992), Qiu and Bhandarkar (1996), and Qiu and Yandell (1997) and the references cited there.

2. JUMP DETECTION IN THE REGRESSION FUNCTIONS

In this section we discuss the jump detection in the regression function itself. This corresponds to $m = 0$ in Model (1.1)–(1.2). For simplicity of presentation, we assume that the design points are equally spaced over [0, 1]. Most of the derivation of the jump-detection criterion presented here is intuitive, although mathematically rigorous arguments can be constructed. Based on the derived criterion, we suggest a jump-detection algorithm.

As described in Section 1, we fit an LS line, $\hat{Y}^{(i)}(t) = \hat{\beta}_0^{(i)} + \hat{\beta}_1^{(i)} t, t \in N(t_i)$, in a neighborhood $N(t_i)$ at each design point t_i, for $l + 1 \leq i \leq n - l$. Throughout this article we make the following assumption (AS1) on the NJRM of Equations (1.1)–(1.2).

AS1. Only one jump is possible in any neighborhood $N(t_i)$. If t_i is a jump point, then no other jumps exist in $N(t_{i-k}) \cup N(t_i) \cup N(t_{i+k})$.

Remark 2.1. AS1 implies that jump locations are not very close to each other. Alternatively, there are enough data (n large, k/n small) to distinguish nearby jumps. This assumption seems to be reasonable in many applications.

In Appendix A we present a theorem (Theorem A.1) that gives some properties of $\hat{\beta}_1^{(i)}$. By that theorem, $\hat{\beta}_1^{(i)} \sim B_1(t_i) := g'(t_i)$ when there is no jump in $N(t_i)$, and $\hat{\beta}_1^{(i)} \sim B_1(t_i) := g'_+(t_i) + h_1(r)C_0 - \gamma(r)C_1$ if a jump exists in $N(t_i)$ and the jump location is at $t_{i-l+r}, 0 \leq r \leq 2l$, where "$\sim$" means that noise in the data and a high-order term are ignored, C_0 and C_1 are the jump magnitudes of $f(t)$ and its first-order derivative at the jump location, $\gamma(r)$ is a positive function taking values between 0 and 1, $h_1(r) := [6nr/k(k+1)](1 - r/(k-1))C_0$, and $g'_+(t_i) = g'(t_i)$ if $r \neq l$.

Remark 2.2. Some quantities, including $k, \hat{\beta}_1^{(i)}$, and $h_1(r)$, depend on n. We did not make this explicit in notations for simplicity. Their meaning should be clear from the context.

Example 2.1. Let $f(t) = 5t^2 + I_{[.5,1]}(t)$. Consider a sample of size 100 and let k be 7. Then $\hat{\beta}_1^{(i)} \sim B_1(t_i) = 10t_i + I_{\{.47\le t_i \le .53\}}[6(.53-t_i)/(.07)\cdot(.08)](1-(.53-t_i)/.06), 4 \le i \le 97$. $\{B_1(t_i), 4 \le i \le 97\}$ is shown in Figure 2(a).

From Figure 2(a) and Theorem A.1 in Appendix A, we can see that $\{\hat{\beta}_1^{(i)}\}$ carries useful information about the jumps. This information is mainly in the "jump factor" $h_1(r)C_0$ of $B_1(t_i)$. $h_1(r)$ has a maximum value at $r = l$ of $1.5n(k-1)/k(k+1) = O(n/k)$, which tends to infinity with increasing n. If we have prior information about the bound of the "continuous factor" $g'(t_i)$ (when t_i is not a jump point), then those points could be flagged as jump points if their LS slopes are bigger than the prior bound. A prior bound may not be available in many applications, however. Our strategy to overcome this difficulty is to find an operator that simultaneously removes the continuous factor from $B_1(t_i)$ and preserves the jump factor. Notice that the continuous factors $g'(t_i)$ and $g'(t_j)$ are close to each other when t_i and t_j are close. Therefore, a difference-type operator can remove the continuous factor. When t_i and t_j are far enough apart so that only one of $B_1(t_i)$ and $B_1(t_j)$ can have a jump factor, the difference between $B_1(t_i)$ and $B_1(t_j)$ preserves the jump factor. Many difference-type operators could be used. In this article, we suggest using the following:

$$\Delta_1^{(i)} := \begin{cases} \hat{\beta}_1^{(i)} - \hat{\beta}_1^{(i-l)} \\ \quad \text{if } |\hat{\beta}_1^{(i)} - \hat{\beta}_1^{(i-l)}| \le |\hat{\beta}_1^{(i)} - \hat{\beta}_1^{(i+l)}| \\ \hat{\beta}_1^{(i)} - \hat{\beta}_1^{(i+l)} \\ \quad \text{if } |\hat{\beta}_1^{(i)} - \hat{\beta}_1^{(i-l)}| > |\hat{\beta}_1^{(i)} - \hat{\beta}_1^{(i+l)}| \end{cases} \tag{2.1}$$

for $k \le i \le n-k+1$. That is, we use the difference of smaller magnitude. By the preceding intuitive explanations, we can see that

$$\Delta_1^{(i)} \sim J_1(t_i) := \begin{cases} 0 \\ \quad \text{if there is no jump in } N(t_i) \\ h_2(r)C_0 \\ \quad \text{if there is a jump at } t_{i-l+r} \\ \text{with } 0 \le r \le k-1, \end{cases} \tag{2.2}$$

where $h_2(r)$ is the one of $h_1(r) - h_1(r-l)$ and $h_1(r) - h_1(r+l)$ with smaller magnitude and $h_1(j) = 0$ when $j < 0$ or $j > k-1$.

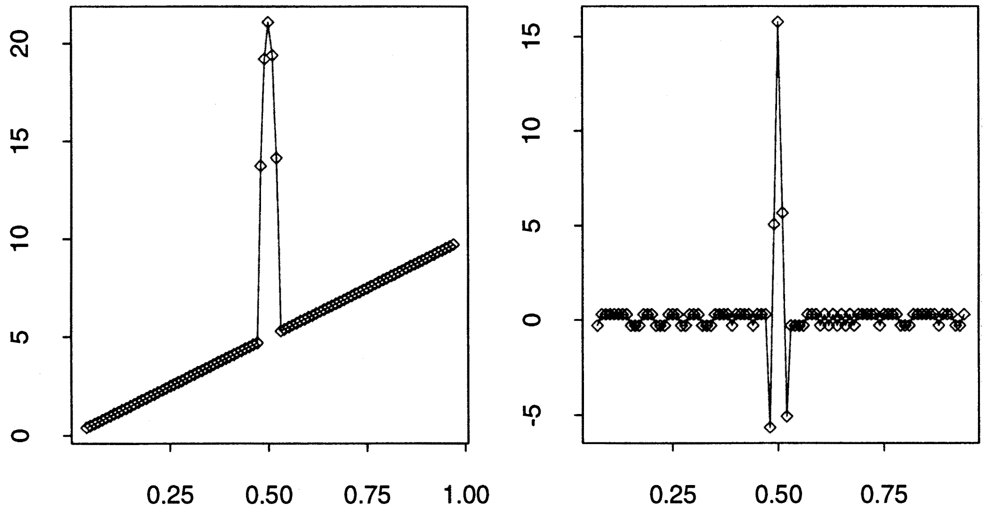
$h_2(r)$ has the same maximum value as $h_1(r)$. In the case of Example (2.1), $\{J_1(t_i), 7 \le i \le 94\}$ is shown in Figure 2(b). From (2.2) and Figure 2(b), we can see that $\{\Delta_1^{(i)}\}$ does retain the jump information and filter the continuous factors at the same time. Hence, it could be used as our jump-detection criterion.

Remark 2.3. A possible alternative operator to (2.1) is as follows. For $k \le i \le n-k+1$, define

$$\Delta_*^{(i)} := \frac{|\hat{\beta}_1^{(i)} - \hat{\beta}_1^{(i+l)}|(\hat{\beta}_1^{(i)} - \hat{\beta}_1^{(i-l)}) + |\hat{\beta}_1^{(i)} - \hat{\beta}_1^{(i-l)}|(\hat{\beta}_1^{(i)} - \hat{\beta}_1^{(i+l)})}{|\hat{\beta}_1^{(i)} - \hat{\beta}_1^{(i-l)}| + |\hat{\beta}_1^{(i)} - \hat{\beta}_1^{(i+l)}|},$$

where $\Delta_*^{(i)}$ is a weighted average of $\hat{\beta}_1^{(i)} - \hat{\beta}_1^{(i-l)}$ and $\hat{\beta}_1^{(i)} - \hat{\beta}_1^{(i+l)}$. From Theorem A.1 and Figure 2(a), we can see that $\Delta_*^{(i)}$ is small when there is no jump in $N(t_i)$. When there is a jump at t_i, $\Delta_*^{(i)} \sim h_1(l)C_0$, where C_0 is the jump magnitude.

