Structural Break Estimation for Nonstationary Time Series Models

Richard A. DAVIS, Thomas C. M. LEE, and Gabriel A. RODRIGUEZ-YAM

This article considers the problem of modeling a class of nonstationary time series using piecewise autoregressive (AR) processes. The number and locations of the piecewise AR segments, as well as the orders of the respective AR processes, are assumed unknown. The minimum description length principle is applied to compare various segmented AR fits to the data. The goal is to find the "best" combination of the number of segments, the lengths of the segments, and the orders of the piecewise AR processes. Such a "best" combination is implicitly defined as the optimizer of an objective function, and a genetic algorithm is implemented to solve this difficult optimization problem. Numerical results from simulation experiments and real data analyses show that the procedure has excellent empirical properties. The segmentation of multivariate time series is also considered. Assuming that the true underlying model is a segmented autoregression, this procedure is shown to be consistent for estimating the location of the breaks.

KEY WORDS: Changepoint; Genetic algorithm; Minimum description length principle; Nonstationarity.

1. INTRODUCTION

In this article we consider the problem of modeling a nonstationary time series by segmenting the series into blocks of different autoregressive (AR) processes. The number of breakpoints, denoted by m, as well as their locations and the orders of the respective AR models are assumed unknown. We propose an automatic procedure for obtaining such a partition.

To describe the setup, for $j = 1, \ldots, m$, denote the breakpoint between the jth and $(j+1)$st AR processes as τ_j, and set $\tau_0 = 1$ and $\tau_{m+1} = n + 1$. Then the jth piece of the series is modeled as an AR process,

$$Y_t = X_{t,j}, \qquad \tau_{j-1} \le t < \tau_j, \tag{1}$$

where $\{X_{t,j}\}$ is the AR(p_j) process

$$X_{t,j} = \gamma_j + \phi_{j1} X_{t-1,j} + \cdots + \phi_{j,p_j} X_{t-p_j,j} + \sigma_j \varepsilon_t,$$

$\boldsymbol{\psi}_j := (\gamma_j, \phi_{j1}, \ldots, \phi_{j,p_j}, \sigma_j^2)$ is the parameter vector corresponding to this AR(p_j) process, and the noise sequence $\{\varepsilon_t\}$ is iid with mean 0 and variance 1. Given an observed series $\{y_i\}_{i=1}^n$, the objective is to obtain a "best"-fitting model from this class of piecewise AR processes. This is equivalent to finding the "best" combination of the number of pieces $m + 1$, the breakpoint locations $\tau_1, \ldots, \tau_m$, and the AR orders $p_1, \ldots, p_{m+1}$. We propose an automatic piecewise autoregressive modeling procedure, referred to as Auto-PARM, for obtaining such a partition.

Note that once these parameters are specified, maximum likelihood estimates of the AR parameters, $\boldsymbol{\psi}_j$, for each segment are easily computed. The primary objective of the methodology developed in this article is to actually estimate structural breaks for a time series. Under this scenario, it is assumed that some aspect of a time series changes at various times; such a change might be a shift in the mean level of the process, a change in variance, and/or a change in the dependence structure of the process. The sequence of time series between two changepoints is assumed to be modeled as a sequence of stationary processes, each of which can be adequately modeled by an AR process. Potential applications of this setup can be found in social sciences, in which time series may be impacted by changes in government policies and time series from signal processing, engineering, and manufacturing, where production processes are often subject to unpredictable changes in the manufacturing process.

As a secondary objective, our methodology can also be viewed as a procedure for approximating locally stationary time series by piecewise AR processes. To see this, note that the piecewise AR process considered in (1) is a special case of the piecewise stationary process (see also Adak 1998)

$$\tilde{Y}_{t,n} = \sum_{j=1}^{m+1} X_{t,j} I_{[\tau_{j-1}/n, \tau_j/n)}(t/n),$$

where $\{X_{t,j}\}, j = 1, \ldots, m+1$, is a sequence of stationary process. Ombao, Raz, Von Sachs, and Malow (2001) argued that under certain conditions, locally stationary processes (in the sense of Dahlhaus 1997) can be well approximated by piecewise stationary processes. Roughly speaking, a process is locally stationary if its time-varying spectrum at time t and frequency ω is $|A(t/n, \omega)|^2$, where $A(u, \omega)$, $u \in [0, 1]$, $\omega \in [-1/2, 1/2]$, is a continuous function in u. Because AR processes are dense in the class of weakly stationary (purely nondeterministic) processes, the piecewise AR process is *dense* in the class of locally stationary processes.

The foregoing problem of finding a "best" combination of m, τ_j's, and p_j's can be treated as a statistical model selection problem, in which candidate models may have different numbers of parameters. To solve this selection problem, we apply the minimum description length (MDL) principle of Rissanen (1989) to define a best-fitting model. (See Saito 1994 and Hansen and Yu 2000 for comprehensive reviews of MDL.) The basic idea behind the MDL principle is that the best-fitting model is the one that enables maximum compression of the data. Successes in applying MDL to a various practical problems have been widely reported in the literature (e.g., Lee 2000; Hansen and Yu 2001; Jornsten and Yu 2003).

Richard Davis is Professor (E-mail: *rdavis@stat.colostate.edu*), Thomas Lee is Associate Professor (E-mail: *tlee@stat.colostate.edu*), and Gabriel Rodriguez-Yam is Postdoctoral Fellow (E-mail: *rodrigue@stat.colostate.edu*), Department of Statistics, Colorado State University, Fort Collins, CO 80523. This research was supported in part by National Science Foundation grants DMS-03-08109 (Davis) and DMS-02-03901 (Lee), by an IBM Faculty Research Award, and by EPA STAR grant CR-829095. The authors thank Hernando Ombao for sharing his computer code implementation of Auto-SLEX. They also thank the associate editor and the referee for their constructive comments and suggestions, most of which were incorporated into the final manuscript.

Journal of the American Statistical Association
March 2006, Vol. 101, No. 473, Theory and Methods
DOI 10.1198/016214505000000745

As demonstrated later, the best-fitting model derived by the MDL principle is defined implicitly as the optimizer of some criterion. Practical optimization of this criterion is not a trivial task, because the search space (consisting of m, τ_j's and p_j's) is enormous. To tackle this problem, we use a genetic algorithm (GA), described by, for example, Holland (1975). GAs are becoming popular tools in statistical optimization applications (e.g., Gaetan 2000; Pittman 2002; Lee and Wong 2003) and seem particularly well suited for our MDL optimization problem, as can be seen in our numerical studies.

Various versions of the aforementioned breakpoint detection problem have been considered in the literature. For example, Bai and Perron (1998, 2003) examined the multiple changepoint modeling for the case of multiple linear regression, Inclan and Tiao (1994) and Chen and Gupta (1997) considered the problem of detecting multiple variance changepoints in a sequence of independent Gaussian random variables, and Kim and Nelson (1999) provided a summary of various applications of the hidden Markov approach to econometrics. Kitagawa and Akaike (1978) implemented an "on-line" procedure based on the Akaike information criterion (AIC) to determine segments. To implement their method, suppose that an AR(p_0) model has been fitted to the dataset $\{y_1, y_2, \ldots, y_{n_0}\}$ and that a new block $\{y_{n_0+1}, \ldots, y_{n_0+n_1}\}$ of n_1 observations becomes available, which can be modeled as an AR(p_1) model. Then the time n_0 is considered a breaking point when the AIC value of the two independent pieces is smaller than the AIC of the AR that results when the dataset $\{y_1, \ldots, y_{n_0+n_1}\}$ is modeled as a single AR model of order p_2. Each $p_j, j = 0, 1, 2$, is selected among the values $0, 1, \ldots, K$ (where K is a predefined value) that minimizes the AIC. The iteration is continued until no more data are available. Like K, n_1 is a predefined value.