Remark 2.4. The difference operator in (2.1) also narrows the regions that contain jump information. A jump



(a) position t

(b) position t

Figure 2. If $f(t) = 5t^2 + I_{[.5,1]}(t)$, $n = 100$, $k = 7$, Then $\hat{\beta}_1^{(i)} \sim B_1(t_i)$. $\{B_1(t_i)\}$ consists of both the continuous and the jump factors. It is shown in plot (a) by the "diamond" points. After using the difference operator defined in (2.1), $\{J_1(t_i)\}$ consists mainly of the jump factor. It is shown in plot (b).

affects $\hat{\beta}_1^{(i)}$ if t_i is within $2k$ of that jump point, but it only affects $\Delta_1^{(i)}$ when t_i is less than k units away. We also pay a price for this increased specificity, however, because the variance of $\Delta_1^{(i)}$ is usually bigger than the variance of $\hat{\beta}_1^{(i)}$. Although the variances of both statistics converge to 0 for increasing neighborhood size, this would be important in finite-sample cases. In applications, we suggest plotting both $\{\hat{\beta}_1^{(i)}\}$ and $\{\Delta_1^{(i)}\}$. In many cases the plot of $\{\hat{\beta}_1^{(i)}\}$ could be very helpful in demonstrating the jumps. This can be seen in the real data example (Sec. 4.3) and in the simulation examples (Secs. 4.1 and 4.2) as well.

Large values of $|\Delta_1^{(i)}|$ indicate possible jumps near t_i. If there is no jump in $N(t_i)$, then $\hat{\beta}_1^{(i)} - \hat{\beta}_1^{(i-l)}$ is approximately normally distributed with mean 0 because it is a linear combination of the observations. From (2.1), $P(|\Delta_1^{(i)}| > u_{1i}) \leq P(|\hat{\beta}_1^{(i)} - \hat{\beta}_1^{(i-l)}| > u_{1i})$ for any $u_{1i} > 0$. Therefore, consider the threshold value $u_{1i} = Z_{\alpha_n/2}\sigma^{(i)}$, with $\sigma^{(i)}$ = SD of $\hat{\beta}_1^{(i)} - \hat{\beta}_1^{(i-l)}$. Clearly, $\sigma^{(i)}$ is independent of i. After some calculations, we have $\sigma^{(i)} = (n/k)\sqrt{6(5k-3)/(k^2-1)}\sigma$. Therefore, a natural choice of the threshold value is

$$u_1 = \hat{\sigma} Z_{\alpha_n/2} \frac{n}{k}\sqrt{\frac{6(5k-3)}{k^2-1}}, \qquad (2.3)$$

where $\hat{\sigma}$ is a consistent estimate of σ.

The design points $\{t_{i_j} : |\Delta_1^{(i_j)}| > u_1, j = 1, 2, \ldots, n_1\}$ can be flagged as candidate jump positions. If t_{i_j} is flagged, then its neighboring design points will be flagged with high probability. Therefore it is useful to eliminate some of the candidates in $\{t_{i_j}, j = 1, 2, \ldots, n_1\}$. We do this by the following modification procedure that was first suggested by Qiu (1994).

Modification Procedure. For a set of candidates $\{t_{i_j}, j = 1, 2, \ldots, n_1\}$, if there are indices $r_1 < r_2$ such that the increments of the sequence $i_{r_1} < \cdots < i_{r_2}$ are all less than the window width k but $i_{r_1} - i_{r_1-1} > k$ and $i_{r_2+1} - i_{r_2} > k$, then we say that $\{t_{i_j}, j = r_1, r_1+1, \ldots, r_2\}$ is a *tie set* in $\{t_{i_j}, j = 1, 2, \ldots, n_1\}$. Select the middle point $(t_{i_{r_1}} + t_{i_{r_2}})/2$ for each tie as a jump-position candidate, replacing the tie set in the candidate set $\{t_{i_j}, j = 1, 2, \ldots, n_1\}$. That is, reduce the candidate set to one representative, the middle point, from each tie set. After the preceding modification procedure, the present candidates include two types of points, those that do not belong to any tie and the middle points of all of the ties.

The jump-detection method is summarized in the following algorithm.

The Zero-Order Jump-Detection Algorithm

1. For any $t_i, l+1 \leq i \leq n-l$, fit an LS line in $N(t_i)$.
2. Use Formula (2.1) to calculate $\Delta_1^{(i)}, k \leq i \leq n-k+1$.
3. Use Formula (2.3) to calculate the threshold value u_1.
4. Flag the design points $\{t_{i_j}, j = 1, 2, \ldots, n_1\}$, where t_{i_j} satisfies $|\Delta_1^{(i_j)}| > u_1$ for $j = 1, 2, \ldots, n_1$.
5. Use the modification procedure to determine the final candidates set $\{b_i, i = 1, 2, \ldots, q_1\}$. Then conclude that jumps exist at $b_1 < b_2 < \cdots < b_{q_1}$.

Remark 2.5. The LS lines fitted in Step 1 of the preceding algorithm can be updated easily from one design point to the next one because only two points change. Thus, the whole algorithm requires $O(n)$ calculations. This remark is also true in the general setup.

Remark 2.6. Our jump-detection algorithm can also be implemented using the centered model

$$\hat{Y}^{(i)}(t) = \hat{\beta}_0^{(i)} + \hat{\beta}_1^{(i)}(t - t_i) + \cdots + \hat{\beta}_{m+1}^{(i)}(t - t_i)^{m+1},$$
$$t \in N(t_i), i = l+1, \ldots, n-l.$$

In many situations, it is more convenient to use this form.

Theorem 2.1. Besides the conditions stated in Theorem A.1 in Appendix A, if the confidence level α_n in (2.3) satisfies the conditions that (a) $\lim_{n\to\infty}\alpha_n = 0$, (b) $\lim_{n\to\infty} Z_{\alpha_n/2}/\sqrt{\log\log k} = \infty$, and (c) $\lim_{n\to\infty} Z_{\alpha_n/2}/\sqrt{k}$

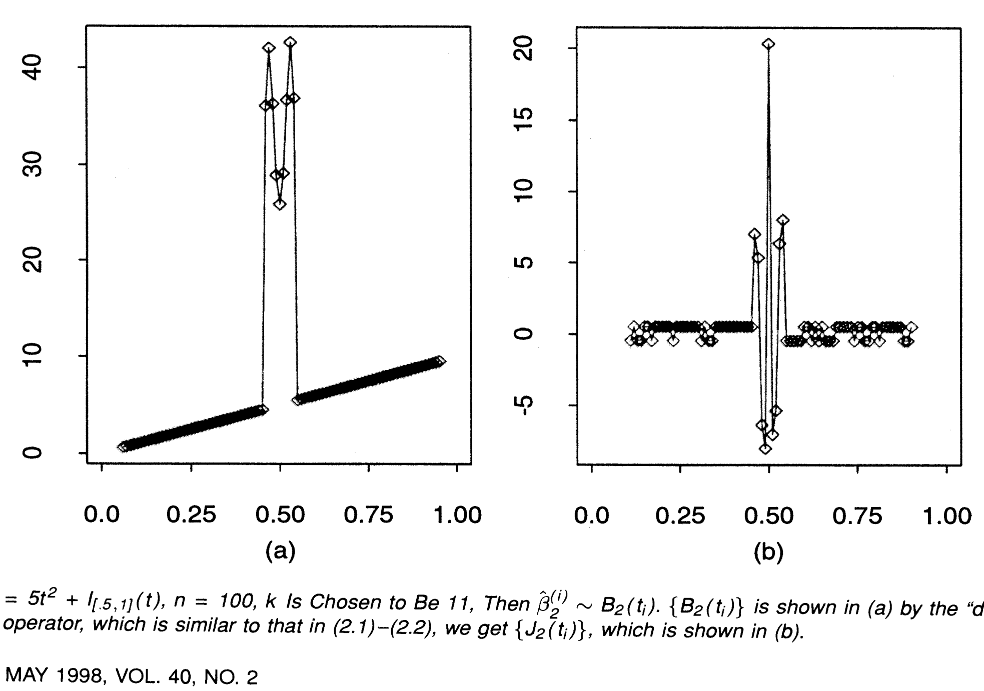


Figure 3. If $f'(t) = 5t^2 + I_{[.5,1]}(t)$, n = 100, k Is Chosen to Be 11, Then $\hat{\beta}_2^{(i)} \sim B_2(t_i)$. $\{B_2(t_i)\}$ is shown in (a) by the "diamond" points. After using the difference operator, which is similar to that in (2.1)–(2.2), we get $\{J_2(t_i)\}$, which is shown in (b).