Ombao et al. (2001) implemented a segmentation procedure using the SLEX transformation, a family of orthogonal transformations. For a particular segmentation, a "cost" function is computed as the sum of the costs at all of the blocks that define the segmentation. The best segmentation is then defined as the one with minimum cost. Again, because it is not computationally feasible to consider all possible segmentations, Ombao et al. (2001) assume that the segment lengths follow a dyadic structure, that is, an integer power of 2. Bayesian approaches have also been studied (see, e.g., Lavielle 1998; Punskaya, Andrieu, Doucet, and Fitzgerald 2002). Both of these procedures choose the final optimal segmentation as the one that maximizes the posterior distribution of the observed series. Numerical results suggest that both procedures have excellent empirical properties; however, theoretical results supporting these procedures are lacking.

For most of the aforementioned procedures, including Auto-PARM, the "best" segmentation is defined as the optimizer of an objective function. Sequential-type searching algorithms for locating such an optimal segmentation are adopted by some of these procedures (e.g., Kitagawa and Akaike 1978; Inclan and Tiao 1994; Ombao et al. 2001). On the one hand, one would expect that these sequential procedures, when compared with our GA approach, would require less computational time to locate a good approximation to the true optimizer. On the other hand, because the GA approach examines a much bigger portion of the search space for the optimization, one should also expect the GA approach to provide better approximations to the true optimizer. A detailed comparison of the Auto-PARM procedure and the Auto-SLEX procedure of Ombao et al. (2001) is given in Section 4.

The rest of this article is organized as follows. In Section 2 we derive an expression for the MDL for a given piecewise AR model. In Section 3 we give an overview of the GA and discuss its implementation to the segmentation problem. In Section 4 we study the performance of the GA via simulation, and in Section 5 we apply the GA to two test datasets that have been used in the literature. We discuss the case of a multivariate time series and an application in Section 6. In Section 7 we summarize our findings and discuss the relative merits of Auto-PARM and other structural break detection procedures. Finally, we provide some theoretical results supporting our procedure in the Appendix.

2. MODEL SELECTION USING MINIMUM DESCRIPTION LENGTH

2.1 Derivation of Minimum Description Length

This section applies the MDL principle to select a best-fitting model from the piecewise AR model class defined by (1). Denote this whole class of piecewise AR models by $\mathcal{M}$ and any model from this class by $\mathcal{F} \in \mathcal{M}$. In the current context, the MDL principle defines the "best"-fitting model from $\mathcal{M}$ as the one that produces the shortest code length that completely describes the observed data $\mathbf{y} = (y_1, y_2, \ldots, y_n)$. Loosely speaking, the code length of an object is the amount of memory space required to store the object. In the applications of MDL, one classical way to store $\mathbf{y}$ is to split $\mathbf{y}$ into two components, a fitted model $\hat{\mathcal{F}}$ plus the portion of $\mathbf{y}$ that is unexplained by $\hat{\mathcal{F}}$. This latter component can be interpreted as the residuals, denoted by $\hat{\mathbf{e}} = \mathbf{y} - \hat{\mathbf{y}}$, where $\hat{\mathbf{y}}$ is the fitted vector for $\mathbf{y}$. If $CL_{\mathcal{F}}(z)$ denotes the code length of object z using model $\mathcal{F}$, then we have the following decomposition:

$$CL_{\mathcal{F}}(\mathbf{y}) = CL_{\mathcal{F}}(\hat{\mathcal{F}}) + CL_{\mathcal{F}}(\hat{\mathbf{e}}|\hat{\mathcal{F}}),$$

where $CL_{\mathcal{F}}(\hat{\mathcal{F}})$ is the code length of the fitted model $\hat{\mathcal{F}}$ and $CL_{\mathcal{F}}(\hat{\mathbf{e}}|\hat{\mathcal{F}})$ is the code length of the corresponding residuals (conditional on the fitted model $\hat{\mathcal{F}}$). In short, the MDL principle suggests that a best-fitting piecewise AR model $\hat{\mathcal{F}}$ is the one that minimizes $CL_{\mathcal{F}}(\mathbf{y})$.

Now the task is to derive expressions for $CL_{\mathcal{F}}(\hat{\mathcal{F}})$ and $CL_{\mathcal{F}}(\hat{\mathbf{e}}|\hat{\mathcal{F}})$. We begin with $CL_{\mathcal{F}}(\hat{\mathcal{F}})$. Let $n_j := \tau_j - \tau_{j-1}$ denote the number of observations in the jth segment of $\hat{\mathcal{F}}$. Because $\hat{\mathcal{F}}$ is composed of m, τ_j's, p_j's, and $\hat{\boldsymbol{\psi}}_j$'s, we further decompose $CL_{\mathcal{F}}(\hat{\mathcal{F}})$ into

$$\begin{aligned} CL_{\mathcal{F}}(\hat{\mathcal{F}}) &= CL_{\mathcal{F}}(m) + CL_{\mathcal{F}}(\tau_1, \ldots, \tau_m) \\ &\quad + CL_{\mathcal{F}}(p_1, \ldots, p_{m+1}) \\ &\quad + CL_{\mathcal{F}}(\hat{\boldsymbol{\psi}}_1) + \cdots + CL_{\mathcal{F}}(\hat{\boldsymbol{\psi}}_{m+1}) \\ &= CL_{\mathcal{F}}(m) + CL_{\mathcal{F}}(n_1, \ldots, n_{m+1}) \\ &\quad + CL_{\mathcal{F}}(p_1, \ldots, p_{m+1}) \\ &\quad + CL_{\mathcal{F}}(\hat{\boldsymbol{\psi}}_1) + \cdots + CL_{\mathcal{F}}(\hat{\boldsymbol{\psi}}_{m+1}). \end{aligned}$$

The last expression was obtained by the fact that complete knowledge of $(\tau_1, \ldots, \tau_m)$ implies complete knowledge of

$(n_1, \ldots, n_{m+1})$, and vice versa. In general, to encode an integer I whose value is not bounded, approximately $\log_2 I$ bits are needed. Thus $CL_{\mathcal{F}}(m) = \log_2 m$ and $CL_{\mathcal{F}}(p_j) = \log_2 p_j$. But if the upper bound (say I_U) of I is known, then approximately $\log_2 I_U$ bits are required. Because all n_j's are bounded by n, $CL_{\mathcal{F}}(n_j) = \log_2 n$ for all j. To calculate $CL_{\mathcal{F}}(\hat{\boldsymbol{\psi}}_j)$, we use the following result of Rissanen: A maximum likelihood estimate of a real parameter computed from N observations can be effectively encoded with $\frac{1}{2}\log_2 N$ bits. Because each of the $p_j + 2$ parameters of $\hat{\boldsymbol{\psi}}_j$ is computed from n_j observations,

$$CL_{\mathcal{F}}(\hat{\boldsymbol{\psi}}_j) = \frac{p_j + 2}{2}\log_2 n_j.$$

Combining these results, we obtain

$$CL_{\mathcal{F}}(\hat{\mathcal{F}}) = \log_2 m + (m+1)\log_2 n + \sum_{j=1}^{m+1}\log_2 p_j + \sum_{j=1}^{m+1}\frac{p_j+2}{2}\log_2 n_j. \quad (2)$$