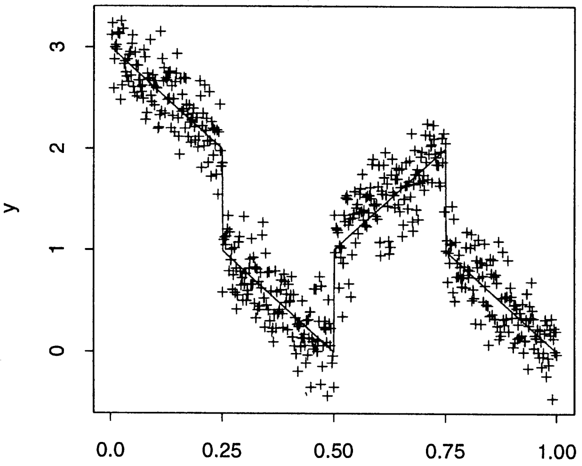


Figure 4. Hall and Titterington Function f (solid lines) and Observations (+).

$= 0$, then (1) $\lim_{n\to\infty} q_1 = p$, a.s. and (2) $\lim_{n\to\infty} b_i = s_i$, a.s., $i = 1, 2, \ldots, p$. The rate of these convergences is $o(n^{-1}\log(n))$. (Proof is given in Appendix B.)

After we detect the possible jump locations $b_1 < b_2 < \cdots < b_{q_1}$, the regression function $f(t)$ could be fitted separately in intervals $\{(b_{i-1}, b_i), i = 1, 2, \ldots, q_1 + 1\}$, where $b_0 = 0$ and $b_{q_1} + 1 = 1$.

To fit $f(t)$ in each interval (b_{i-1}, b_i), we can use either a global smoothing method [e.g., the smoothing spline method (Wahba 1991)] or a local smoothing method [e.g., the kernel-smoothing method (Härdle 1991); the local polynomial kernel method (Wand and Jones 1995)]. By using the kernel-smoothing method, "boundary kernels" are necessary in the border regions of the intervals (e.g., see Stone 1977). When $t \in (b_{i-1}, b_i)$, $\hat{f}(t)$ can be defined as follows:

$$\hat{f}(t) = \frac{\sum_{j=1}^{n} K_{ni}(t_j - t) y_j}{\sum_{j=1}^{n} K_{ni}(t_j - t)} \qquad (2.4)$$

with $K_{ni}(x) = K(x/h_n) I_{\{t+x \in (b_{i-1}, b_i)\}}$, where $K(x)$ is a kernel function with $K(x) = 0$ when $x \notin [-1, 1]$ and h_n is a parameter related to the window size k by $h_n = k/2n$.

3. JUMP DETECTION IN DERIVATIVES

A method for detecting jumps in derivatives can be developed in an analogous manner. We consider only jump detection in the first-order derivative for convenience. As described in Section 1, we fit the following quadratic functions by the LS method for jump detection:

$$\hat{Y}^{(i)}(t) = \hat{\beta}_0^{(i)} + \hat{\beta}_1^{(i)} t + \hat{\beta}_2^{(i)} t^2,$$

$$t \in N(t_i), i = l + 1, \ldots, n - l.$$

Theorem A.2 in Appendix A gives some properties of $\hat{\beta}_2^{(i)}$. It says that, under some regularity conditions, $\hat{\beta}_2^{(i)} \sim B_2(t_i) := g'(t_i)$ when there is no jump in $N(t_i)$, where $g(t)$ is the continuous part of $f'(t)$. When there is a jump in $N(t_i)$ and the jump location is at $t_{i-l+r}, 0 \le r \le 2l$, then $\hat{\beta}_2^{(i)} \sim B_2(t_i) := g'_+(t_i) + h_3(r)C_1 - \gamma(r)C_2$, where C_1 and C_2 are jump magnitudes of $f'(t)$ and its first-order derivative $\gamma(r)$ is a positive function taking values in [0, 1],

$$h_3(r) := \frac{r(k-1-r)k[(k-1)(k-2) - 3r(k-1-r)]}{12(ks_4 - s_2^2)n^3},$$

where $s_p = \sum_{j=i-l}^{i+l} (t_j - t_i)^p, p = 2, 4$, and $g'_+(t_i) = g'(t_i)$ if $r \neq l$.

Example 3.1. Let $f'(t) = 5t^2 + I_{[.5,1]}(t)$. Consider a sample of size 100 and let k be 11. Then $\hat{\beta}_2^{(i)} \sim B_2(t_i) = 10t_i + I_{\{.45 \le t_i \le .55\}} h_3(100(.55 - t_i)), 6 \le i \le 95. \{B_2(t_i)\}$ is shown in Figure 3(a).

We construct $\{\Delta_2^{(i)}\}$ from $\{\hat{\beta}_2^{(i)}\}$ similar to (2.1)–(2.2) in Section 2, and we have $\{J_2(t_i)\}$, which is shown in Figure 3(b) in the case of Example 3.1. The threshold value for

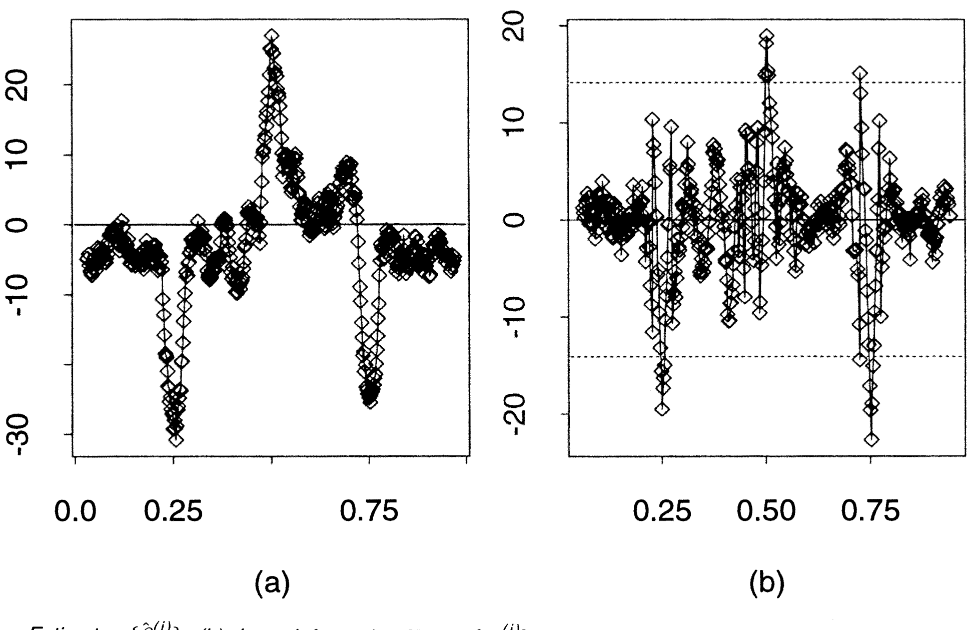


Figure 5. (a) Slope Estimates $\{\hat{\beta}_1^{(i)}\}$; (b) Jump Information Terms $\{\Delta_1^{(i)}\}$. The dotted lines in plot (b) indicate (+) or (−) threshold value u_1 corresponding to $Z_{\alpha_n/2}$ = 3.5.

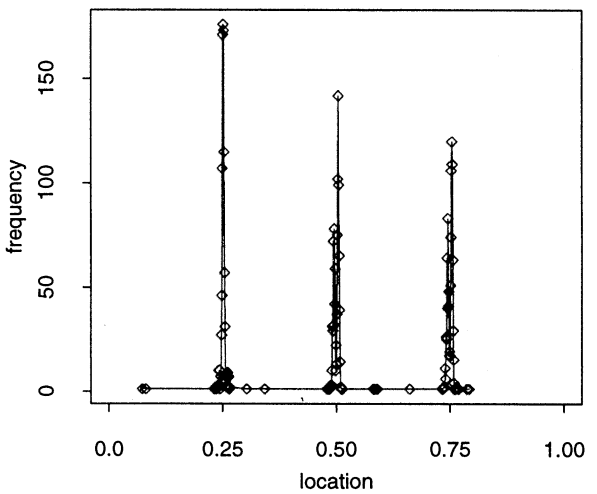


Figure 6. Frequency of Detected Jump Locations by the Zero-Order Jump-Detection Algorithm With n = 512 and k = 31 for 1,000 Replications.

$\{\Delta_2^{(i)}\}$ is derived in a similar way to that for $\{\Delta_1^{(i)}\}$, as

$$u_2 = \hat{\sigma} Z_{\alpha_n/2} \frac{\sqrt{k^2 s_4 - (k+1) s_2^2}}{k s_4 - s_2^2}. \tag{3.1}$$

The jump-detection algorithm in Section 2 can be used here with $\{\Delta_2^{(i)}\}$ and u_2 substituting for $\{\Delta_1^{(i)}\}$ and u_1.

4. NUMERICAL EXAMPLES

4.1 Jumps in Mean Response

We conducted some simulations using the example from Hall and Titterington (1992), the data from which is shown in Figure 4. Five hundred and twelve observations $\{Y_i\}$ are obtained from $f(t_i) + \varepsilon_i$ for equally spaced $t_i = i/512$, with errors from $\mathrm{N}(0, \sigma^2)$ and $\sigma = .25$. The regression function is $f(t) = 3 - 4t$ when $0 \le t \le .25$, $f(t) = 2 - 4t$ when $.25 < t \le .5$, $f(t) = -1 + 4t$ when $.5 < t \le .75$, and $f(t) = 4 - 4t$ when $.75 < t \le 1$. $f(t)$ has three jumps, at .25, .5 and .75, and corresponding jump magnitudes of $-1, 1$, and -1, respectively.

We use the zero-order jump-detection algorithm to detect the jumps, initially with $k = 31$. $\{\hat{\beta}_1^{(i)}\}$ is shown in Figure 5(a). According to the discussions in Section 2, $\hat{\beta}_1^{(i)} \sim B_1(t_i)$, including a continuous factor and a jump factor. The jump factor has its effect only in the neighborhoods of the jump locations. The continuous factor is negative in

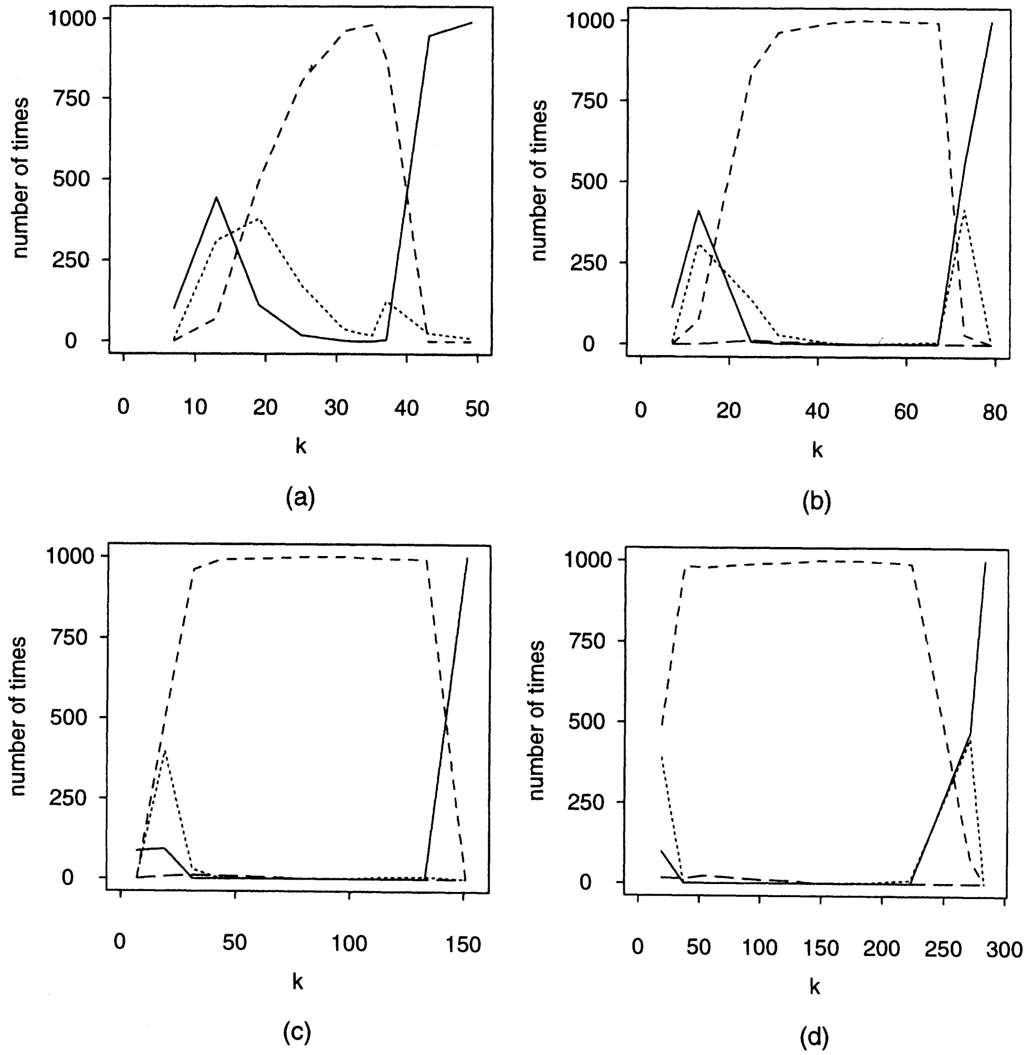


Figure 7. Distributions of the Number of Jumps Detected by the Zero-Order Jump-Detection Algorithm in 1,000 Replications for Several n and k. (a) n = 256; (b) n = 512; (c) n = 1,024; (d) n = 2,048: —, jumps detected = 1; ----, jumps detected = 2; – – –, jumps detected = 3; — —, jumps detected = 4.

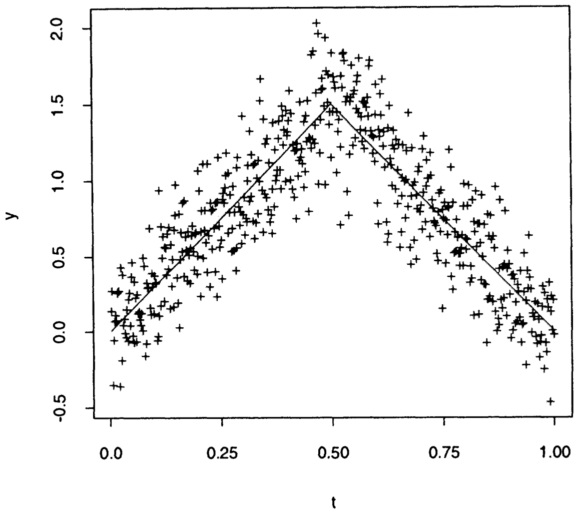


Figure 8. Hall and Titterington Function f (solid lines) and Observations (+).

intervals $0 \le t \le .25, .25 < t \le .5$, and $.75 < t \le 1$ and positive in the interval $.5 < t \le .75$. All of these features are evident from Figure 5(a). We then use operator (2.1) to remove the continuous factors from $\{\hat{\beta}_1^{(i)}\}$. The remaining jump factors $\{\Delta_1^{(i)}\}$ are shown in Figure 5(b). From this graph, we can see that $\{\Delta_1^{(i)}\}$ is approximately 0 when t_i is not in the neighborhoods of the jumps. This shows that the continuous factor is mostly removed. As we noticed in Remark 2.4, $\{\Delta_1^{(i)}\}$ seems noisier than $\{\hat{\beta}_1^{(i)}\}$. Figure 5(a) reveals the jumps very well in this case because the continuous factor does not contaminate much of the jump information. From the graphs we can approximate the jump positions and estimate the jump magnitudes from the relationship $M \sim h_2((k-1)/2)C_0 = [1.5n(k-1)/k(k+1)]C_0$, where M is the maximum/minimum value of $\{\Delta_1^{(i)}\}$ in the neighborhood of each of the detected jump positions and C_0 is the corresponding jump magnitude. For example, in Figure 5(b), we can see that there is a jump near .75 and that M is about -23. Thus, the jump magnitude C_0 is about $C_0 \approx [1.5n(k-1)/k(k+1)]^{-1}M \approx -.99$.