Next, we derive an expression for $CL_{\mathcal{F}}(\hat{\mathbf{e}}|\hat{\mathcal{F}})$, that is, the code length for the residuals $\hat{\mathbf{e}}$. From Shannon's classical results in information theory, Rissanen demonstrated that the code length of $\hat{\mathbf{e}}$ is given by the negative of the log-likelihood of the fitted model $\hat{\mathcal{F}}$. To proceed, let $\mathbf{y}_j := (y_{\tau_{j-1}}, \ldots, y_{\tau_j - 1})$ be the vector of observations for the jth piece in (1). For simplicity, we consider that $\boldsymbol{\mu}_j$, the mean of the jth piece in (1), is $\mathbf{0}$. Denote the covariance matrix of $\mathbf{y}_j$ as $\mathbf{V}_j^{-1} = \mathrm{cov}\{\mathbf{y}_j\}$, and let $\hat{\mathbf{V}}_j$ be an estimate for $\mathbf{V}_j$. Even though the ε_j's are not assumed to be Gaussian, inference procedures are based on a Gaussian likelihood. Such inference procedures are often called *quasi-likelihood*. Assuming that the segments are independent, the Gaussian likelihood of the piecewise process is given by

$$L(m, \tau_0, \tau_1, \ldots, \tau_m, p_1, \ldots, p_{m+1}, \boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_{m+1}; \mathbf{y}) = \prod_{j=1}^{m+1}(2\pi)^{-n_j/2}|\mathbf{V}_j|^{1/2}\exp\left\{-\frac{1}{2}\mathbf{y}_j^T\mathbf{V}_j\mathbf{y}_j\right\},$$

and hence the code length of $\hat{\mathbf{e}}$ given the fitted model $\hat{\mathcal{F}}$ is

$$\begin{aligned} &CL_{\mathcal{F}}(\hat{\mathbf{e}}|\hat{\mathcal{F}}) \\ &\quad\approx -\log_2 L(m, \tau_0, \tau_1, \ldots, \tau_m, \hat{\boldsymbol{\psi}}_1, \ldots, \hat{\boldsymbol{\psi}}_{m+1}; \mathbf{y}) \\ &\quad= \sum_{j=1}^{m+1}\left\{\frac{n_j}{2}\log(2\pi) - \frac{1}{2}\log|\hat{\mathbf{V}}_j| + \frac{1}{2}\mathbf{y}_j^T\hat{\mathbf{V}}_j\mathbf{y}_j\right\}\log_2 e. \end{aligned} \quad (3)$$

Combining (2) and (3) and using logarithm base e rather than base 2, we obtain the approximation for $CL_{\mathcal{F}}(\mathbf{y})$,

$$\log m + (m+1)\log n + \sum_{j=1}^{m+1}\log p_j + \sum_{j=1}^{m+1}\frac{p_j+2}{2}\log n_j + \sum_{j=1}^{m+1}\left\{\frac{n_j}{2}\log(2\pi) - \frac{1}{2}\log|\hat{\mathbf{V}}_j| + \frac{1}{2}\mathbf{y}_j^T\hat{\mathbf{V}}_j\mathbf{y}_j\right\}. \quad (4)$$

Using the standard approximation to the likelihood for AR models [i.e., $-2\log(\text{likelihood})$ by $n_j \log\hat{\sigma}_j^2$, where $\hat{\sigma}_j^2$ is the Y–W estimate of σ_j^2 (Brockwell and Davis 1991)], we define

$$\begin{aligned} &\mathrm{MDL}(m, \tau_1, \ldots, \tau_m, p_1, \ldots, p_{m+1}) \\ &\quad= \log m + (m+1)\log n + \sum_{j=1}^{m+1}\log p_j \\ &\qquad+ \sum_{j=1}^{m+1}\frac{p_j+2}{2}\log n_j + \sum_{j=1}^{m+1}\frac{n_j}{2}\log(2\pi\hat{\sigma}_j^2). \end{aligned} \quad (5)$$

We propose selecting the best-fitting model for $\mathbf{y}$ as the model $\mathcal{F} \in \mathcal{M}$ that minimizes $\mathrm{MDL}(m, \tau_1, \ldots, \tau_m, p_1, \ldots, p_{m+1})$.

2.2 Consistency

To this point, we have not assumed the existence of a true model for the time series. But to study theoretical properties of these estimates, an underlying model must first be specified. Here we assume that there exist true values m_0 and λ_j^0, $j = 1, \ldots, m_0$, such that $0 < \lambda_1^0 < \lambda_2^0 < \cdots < \lambda_{m_0}^0 < 1$. The observations $y_1, \ldots, y_n$ are assumed to be a realization from the piecewise AR process defined in (1) with $\tau_i = [\lambda_i^0 n]$, $i = 1, 2, \ldots, m_0$, where $[x]$ is the greatest integer that is less than or equal to x. In estimating the breakpoints $\tau_1, \ldots, \tau_{m_0}$, it is necessary to require that the segments have a sufficient number of observations to adequately estimate the specified AR parameter values. Otherwise, the estimation is overdetermined, resulting in an infinite value for the likelihood. So to ensure sufficient separation of the breakpoints, choose $\epsilon > 0$ small such that $\epsilon \ll \min_{i=1,\ldots,m_0+1}(\lambda_i^0 - \lambda_{i-1}^0)$ and set

$$A_m = \big\{(\lambda_1, \ldots, \lambda_m), 0 < \lambda_1 < \lambda_2 < \cdots < \lambda_m < 1, \lambda_i - \lambda_{i-1} \geq \epsilon, i = 1, 2, \ldots, m+1\big\},$$

where $\lambda_0 := 0$ and $\lambda_{m+1} := 1$. Setting $\boldsymbol{\lambda} := (\lambda_1, \ldots, \lambda_m)$ and $\mathbf{p} = (p_1, \ldots, p_{m+1})$, the parameters m, $\boldsymbol{\lambda}$, and $\mathbf{p}$ are then estimated by minimizing MDL over $m \leq M_0$, $0 \leq \mathbf{p} \leq P_0$, and $\boldsymbol{\lambda} \in A_m$. That is,

$$\hat{m}, \hat{\boldsymbol{\lambda}}, \hat{\mathbf{p}} = \arg\min_{\substack{m \leq M_0, 0 \leq \mathbf{p} \leq P_0 \\ \boldsymbol{\lambda} \in A_m}} \frac{2}{n}\mathrm{MDL}(m, \boldsymbol{\lambda}, \mathbf{p}),$$

where M_0 and P_0 are upper bounds for m and p_j. In the Appendix we prove the following consistency result.

Proposition 1. For the model specified in (1), when m_0, the number of breakpoints, is known, then $\hat{\lambda}_j \to \lambda_j^0$, a.s., $j = 1, 2, \ldots, m_0$.

In Proposition 1, the true number of breaks, m_0, is assumed known. As the simulation studies in Section 4 show, for unknown m_0, the estimator $\hat{m}_0$ obtained with our procedure seems to be consistent, although we do not have a proof. Even in the independent case, the consistency of $\hat{m}_0$ is known in only some special cases (e.g., Lee 1997; Yao 1988).

3. OPTIMIZATION USING GENETIC ALGORITHMS

Because the search space is enormous, optimization of $\mathrm{MDL}(m, \tau_1, \ldots, \tau_m, p_1, \ldots, p_{m+1})$ is a nontrivial task. In this section we propose using a GA to effectively tackle this problem.

3.1 General Description

The basic idea of the canonical form of GAs can be described as follows. An initial set, or population, of possible solutions to an optimization problem is obtained and represented in vector form. These vectors, often called *chromosomes*, are free to "evolve" in the following way. Parent chromosomes are randomly chosen from the initial population, and chromosomes having lower (higher) values of the objective criterion to be minimized (maximized) would have a higher likelihood of being chosen. Then offspring are produced by applying a *crossover* or a *mutation* operation to the chosen parents. Once sufficient numbers of such second-generation offspring are produced, third-generation offspring are produced from these second-generation offspring in a similar fashion. This process continues for a number of generations. If one believes in Darwin's *Theory of Natural Selection*, then the expectation is that objective criterion values of the offspring will gradually improve over generations and approach the optimal value.