As we noted in Section 2, if the noise is not taken into account, $\{\Delta_1^{(i)}\}$ has a peak with value $[1.5n(k-1)/k(k+1)]C_0$ at a jump point with jump magnitude C_0. Comparing with the threshold value in (2.3), this jump could be detected if the jump magnitude satisfies

$$C_0 > \frac{\hat{\sigma} Z_{\alpha_n/2}(6(k+1)(5k-3))^{1/2}}{1.5(k-1)^{3/2}}. \tag{4.1}$$

Although n is not explicitly in (4.1), it is actually hidden in k because the preceding arguments are true only in the case that k/n is small and n is large. Equation (4.1) tells us that (a) if σ is bigger (the data are noisier), then only jumps with larger jump magnitudes could be detected; (b) if the confidence level is set higher (α_n is small and $Z_{\alpha_n/2}$ is large), then the algorithm is more conservative (jumps with small jump magnitudes would probably be missed); (c) if k is larger (n is also larger), the algorithm could detect jumps with smaller magnitudes or could detect the same jumps with higher confidence level. This last point also implies that, to detect the same jump, the confidence level could be set higher when the sample size is larger.

In our simulation based on this example, we use $n = 512$ and $k = 31$. The peak value of $\{\Delta_1^{(i)}\}$ is about 23.2258. The variance of $\{\Delta_1^{(i)}\}$ is less than 4.0245 [cf. the derivation of (2.3)]. We choose $Z_{\alpha_n/2} = 3.5$ in the threshold, which corresponds to $\alpha_n = .0004$. By (2.3), the threshold value is 14.08, which is smaller than the peak value by more than 2 times the variance of $\{\Delta_1^{(i)}\}$. By (4.1), the algorithm could detect jumps with minimum jump magnitude of about .6. If $k = 49$, which is an optimal choice as discussed later, then this minimum magnitude could be decreased to about .47.

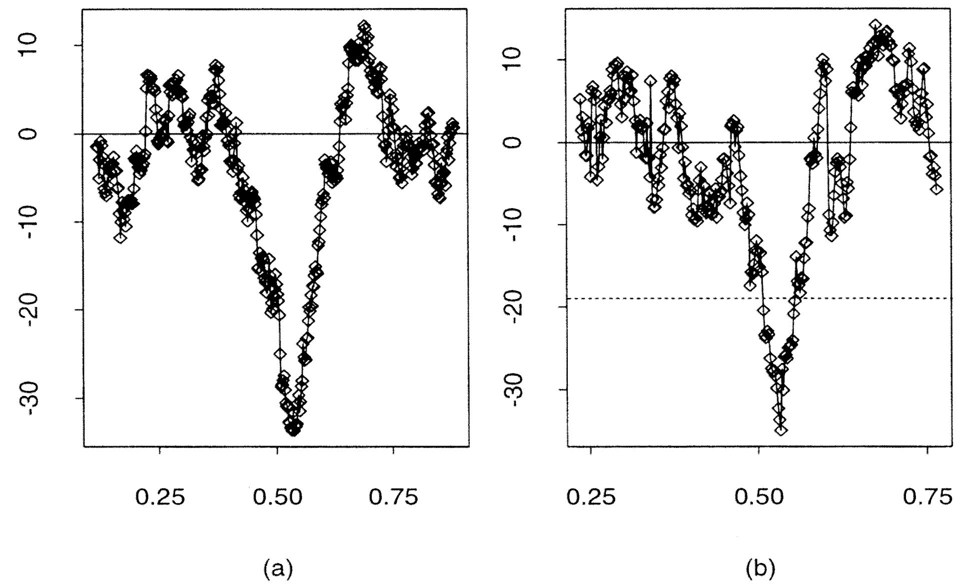


Figure 9. (a) $\{\hat{\beta}_2^{(i)}\}$ and (b) $\{\Delta_2^{(i)}\}$ of the First-Order Jump-Detection Algorithm. The dotted line in plot (b) indicates (−) threshold value u_2 corresponding to $Z_{\alpha_n/2} = 3.5$.

One thousand independent trials were performed. Three jumps were detected in 963 of these trials, two jumps were detected in 29, and one jump was detected once. The detected jump locations are shown in Figure 6.

The choice of $k = 31$ is somewhat arbitrary. We investigated this with simulations for several n and k. The results, summarized in Figure 7, show that, if k is chosen very small, some jumps are frequently missed because the noise of $\Delta_1^{(i)}$ swamps the jump information. If k is very large, the wide window width contaminates the jump information with effects of the continuous factors. The corresponding results are not impressive either. We have calculated ratios of the "best" window widths (the smallest window widths that give the best results) to the sample sizes to be $35/256 > 49/512 > 79/1{,}024 > 145/2{,}048$. This suggests that the ratio of the window size to the sample size should be decreasing as n increases.

4.2 Jump Detection in Slope

Some simulation results of jump detection in the first-order derivative are now presented. We also use the example of Hall and Titterington (1992), which is shown in Figure 8. Five hundred and twelve observations $\{Y_i\}$ are obtained from $f(t_i) + \varepsilon_i$ for equally spaced $t_i = i/512$, with iid errors from $\mathrm{N}(0, \sigma^2)$ and $\sigma = .25$. $f(t) = 3t$ when $0 \le t \le .5$ and $f(t) = 3 - 3t$ when $.5 < t \le 1$. Thus, F has one first-order jump at $t = .5$.

When $k = 121, \{\hat{\beta}_2^{(i)}\}$ and $\{\Delta_2^{(i)}\}$ are shown in Figure 9, (a) and (b), respectively. We performed our simulations with a variety of k and n. In each case, 1,000 replications were used. Part of the results are presented in Figure 10.

Comparing Figure 7 with Figure 10, we find that the window width k should be chosen larger for slope-change detection. The ratios of the "best" window widths to the sample sizes here are $107/256 > 185/512 > 285/1{,}024 > 469/2{,}048$, also decreasing as n increases. From Figure 10, it seems that k should be chosen as large as possible, but the boundary problem is more serious with larger k. Thus, there is a trade-off on this issue. We plot the detected jump locations in 1,000 replications with $k = 181$ and $n = 512$ in

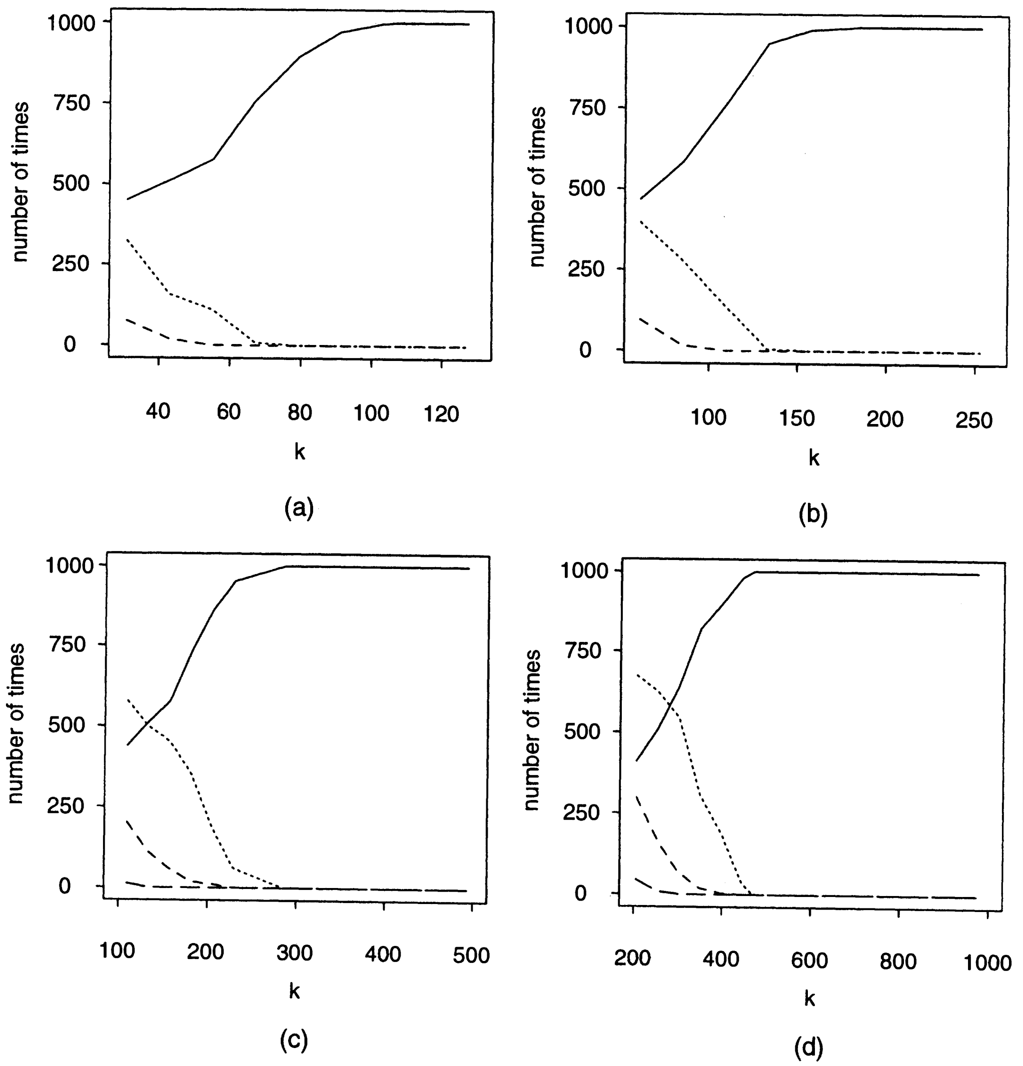


Figure 10. Distributions of the Number of Jumps Detected by the First-Order Jump-Detection Algorithm in 1,000 Replications for Several k and n. (a) n = 256; (b) n = 512; (c) n = 1,024; (d) n = 2,048: —, jumps detected = 1; - - - -, jumps detected = 2; — — —, jumps detected = 3; — —, jumps detected = 4.

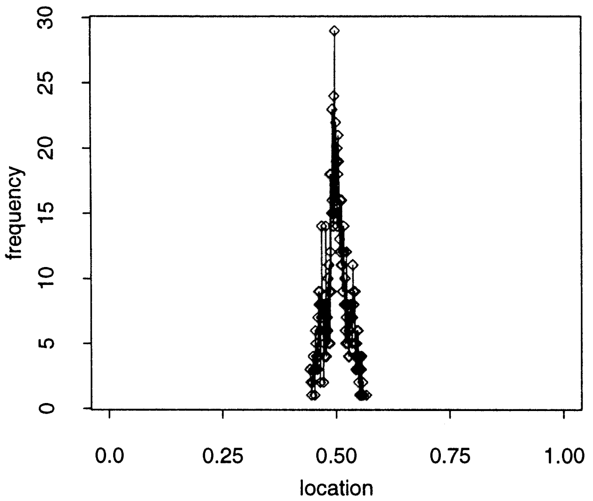


Figure 11. Frequency of Detected Jump Locations by the First-Order Jump-Detection Algorithm With n = 512 and k = 181 for 1,000 Replications

Figure 11. Comparing this graph with Figure 6, we can see that it is more difficult to detect jumps in derivatives than in the regression function itself.

4.3 The Sea-level Pressure Data Revisited

In Figure 1, we use $k = 15$ in both methods, suggested by the decreasing ratios of the "best" window widths to the sample sizes, as we found in Figure 7. With this window size, values of the jump-detection criterion are shown in Figure 12(b). A fitted polynomial regression function of order 4 has S.D. $\hat{\sigma} = .977$, leading to a jump threshold value $u_1 = 16.8$ for significance level .01. From the results, only $|\Delta_1^{(40)}| = |-18.077|$ (which corresponds to year 1960) exceeds u_1. Hence, a jump appears to exist at year 1960 with significance level .01. The slope estimates $\{\hat{\beta}_1^{(i)}\}$ are shown in Figure 12(a). This plot reveals the jump around year 1960 very clearly.

5. COMPARISON WITH THE KERNEL-TYPE METHODS

5.1 Some Background

Here we briefly introduce some kernel-type methods and compare their strengths and limitations relative to those of our algorithm. These considerations are important for practitioners when choosing an appropriate method for a specific application.

The methods suggested by Müller (1992) and Qiu, Asano, and Li (1991) were developed under the assumption that there is only one jump point. Let

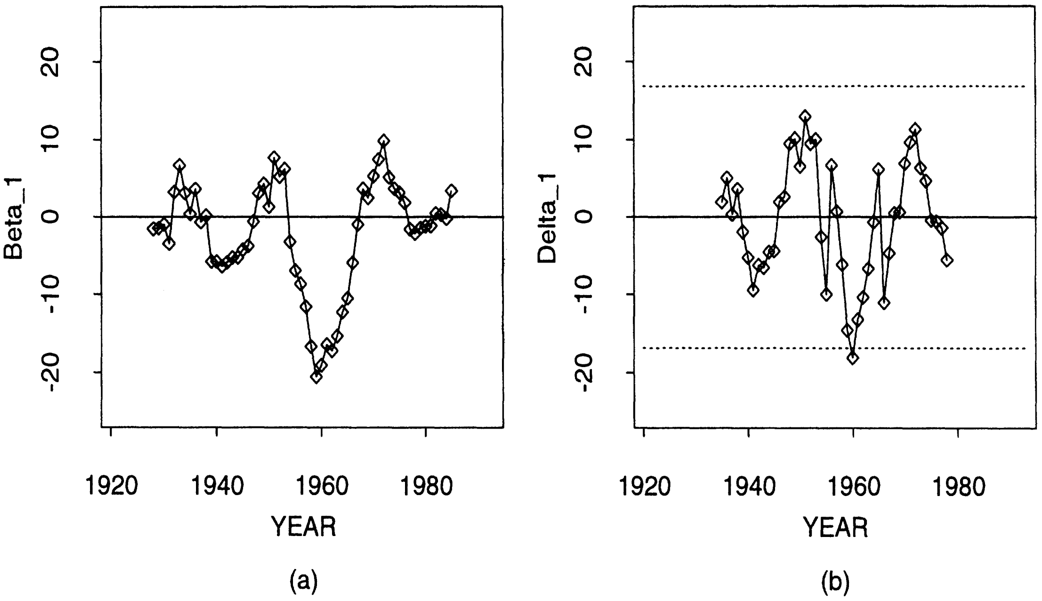
$$J(t) = \hat{m}_1(t) - \hat{m}_2(t) \tag{5.1}$$

and

$$|J(\hat{s})| = \max_{0 \leq t \leq 1} |J(t)|,$$

where $\hat{m}_1(t)$ and $\hat{m}_2(t)$ are two kernel estimators of the regression function $f(t)$ defined by a bandwidth h and two kernel functions $K_1(x)$ and $K_2(x)$ satisfying $K_1(x) = K_2(-x)$. Then $\hat{s}$ and $|J(\hat{s})|$ are defined as the estimators of the jump position and the corresponding jump magnitude, respectively. Qiu (1994) generalized these methods to the case with an unknown number of jumps but required that the jump magnitudes have a known lower bound.

Wu and Chu (1993) proposed a method to detect jumps when the number of jumps is unknown. Their proposal consisted of several steps. First, a series of hypothesis tests is performed for $H_0 : p = j$ versus $H_a : p > j$ until an acceptance is encountered where $j \geq 0$ and p is the true number of jump points. Then p maximizers $\{\hat{s}_j\}_{j=1}^{p}$ of $J(t)$ are defined as the jump position estimators. Finally, they used a rescaled $S(\hat{s}_j)$ to estimate the jump magnitude $d_j - d_{j-1}$



(a)

(b)

Figure 12. (a) Slope Estimates $\{\hat{\beta}_1^{(i)}\}$ of the Bombay (India) Sea-Level Pressure Data; (b) Values of the Jump-Detection Criterion. The dotted lines in plot (b) indicate (+) or (-) threshold value u_1 corresponding to α_n – .01.

for $j = 1, 2, \ldots, p$, where

$$S(t) = \hat{m}_3(t) - \hat{m}_4(t), \tag{5.2}$$

where $\hat{m}_3(t)$ and $\hat{m}_4(t)$ are two new kernel estimators of $f(t)$ defined by a bandwidth g and kernel functions $K_3(x)$ and $K_4(x)$.