In a crossover operation, one child chromosome is produced from "mixing" two parent chromosomes. The aim is to allow the possibility of the child receiving different best parts from its parents. A typical "mixing" strategy is that every child gene location has an equal chance of receiving either the corresponding father gene or the corresponding mother gene. This crossover operation is the distinct feature that makes GAs different from other optimization methods. (For possible variants of the crossover operation, see Davis 1991.)

In a mutation operation, one child chromosome is produced from one parent chromosome. The child is essentially the same as its parent except for a small number of genes in which randomness is introduced to alter the types of genes. Such a mutation operation prevents the algorithm from being trapped in local optima.

To preserve the best chromosome of a current generation, an additional step, called the *elitist* step, may be performed. Here the worst chromosome of the next generation is replaced with the best chromosome of the current generation. Including this elitist step guarantees the monotonicity of the algorithm.

There are many variations of the foregoing canonical GA. For example, parallel implementations can be applied to speed up the convergence rate as well as to reduce the chance of convergence to suboptimal solutions (Forrest 1991; Alba and Troya 1999). In this article we implement the *island model*. Rather than run only one search in one giant population, the island model simultaneously runs NI (number-of-islands) canonical GAs in NI different subpopulations. The key feature is, periodically, a number of individuals are migrated among the islands according to some migration policy. The migration can be implemented in numerous ways (Martin, Lienig, and Cohoon 2000; Alba and Troya 2002). In this article we adopt the following migration policy: After every M_i generations, the worst M_N chromosomes from the jth island are replaced by the best M_N chromosomes from the $(j-1)$st island, $j = 2, \ldots, NI$. For $j = 1$, the best M_N chromosomes are migrated from the NIth island. In our simulations we used $NI = 40$, $M_i = 5$, $M_N = 2$, and a subpopulation size of 40.

3.2 Implementation Details

This section provides details of our implementation of the GAs that is tailored to our piecewise AR model fitting.

Chromosome Representation. The performance of a GA certainly depends on how a possible solution is represented as a chromosome, and for the current problem a chromosome should carry complete information for any $\mathcal{F} \in \mathcal{M}$ about the breakpoints, τ_j, as well as the AR orders, p_j. Once these quantities are specified, maximum likelihood estimates of other model parameters can be uniquely determined. Here we propose using the following chromosome representation: a chromosome $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_n)$ is of length n with gene values δ_t defined as

$$\delta_t = \begin{cases} -1 & \text{if no break point at } t \\ p_j & \text{if } t = \tau_{j-1} \text{ and the AR order} \\ & \quad \text{for the } j\text{th piece is } p_j. \end{cases}$$

Furthermore, the following "minimum span" constraint is imposed on $\boldsymbol{\delta}$: say if the AR order of a certain piece in $\mathcal{F}$ is p, then the length of this piece must have at least m_p observations. This predefined integer m_p is chosen to guarantee that there are sufficient observations to obtain quality estimates for the parameters of the AR(p) process. Also, in the practical implementation of the algorithm, one needs to impose an upper bound P_0 on the order p_j's of the AR processes. There seems to be no universal choice for P_0, because for complicated series one needs a large P_0 to capture for example seasonality, whereas for small series P_0 cannot be larger than the number of observations n. For all of our numerical examples, we set $P_0 = 20$, and the corresponding minimum span m_p's are listed in Table 1.

Our empirical experience suggests that the foregoing representation scheme, together with the minimum span constraint, is extremely effective for the purpose of using GAs to minimize $\mathrm{MDL}(m, \tau_1, \ldots, \tau_m, p_1, \ldots, p_{m+1})$. This is most likely due to the fact that the location information of the breakpoints and the order of the AR processes are explicitly represented.

Initial Population Generation. Our implementation of the GA starts with an initial population of chromosomes generated at random. For this procedure, the user value π_B, the probability that the "jth location" of the chromosome being generated is a breakpoint, is needed. A large value of π_B makes the initial chromosomes have a large number of break points; thus a small value is preferred. We use $\pi_B = \min(m_p)/n = 10/n$. (We present a sensitivity analysis for this parameter in Sec. 4.) Once a location is declared to be a break, an AR order is selected from the uniform distribution with values $0, 1, \ldots, P_0$. The following strategy is used to generate each initial chromosome. First, select a value for p_1 from $\{0, \ldots, P_0\}$ with equal probabilities and set $\delta_1 = p_1$; that is, the first AR piece is of order p_1. Then the next $m_{p_1} - 1$ genes δ_i's (i.e., $\delta_2, \ldots, \delta_{m_{p_1}}$) are set to -1, so that the foregoing minimum span constraint is imposed for this first piece. Now the next gene in line, $\delta_{m_{p_1}+1}$, will either be initialized as a breakpoint (i.e., assigned a nonnegative integer p_2) with probability π_B or be assigned -1 with probability $1 - \pi_B$. If it is to be initialized as a breakpoint, then we set $\delta_{m_{p_1}+1} = p_2$, where p_2 is randomly drawn from $\{0, \ldots, P_0\}$. This implies that the second AR process is of order p_2, and

Table 1. Values of m_p Used in the Simulations

p	0–1	2	3	4	5	6	7–10	11–20
m_p	10	12	14	16	18	20	25	50

the next $m_{p_2} - 1$ δ_i's will be assigned -1 so that the minimum span constraint is guaranteed. But if $\delta_{m_{p_1}+1}$ is to be assigned with -1, then the initialization process will move to the next gene in line and determine whether this gene should be a breakpoint gene or a "-1" gene. This process continues in a similar fashion, and a random chromosome is generated when the process hits the last gene, δ_n.

Crossover and Mutation. Once a set of initial random chromosomes is generated, new chromosomes are generated by either a crossover operation or a mutation operation. In our implementation we set the probability for conducting a crossover operation as $\pi_C = 1 - \min(m_p)/n = (n - 10)/n$.

For the crossover operation, two parent chromosomes are chosen from the current population of chromosomes. These two parents are chosen with probabilities inversely proportional to their ranks sorted by their MDL values. In other words, chromosomes with smaller MDL values will have a higher likelihood of being selected. From these two parents, the gene values, δ_i's, of the child chromosome will be inherited in the following manner. First, for $t = 1$, δ_t takes on the corresponding δ_t value from either the first or the second parent with equal probabilities. If this value is -1, then the same gene-inheriting process is repeated for the next gene in line (i.e., δ_{t+1}). If this value is not -1, then it is a nonnegative integer p_j denoting the AR order of the current piece. In this case the minimum span constraint is imposed (i.e., the next $m_{p_j} - 1$ δ_t's are set to -1), and the same gene-inheriting process is applied to the next available δ_t.

For mutation, one child is reproduced from one parent. Again, this process starts with $t = 1$, and every δ_t (subject to the minimum span constraint) can take one of the following three possible values: (a) with probability π_P, it takes the corresponding δ_t value from the parent; (b) with probability π_N, it takes the value -1; and (c) with probability $1 - \pi_P - \pi_N$, it takes the a new randomly generated AR order p_j. In the examples that follow we set $\pi_P = .3$ and $\pi_N = .3$.

Declaration of Convergence. Recall that we adopt the island model in which migration is allowed for every $M_i = 5$ generations. At the end of each migration, the overall best chromosome (i.e., the chromosome with smallest MDL) is noted. If this best chromosome does not change for 10 consecutive migrations, or if the total number of migrations exceeds 20, then this best chromosome is taken as the solution to this optimization problem.

4. SIMULATION RESULTS

We conducted five sets of simulation experiments to evaluate the practical performances of Auto-PARM. The experimental setups of the first two simulations are from Ombao et al. (2001), who used them to test their Auto-SLEX procedure. In the first simulation, the pieces of the true process follow a dyadic structure; that is, the length of each segment is a integer power of 2. In the second and fourth simulations the true process does not contain any structural breaks, but its time-varying spectrum changes slowly over time. In the third simulation the process contains three pieces, one of which is an autoregressive moving average [ARMA(1, 1)] process and another of which is a moving average [MA(1)] process. In the last simulation the process has two distinctive features: The pieces do not follow a dyadic structure, and the length of one of the pieces is very short.

For the results reported in this section and in Section 5, we obtained slightly better results by minimizing MDL based on the exact likelihood function evaluated at Yule–Walker estimates. That is, we used MDL as defined by (5) in all of the simulation results in this section. Throughout the section we obtained the results reported for Auto-SLEX using computer code provided by Dr. Hernando Ombao.

4.1 Piecewise Stationary Process With Dyadic Structure

In this simulation example, the target nonstationary series is generated with the model

$$Y_t = \begin{cases} .9Y_{t-1} + \varepsilon_t & \text{if } 1 \le t \le 512 \\ 1.69Y_{t-1} - .81Y_{t-2} + \varepsilon_t & \text{if } 513 \le t \le 768 \\ 1.32Y_{t-1} - .81Y_{t-2} + \varepsilon_t & \text{if } 769 \le t \le 1{,}024, \end{cases} \tag{6}$$

where $\varepsilon_t \sim \text{iid}\, \mathrm{N}(0, 1)$. The main feature of this model is that the lengths of the pieces are a power of 2. This in fact is ideally suited for the Auto-SLEX procedure of Ombao et al. (2001). A typical realization of this process is shown in Figure 1. For $\omega \in [0, .5)$, let $f_j(\omega)$ be the spectrum of the jth piece, that is,

$$f_j(\omega) = \sigma_j^2 \left| 1 - \phi_{j1} \exp\{-i2\pi\omega\} - \cdots - \phi_{jp_j} \exp\{-i2\pi p_j \omega\} \right|^{-2}. \tag{7}$$

Then for $t \in [\tau_{j-1}, \tau_j)$, the *time-varying spectrum* of the process Y_t in (1) is $f(t/n, \omega) = f_j(\omega)$. The true spectrum of the process in (6) is shown in the middle part of Figure 2, where darker shades represent higher power.

We applied Auto-PARM to the realization in Figure 1 and obtained two breakpoints located at $\hat{\tau}_1 = 512$ and $\hat{\tau}_2 = 769$, indicated by the dotted vertical lines in the figure. Auto-PARM correctly identified the AR orders ($\hat{p}_1 = 1$, $\hat{p}_2 = 2$, and $\hat{p}_3 = 2$) for this realization. From this segmentation, the time-varying spectrum of this realization was estimated as $\hat{f}_{t/n}(\omega) = \hat{f}_j(\omega)$, where $\hat{f}_j(\omega)$ is obtained by replacing parameters in (7) with their corresponding estimates. The estimated time-varying spectrum is displayed in Figure 2(a). Our implementation of

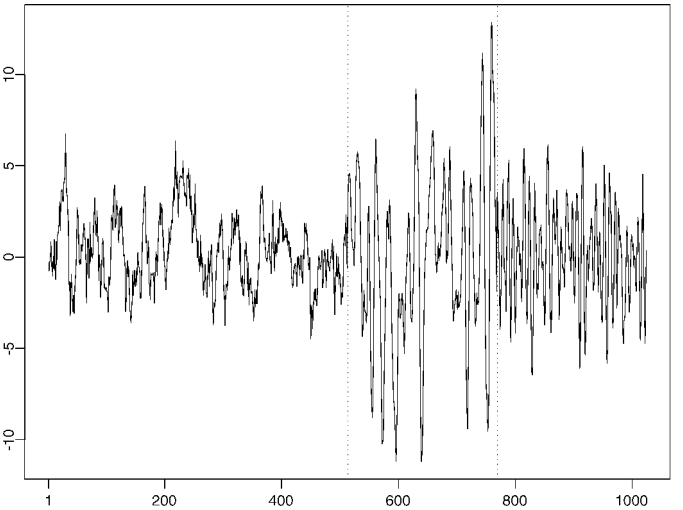

Figure 1. A Realization From the Piecewise Stationary Process in (6).

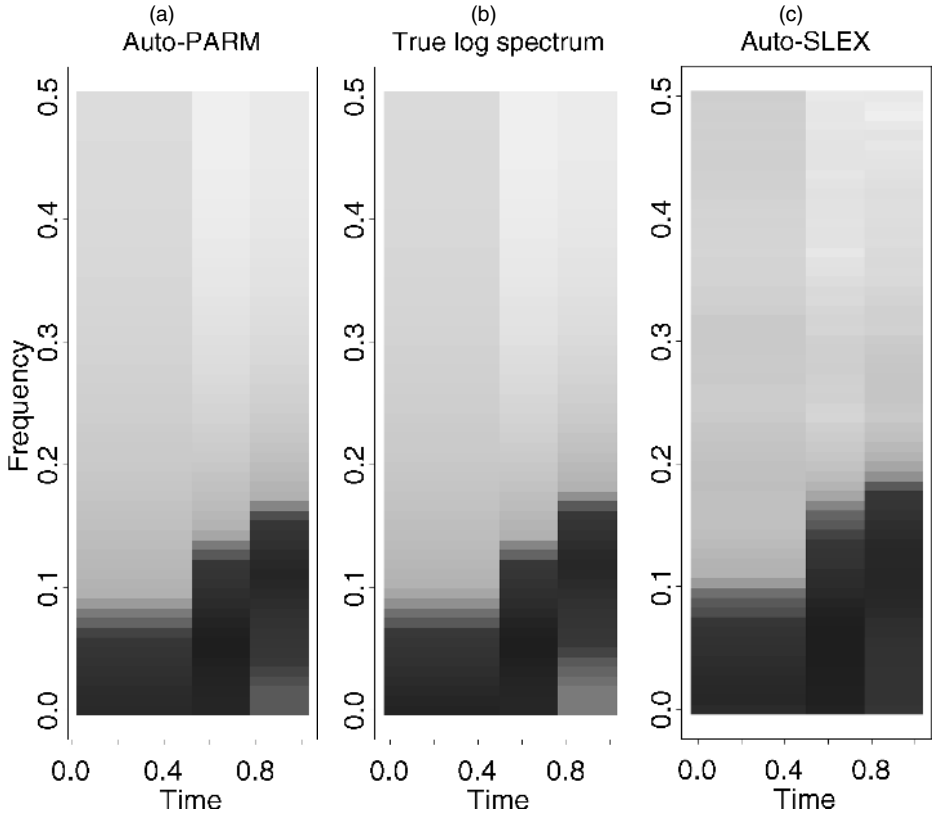

Figure 2. True Time-Varying Log-Spectrum of the Process in (6) (b) and Auto-PARM (a) and Auto-SLEX (c) Estimates From the Realization of Figure 1.

Auto-PARM, written in Compaq Visual Fortran, took 2.34 seconds on a 1.6-GHz Intel Pentium M processor. The Auto-SLEX time-varying spectrum of this realization is shown in Figure 2(c).

Next, we simulated, 200 realizations of the process in (6), and applied Auto-PARM to segment each of these realizations. Table 2 lists the percentages of the fitted number of segments. For comparative purposes, the table also gives the corresponding values of the Auto-SLEX method. Notice that Auto-PARM gave the correct number of segments for 96% of the 200 realizations, whereas Auto-SLEX gave the correct segmentation for 73% of the realizations. Table 2 also reports, for each $\hat{m}$, the mean and standard deviation of $\hat{\lambda}_j := (\hat{\tau} - 1)/n$, $j = 1, \ldots, \hat{m} - 1$, where $\hat{\tau}_j$ is the Auto-PARM estimate of τ_j. For convenience we refer to $\hat{\lambda}_j$ as the *relative* breakpoint.