A main limitation of the jump-detection criteria (5.1) and (5.2) is that they do not take into account the derivatives in detecting jumps in the regression function. Suppose that $f(t)$ is steep but continuous around some point t^*. Then both $S(t^*)$ and $J(t^*)$ could be large because of large derivative values. In other words, (5.1) and (5.2) do not exclude the continuous information from the jump information, which has been considered in our criterion $\Delta_1^{(i)}$ (Sec. 2). Müller (1992) used high-order kernels to detect jumps in derivatives. Our algorithm simply fits local polynomials with coefficients directly related to the derivatives of the regression function.

5.2 An Example

A main purpose of this example is to show how the slope affects the performance of jump detection in the method of Wu and Chu (1993) and the method described here.

Consider the regression function $f(t) = c(.5 - t) + I_{[.5,1]}(t)$ having a single jump at $t = .5$ and slope c at continuous points. We choose $n = 512$ and $\sigma = .25$ as in Section 4. The regression function with $c = 4$ is displayed in Figure 13(a) along with its noisy version. We then apply the LS method and the method of Wu and Chu (1993) to detect the jump. Parameters in the LS procedure are chosen to be the same as those in Section 4. We use the same kernel functions as those reported in their simulation examples. The bandwidth h is chosen .06 ($\approx 31/512$), which is compatible with the window size used in the LS procedure. The bandwidth g is chosen $2h$, which was suggested by Wu and Chu (1993).

We let the value of c vary among 3.0, 3.2, 3.4, 3.6, 3.8, and 4.0. For each c value, the simulation is repeated 1,000 times. In each simulation, estimators of the number of jumps (the true value is 1) by both methods are recorded. The Wu and

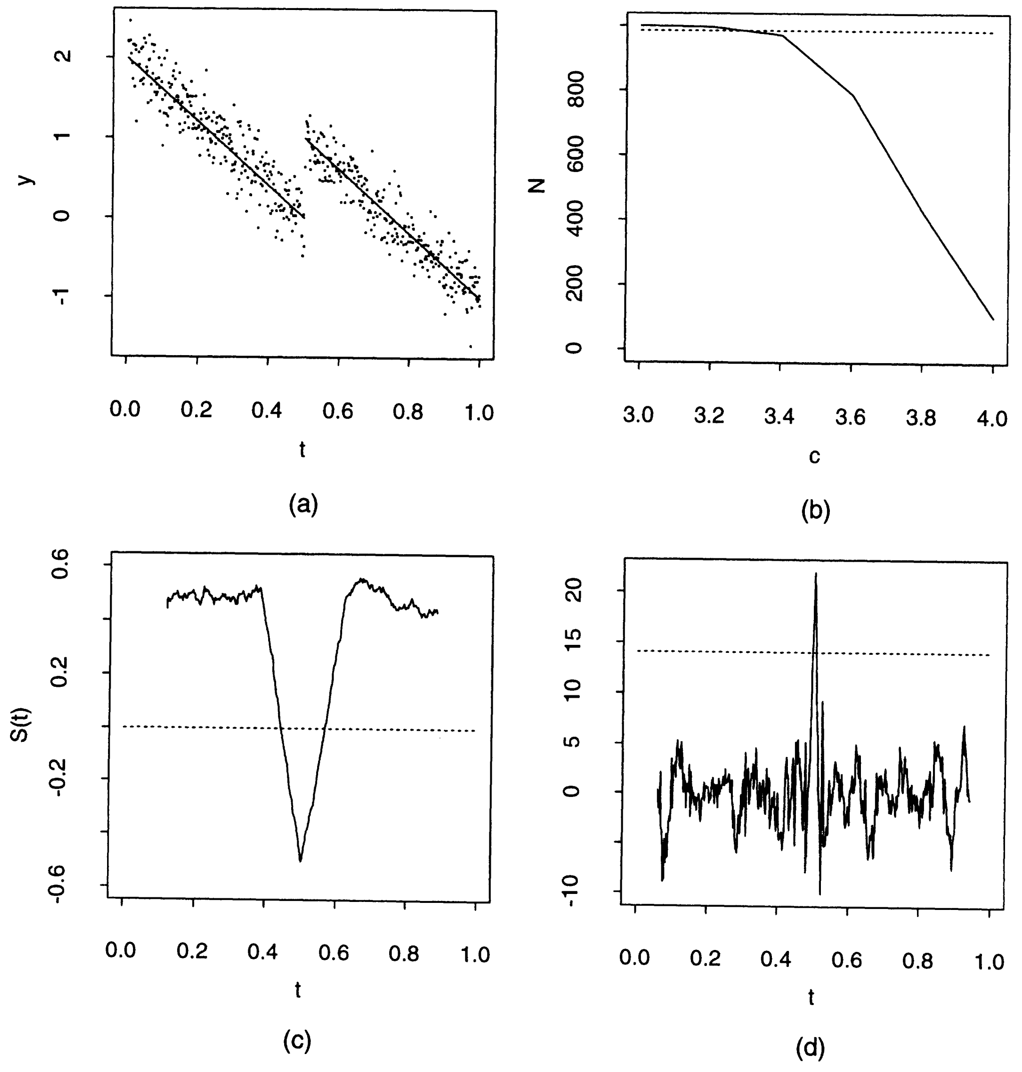


Figure 13. (a) Regression Function and its Noisy Version; (b) Numbers of Correct Estimations (N) of the Number of Jumps out of 1,000 Replications When the Slope c of the Regression Function Changes From 3.0 to 4.0, (—, Wu and Chu method; ----, LS method); (c) $\{S(t_i)\}$ of the Wu and Chu Procedure; (d) $\{\Delta_1^{(i)}\}$ of the LS Procedure. The dotted line in plot (c) indicates $S(t_i) = 0$. The dotted line in plot (d) denotes the threshold value of the LS procedure.

Chu procedure gives a correct result if it rejects H_0 for $H_0 : p = 0$ versus $H_a : p > 0$ and accepts H_0 for $H_0 : p = 1$ versus $H_a : p > 1$ at the same time. We then count the number of correct estimations from 1,000 replications for each method. The results are presented in Figure 13(b). It can be seen that the performance of the Wu and Chu procedure is worse when c becomes larger [$f(t)$ is steeper at continuous points]. The performance of our procedure, however, is quite stable. We plot $S(t_i)$ with $c = 4$ in Figure 13(c). It can be seen that $S(t_i)$ are relatively large at the continuous points because of large derivative values of $f(t)$. As a comparison, $\Delta_1^{(i)}$ [in plot (d)] values are approximately 0 at the continuous points and have large values around the true jump point.

6. CONCLUDING REMARKS

We have presented a jump-detection algorithm with local polynomial fitting that is intuitively appealing and simple to use. Limited simulations show that it has potential to work well in practice.

Possible future research includes (a) determination of the value of m, which is not considered in this article but may be important in applications; (b) selection of the window width k when the sample size is fixed, both from theoretical analysis and from simulation study; and (c) generalization of this method to multivariate cases, especially the jump surface case, which is directly related to many application areas such as image processing.

ACKNOWLEDGMENTS

The authors are grateful to the editor, the associate editor, and two anonymous referees whose helpful suggestions led to a great improvement of the presentation.

APPENDIX A: PROPERTIES OF THE ESTIMATED LS COEFFICIENTS

The following two theorems give some properties of the estimated LS coefficients used in Sections 2 and 3. Their proofs were given by Qiu (1996).