Table 2. Summary of the Estimated Breakpoints From Both the Auto-SLEX and Auto-PARM Procedures for the Process (6)

Number of segments	Auto-SLEX		Auto-PARM			
	Breakpoints		Breakpoints			
	(%)	ASE	(%)	Mean	SE	ASE
2	2.5	.396 (.019)	0			
3	73.0	.121 (.027)	96.0	.500 .750	.007 .005	.049 (.030)
4	11.0	.146 (.040)	4.0	.496 .566 .752	.004 .108 .003	.140 (.036)
5	9.5	.206 (.045)	0			
≥6	4.0	.253 (.103)	0			
All	100.0	.144 (.064)	100.0			.052 (.035)

NOTE: For Auto-PARM, the means and standard errors of the relative breakpoints are also reported.

Table 3 lists the relative frequencies of the AR order p estimated by the Auto-PARM procedure for the 96% of the realizations with three pieces. Of the 200 realizations, 44% have 2 breaks and AR orders 1, 2, and 2. For these realizations, the means and the standard errors of the estimated parameters $\phi_1, \ldots, \phi_{p_j}, \sigma_j^2$ are given in Table 4. From these tables, we can see that Auto-PARM applied to the foregoing piecewise stationary process performs extremely well, especially for locating the breakpoints.

4.1.1 Sensitivity Analysis. We also considered the sensitivity of the GA to the probabilities of initialization (π_B) and crossover (π_C). To assess the sensitivity we applied Auto-PARM to the same realizations used in Table 2 for each combination of values of $\pi_B \in \{.01, .1\}$ and $\pi_C = \{.90, .99\}$. The other parameter values in the implementation of Auto-PARM are as described in Section 3.

Table 3. Relative Frequencies of the AR Order Estimated by the Auto-PARM Procedure for the Realizations of Model (6)

Order	0	1	2	3	4	5	6	≥7
p_1	0	99.0	1.0	0	0	0		
p_2	0	0	67.7	16.7	9.9	3.6	.5	1.5
p_3	0	0	60.4	22.9	5.7	6.8	2.1	2.1

Table 4. Summary of Parameter Estimates Obtained by Auto-PARM for the Realizations That Have Two Breaks and Pieces With Orders 1, 2, and 2

			Parameter		
Segment	*Model*		ϕ_1	ϕ_2	σ^2
I	AR(1)	True	.90		1.00
		Mean	.89		1.02
		SE	(.02)		(.07)
II	AR(2)	True	1.69	−.81	1.00
		Mean	1.65	−.78	1.12
		SE	(.05)	(.05)	(.19)
III	AR(2)	True	1.32	−.81	1.00
		Mean	1.30	−.79	1.07
		SE	(.04)	(.04)	(.13)

NOTE: For each segment, the true parameters, the mean, and the standard errors (in parentheses) are given.

The relative frequency of the number of breakpoints estimated by Auto-PARM is shown in Table 5 (columns 4 and 5). For the replicates with three pieces, the means of the breakpoints and standard errors are given in columns 6 and 7. The frequencies of the correct AR order estimated by Auto-PARM for each piece are given in columns 8, 9, and 10. The averages of the MDL values and the standard error are given in the last two columns. The column labeled "time" gives the average time in seconds to implement Auto-PARM.

From Table 5, we see that distinct values of π_B and π_C give comparable values of MDL. Notice that Auto-PARM runs the fastest for the values selected for π_B and π_C in Section 3, that is, $\pi_B = \min(m_p)/n$ and $\pi_C = 1 - \min(m_p)/n$. As this table shows, there is little impact on the choice of initial values for π_B and π_C in executing Auto-PARM.

4.2 Slowly Varying AR(2) Process

The true model considered in this second simulation experiment does not have a structural break. Rather, the process has a slowly changing spectrum given by the following time-dependent AR(2) model:

$$Y_t = a_t Y_{t-1} - .81 Y_{t-2} + \varepsilon_t, \qquad t = 1, 2, \ldots, 1{,}024, \qquad (8)$$

where $a_t = .8\{1 - .5\cos(\pi t/1{,}024)\}$ and $\varepsilon_t \sim \text{iid}\,\text{N}(0, 1)$. A typical realization of this process is shown in Figure 3, whereas the spectrum of this process is shown in Figure 5(b).

For the realization in Figure 3, the Auto-PARM procedure segmented the process into three pieces with breakpoints located at $\hat{\tau}_1 = 318$ and $\hat{\tau}_2 = 614$ (the vertical dotted lines in this figure). In addition, Auto-PARM modeled each of the three pieces as an AR(2) process. The run time for this fitting was 1.79 seconds. Based on the model found by Auto-PARM, the time-varying spectrum of this realization was computed and is shown in Figure 4(a). The Auto-SLEX time-varying spectrum of this realization is shown in Figure 4(b).

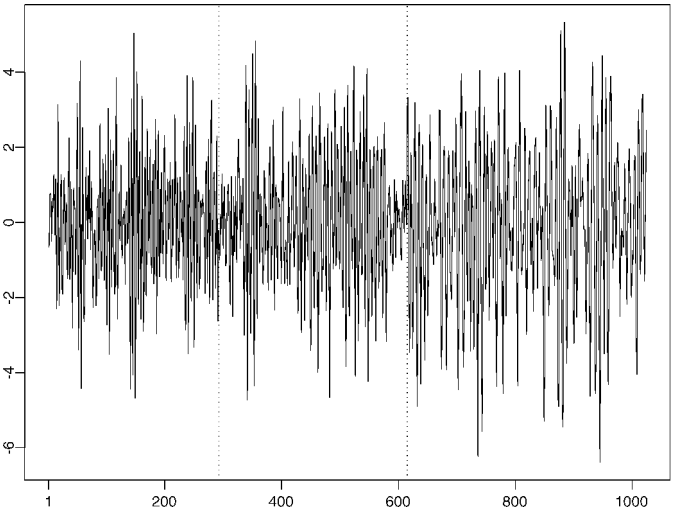

Figure 3. Realization From the Process in (8).

Next, we generated 200 realizations of the foregoing process, and obtained the corresponding Auto-PARM estimates. Because there are no true structural breaks in such realizations, we follow Ombao et al. (2001) and use the average squared error (ASE) as a numerical error measure of performance. The ASE is defined by

$$\text{ASE} = \{n(M_J/2 + 1)\}^{-1} \times \sum_{t=1}^{n} \sum_{k=0}^{M_J/2} \left\{\log \hat{f}(t/n, \omega_k) - \log f(t/n, \omega_k)\right\}^2,$$

where $\hat{f}(\cdot, \cdot)$ is an estimate of the true time-dependent spectrum $f(\cdot, \cdot)$ of the process, J is a prespecified scale satisfying $J < L = \log_2(n)$, and $M_J := n/2^J$ [see eq. (19) in Ombao et al. 2001]. In this simulation we took $J = 4$.

The number of segments, locations of the breakpoints, and the ASEs of the Auto-PARM estimates are summarized in Table 6. Also listed in Table 6 are the ASE values of the Auto-SLEX procedure. From this table, two main observations can be made. First, for each of the simulated processes,

Table 5. Sensitivity Analysis (NI × popsize = 40 × 40): Summary of Sensitivity Analysis of π_B and π_C of Auto-PARM Based on 200 Realizations of (6)

			Number of breaks (%)		Auto-PARM breakpoints		AR order			
							$\hat{p}_1$	$\hat{p}_2$	$\hat{p}_3$	
π_B	π_C	*Time*	*2*	*3*	*Mean*	*SE*	*1*	*2*	*2*	*MDL*
.01	.90	14.97	91.5	8.5	.500	.008	99.5	57.4	60.1	1,520.45
					.749	.007				
.01	.99	3.0	95.5	4.5	.499	.009	99.5	56.5	60.2	1,520.56
					.750	.007				
.10	.90	16.85	95.5	4.5	.499	.010	98.4	53.4	53.4	1,519.41
					.750	.008				
.10	.99	4.9	94.5	5.5	.499	.008	97.9	57.1	57.1	1,519.22
					.750	.007				

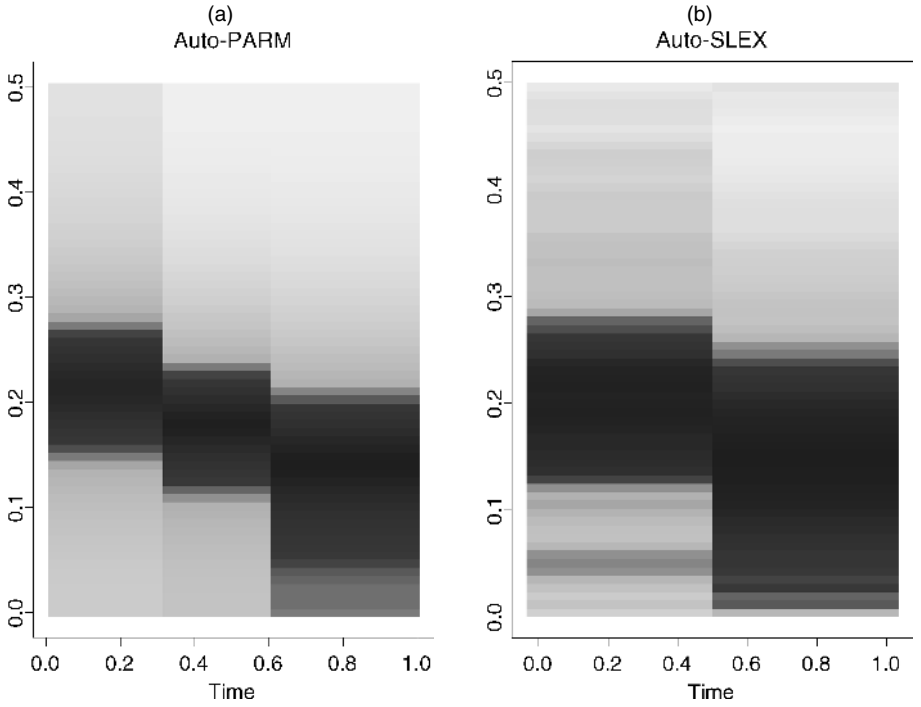

Figure 4. Auto-PARM and Auto-SLEX Estimates of Log-Spectrum of Process in (8) for the Realization From Figure 3.

Auto-PARM produces either two or three segments that are of roughly the same length, whereas the Auto-SLEX procedure tends to split the process into a larger number of segments. Second, the ASE values from Auto-PARM are smaller than those from Auto-SLEX.

To show a "consistency"-like property of Auto-PARM, we computed the average of all of the time-varying spectra of the 200 Auto-PARM and Auto-SLEX estimates. The averaged Auto-PARM spectrum is displayed in Figure 5(a) and looks remarkably similar to the true time-varying spectrum. The averaged Auto-SLEX spectrum is shown in Figure 5(c). Finally, Table 7 summarizes the Auto-PARM estimates of the AR orders for the foregoing process. Notice that most of the segments were modeled as AR(2) processes.

Table 6. Breakpoints and ASE Values From the Auto-PARM and the Auto-SLEX Estimates Computed From 200 Realizations of (8)

Number of segments	*Auto-SLEX*		*Auto-PARM breakpoints*			
	(%)	*ASE*	*(%)*	*Mean*	*SE*	*ASE*
1	0	—	0	—	—	—
2	40.5	.191	37.5	.496	.055	.129
		(.019)				(.015)
3	37.0	.171	62.0	.365	.074	.081
		(.022)		.662	.079	(.016)
4	15.0	.174	.5	.308	—	.10
		(.029)		.538	—	—
				.875	—	
5	5.0	.202				
		(.045)				
≥6	2.5	.223				
		(.037)				
All	100.0	.182	100.0			.099
		(.027)				(.028)

NOTE: Numbers inside parentheses are standard errors of the ASE values.

4.3 Piecewise ARMA Process

Recall that the Auto-PARM procedure assumes that the observed process is composed of a series of stationary AR processes. This third simulation, designed to assess the performance of Auto-PARM when the AR assumption is violated, has a data-generating model given by

$$Y_t = \begin{cases} -.9Y_{t-1} + \varepsilon_t + .7\varepsilon_{t-1} & \text{if } 1 \le t \le 512 \\ .9Y_{t-1} + \varepsilon_t & \text{if } 513 \le t \le 768 \\ \varepsilon_t - .7\varepsilon_{t-1} & \text{if } 769 \le t \le 1{,}024, \end{cases} \tag{9}$$

where $\varepsilon_t \sim \text{iid}\, N(0, 1)$. Notice that the first piece is an ARMA(1, 1) process, whereas the last piece is a MA(1) process. A typical realization of this process is shown in Figure 6.

We applied the Auto-PARM procedure to the realization in Figure 6 and obtained three pieces. The breakpoints are at $\hat{\tau}_1 = 513$ and $\hat{\tau}_2 = 769$ (the dotted vertical lines in the figure), whereas the orders of the AR processes are 4, 1, and 2. The total run time for this fit was 1.53 seconds. The time-varying spectrum (not shown here) based on the model found by Auto-PARM is reasonably close to the true spectrum (not shown here), even though two of the segments are not AR processes.

To assess the large-sample behavior of Auto-PARM, we generated 200 realizations from (9), and obtained the corresponding Auto-PARM estimates. An encouraging result is that for all 200 realizations, Auto-PARM always gave the correct number of stationary segments. The estimates of the breakpoint locations are summarized in Table 9. Table 10 gives the relative frequency of the AR order p_j selected to model the pieces of the realizations. As expected, quite often large AR orders were selected for the ARMA and MA segments.

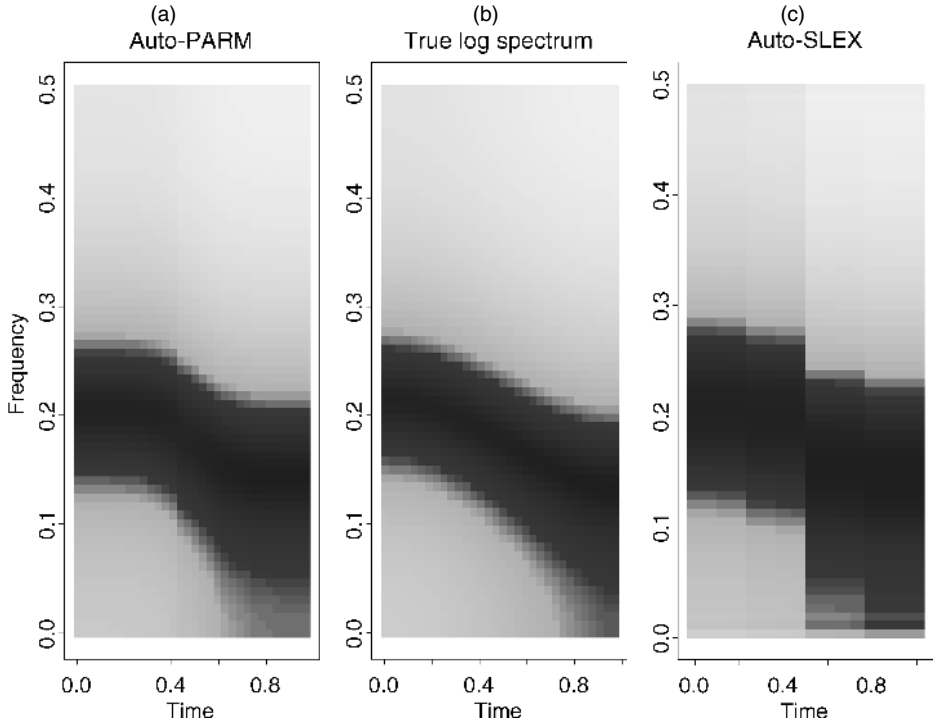

Figure 5. True Time-Varying Log-Spectrum of the Process in (8) (b) and Auto-PARM (a) and Auto-SLEX (c) Log-Spectrum Estimate. (Average of log-spectrum estimate obtained from 200 realizations.)