Theorem A.1. For Model (1.1)–(1.2), suppose that $m = 0, g(t)$, the continuous part of $f(t)$, has continuous first-order derivative over (0, 1) except on the jump points at which it has the first-order right and left derivatives. Let the window width k satisfy the conditions that $\lim_{n\to\infty} k = \infty$ and $\lim_{n\to\infty} k/n = 0$. Then $\hat{\beta}_1^{(i)}$ in Model (1.3) has the following properties. If there is no jump in $N(t_i)$, then

$$\hat{\beta}_1^{(i)} = g'(t_i) + O\left(\frac{n\sqrt{\log\log k}}{k^{3/2}}\right), \text{ a.s.}$$

If there is a jump in $N(t_i)$ and the jump location is at $t_{i-l+r}, 0 \le r \le 2l$, then

$$\hat{\beta}_1^{(i)} = g'_+(t_i) + h_1(r)C_0 - \gamma(r)C_1 + O\left(\frac{n\sqrt{\log\log k}}{k^{3/2}}\right), \text{ a.s.,}$$

where C_0 and C_1 are the jump magnitudes of $f(t)$ and its first-order derivative at the jump location, $\gamma(r)$ is a positive function taking values between 0 and 1, $h_1(r) := [6nr/k(k+1)](1 - (r/k - 1))C_0$, and $g'_+(t_i) = g'(t_i)$ if $r \neq l$.

Remark A.1. The term $O(n\sqrt{\log\log k}/k^{3/2})$ in Theorem A.1 is due to noise in the model.

Theorem A.2. For Model (1.1)–(1.2), suppose that $m = 1$ and $g(t)$, the continuous part of $f'(t)$, has continuous first-order derivative over (0, 1) except on the jump points at which it has the first-order right and left derivatives. The window width k satisfies the conditions that $\lim_{n\to\infty} k = \infty$ and $\lim_{n\to\infty} k/n = 0$. Then $\hat{\beta}_2^{(i)}$ in Model (1.3) has the following properties. If there is no jump in $N(t_i)$, then

$$\hat{\beta}_2^{(i)} = g'(t_i) + O\left(\frac{n^2\sqrt{\log\log k}}{k^{5/2}}\right), \text{ a.s.}$$

If there is a jump in $N(t_i)$ and the jump location is at $t_{i-l+r}, 0 \le r \le 2l$, then

$$\hat{\beta}_2^{(i)} = g'_+(t_i) + h_3(r)C_1 - \gamma(r)C_2 + O\left(\frac{n^2\sqrt{\log\log k}}{k^{5/2}}\right), \text{ a.s.}$$

where C_1 and C_2 are the jump magnitudes of $f'(t)$ and its first-order derivative, $\gamma(r)$ is a positive function taking values in [0, 1],

$$h_3(r) := \frac{r(k-1-r)k[(k-1)(k-2) - 3r(k-1-r)]}{12(ks_4 - s_2^2)n^3},$$

$s_p = \sum_{j=i-l}^{i+l}(t_j - t_i)^p, p = 2, 4, g'_+(t_i) = g'(t_i)$ if $r \neq l$.

APPENDIX B: PROOF OF THEOREM 2.1

For design point $t_i \in (0,1)$, if $|t_i - s_j| > (k+1)/2n$ for any $j = 1, 2, \ldots, p$, then by Theorem A.1,

$$\Delta_1^{(i)} = O\left(\frac{n\sqrt{\log\log k}}{k^{3/2}}\right), \text{ a.s.,} \tag{B.1}$$

where $\{s_j, j = 1, 2, \ldots, p\}$ are the true jump positions as we defined in (1.2). On the other hand, if t_i is a jump point, then

$$\Delta_1^{(i)} \sim h_2(l)C_0 = O\left(\frac{n}{k}\right), \text{ a.s.} \tag{B.2}$$

By (2.3), the threshold value is

$$\begin{aligned} u_1 &= \hat{\sigma} Z_{\alpha_n/2} \frac{n}{k}\sqrt{\frac{6(5k-3)}{k^2-1}} \\ &= O\left(\frac{nZ_{\alpha_n/2}}{k^{3/2}}\right). \end{aligned} \tag{B.3}$$

Combining (B.1)–(B.3) and by the conditions stated in the theorem, the flagged design points $\{t_{i_j}, j = 1, 2, \ldots, n_1\}$ before the modification procedure satisfy

$$\{t_{i_j}, j = 1, 2, \ldots, n_1\} \subset \bigcup_{j=1}^{p} N(s_j), \tag{B.4}$$

where $N(s_j)$ is the neighborhood of s_j as we defined in Section 1.

After we use the modification procedure to delete some deceptive jump candidate points, we know that in each neighborhood $N(s_j)$ there is one and only one point of the final jump candidate set $\{b_j, j = 1, 2, \ldots, q_1\}$. Hence, $\lim_{n\to\infty} q_1 = p$, a.s., and $|b_j - s_j| = o(k/n)$, a.s. If we choose $k = \log n$, then the conclusion of the theorem is obtained.

[Received December 1994. Revised September 1997.]

REFERENCES

Besag, J., Green, P., Higdon, D., and Mengersen, K. (1995), "Bayesian Computation and Stochastic Systems" (with discussion), *Statistical Science*, 10, 3–66.

Eubank, R. L., and Speckman, P. L. (1994), "Nonparametric Estimation of Functions With Jump Discontinuities," in *Change-Point Problems* (IMS Lecture Notes, vol. 23), eds. E. Carlstein, H. G. Müller, and D. Siegmund, Hayward, CA: IMS, pp. 130–144.

Gonzalez, R. C., and Woods, R. E. (1992), *Digital Image Processing*, Reading, MA: Addison-Wesley.

Hall, P., and Titterington, M. (1992), "Edge-Preserving and Peak-Preserving Smoothing," *Technometrics*, 34, 429–440.

Härdle, W. (1991), *Smoothing Techniques: With Implementation in S*, New York: Springer-Verlag.

Hastie, T., and Tibshirani, R. (1987), "Generalized Additive Models: Some Applications," *Journal of the American Statistical Association*, 82, 371–386.

Loader, C. R. (1996), "Change Point Estimation Using Nonparametric Regression," *The Annals of Statistics*, 24, 1667–1678.

McDonald, J. A., and Owen, A. B. (1986), "Smoothing With Split Linear Fits," *Technometrics*, 28, 195–208.

Müller, H. G. (1992), "Change-points in Nonparametric Regression Analysis," *The Annals of Statistics*, 20, 737–761.

Qiu, P. (1991), "Estimation of a Kind of Jump Regression Functions," *Systems Science and Mathematical Sciences*, 4, 1–13.

——— (1994), "Estimation of the Number of Jumps of the Jump Regression Functions," *Communications in Statistics—Theory and Methods*, 23, 2141–2155.

——— (1996), "Nonparametric Estimation of Discontinuous Regression Functions," unpublished Ph.D. thesis, University of Wisconsin-Madison, Dept. of Statistics.

Qiu, P., Asano, C., and Li, X. (1991), "Estimation of Jump Regression Functions," *Bulletin of Informatics and Cybernetics*, 24, 197–212.

Qiu, P., and Bhandarkar, S. M. (1996), "An Edge Detection Technique Using Local Smoothing and Statistical Hypothesis Testing," *Pattern Recognition Letters*, 17, 849–872.

Qiu, P., and Yandell, B. (1997), "Jump Detection in Regression Surfaces," *Journal of Computational and Graphical Statistics*, 6, 332–354.

Shea, D. J., Worley, S. J., Stern, I. R., and Hoar, T. J. (1994), "An Introduction to Atmospheric and Oceanographic Data," Technical Note NCAR/TN-404+IA, Climate and Global Dynamics Division, National Center For Atmospheric Research, Boulder, Colorado.

Shiau, J. H. (1987), "A Note on MSE Coverage Intervals in a Partial Spline Model," *Communications in Statistics—Theory and Methods*, 16, 1851–1866.

Speckman, P. L. (1993), "Detection of Change-points in Nonparametric Regression," unpublished manuscript, University of Missouri, Dept. of Statistics.

Stone, C. J. (1977), "Consistent Nonparametric Regression," *The Annals of Statistics*, 5, 595–620.

Wahba, G. (1986), "Partial Spline Modelling of the Tropopause and Other Discontinuities," in *Function Estimate, Contemporary Mathematics 59*, ed. J. S. Marron, Providence, RI: AMS, pp. 125–135.

——— (1991), *Spline Models for Observational Data*, Philadelphia: SIAM.

Wand, M. P., and Jones, M. C. (1995), *Kernel Smoothing*, London: Chapman & Hall.

Wu, J. S., and Chu, C. K. (1993), "Kernel Type Estimators of Jump Points and Values of a Regression Function," *The Annals of Statistics*, 21, 1545–1566.

Yin, Y. Q. (1988), "Detecting of the Number, Locations and Magnitudes of Jumps," *Communications in Statistics—Stochastic Models*, 4, 445–455.