Table 7. Relative Frequencies of the AR Order Selected by Auto-PARM for the Realizations From the Process (8)

Order	*0*	*1*	*2*	*3*	*4*	*≥5*
Two-segment realizations						
p_1	0	0	97.3	1.3	1.3	0
p_2	0	0	93.3	5.3	1.3	0
Three-segment realizations						
p_1	0	0	100.0	0	0	0
p_2	0	0	94.4	4.8	.8	0
p_3	0	0	91.1	8.1	.8	0

Table 8. Summary of Parameter Estimates of Slowly Varying AR(2) Process Realizations Segmented by Auto-PARM as Two and Three Pieces, Where Each Piece Is an AR(2) Process

		Parameter		
jth piece		ϕ_1	ϕ_2	σ^2
Two-piece realizations with AR(2) pieces: 68				
1	True		−.81	1.00
	Mean	.54	−.79	1.05
	SE	(.04)	(.03)	(.07)
2	True		−.81	1.00
	Mean	1.05	−.79	1.05
	SE	(.04)	(.03)	(.07)
Two-piece realizations with AR(2) pieces: 106				
1	True		−.81	1.00
	Mean	.46	−.80	1.03
	SE	(.06)	(.03)	(.08)
2	True		−.81	1.00
	Mean	.82	−.81	1.01
	SE	(.08)	(.04)	(.10)
3	True		−.81	1.00
	Mean	1.14	−.80	1.06
	SE	(.05)	(.04)	.10

NOTE: For each segment, the true parameters, their mean, and standard deviation (in parentheses) are shown.

4.4 Time-Varying MA(2) Process

Like the example in Section 4.2, the true model considered in this last simulation experiment does not have a structural break. Rather, the process has a changing spectrum given by the following time-dependent MA(2) model:

$$Y_t = \varepsilon_t + a_t\varepsilon_{t-1} + .5\varepsilon_{t-2}, \qquad t = 1, 2, \ldots, 1{,}024, \qquad (10)$$

where $a_t = 1.122\{1 - 1.781\sin(\pi t/2{,}048)\}$ and $\varepsilon_t \sim \text{iid}\,\text{N}(0, 1)$. A typical realization of this process is shown in Figure 7, whereas the spectrum of this process is shown in Figure 9(b).

For the realization in Figure 7, Auto-PARM produced four segments with AR orders 5, 3, 5, and 3, and breakpoints located

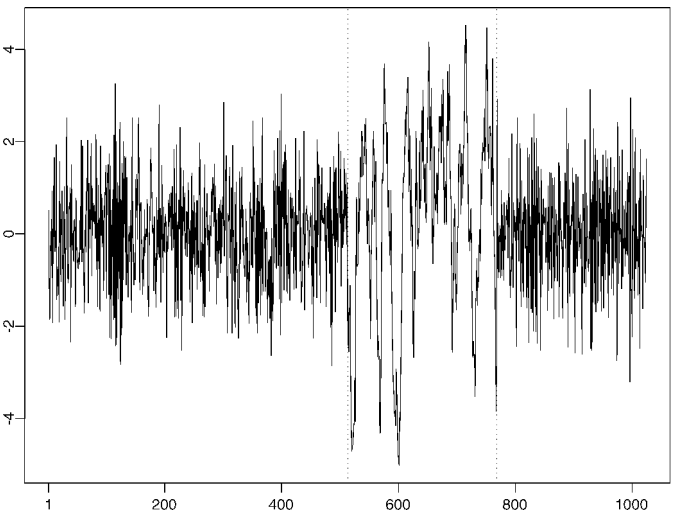

Figure 6. A realization From the Piecewise Stationary Process in (9).

Table 9. Summary of Auto-PARM Estimated Breakpoints Obtained From 200 Realizations From the Process in (9)

Number of segments	%	Relative break points	
		Mean	SE
3	100.0	.50	.005
		.75	.003

Table 10. Relative Frequencies of the AR Order Selected by Auto-PARM for the Realizations From the Process (9)

Order	0	1	2	3	4	5	6	7	≥8
p_1	0	4.0	22.5	40.0	23.5	8.5	1.0	.5	0
p_2	0	89.5	8.5	1.5	.5	0	0	0	0
p_3	0	.5	22.0	45.0	19.5	7.5	4.5	1.0	0

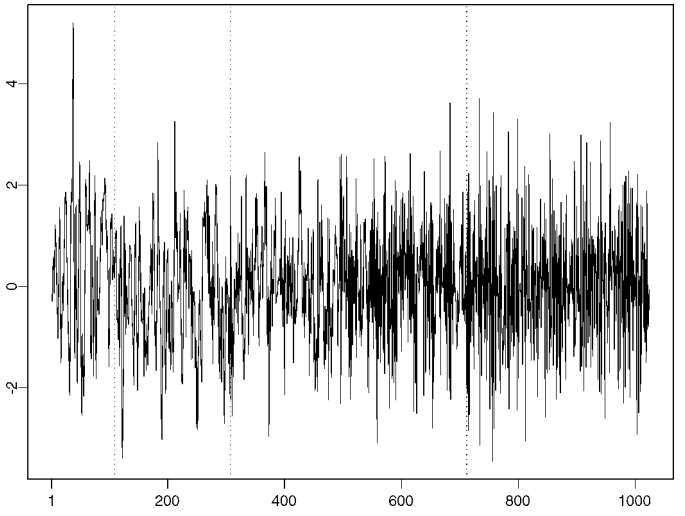

Figure 7. Realization From the Process in (10).

at $\hat{\tau}_1 = 109$, $\hat{\tau}_2 = 307$, and $\hat{\tau}_3 = 712$ (the vertical dotted lines in this figure). The run time for this model fit was 3.76 seconds. Based on the model found by Auto-PARM, the time-varying spectrum of this realization is shown in Figure 8(a). For comparison, the Auto-SLEX time-varying spectrum estimate of this realization is shown in Figure 8(b).

Next we generated 200 realizations of the above process, and the corresponding Auto-PARM estimates. The number of segments, locations of the breakpoints, and the ASEs of Auto-PARM estimates are summarized in Table 11.

From this table, we observe that for most of the realizations, Auto-PARM produces three segments. We computed the average of all of the time-varying spectra of the 200 Auto-PARM estimates; the averaged spectrum is displayed in Figure 9(a) and the average of the 200 Auto-SLEX estimates of the time-varying spectra is shown in Figure 9(c).

The true spectrum in Figure 9 is well estimated by Auto-PARM and Auto-SLEX. Remarkably, Auto-PARM estimates the true spectrum well, despite the fact that it splits the realizations into fewer pieces than Auto-SLEX does.

Table 12 summarizes the Auto-PARM estimates of the AR orders for the foregoing process for those realizations with three pieces. In general, the segments were modeled as AR processes of high order.

4.5 Short Segments

To complement the foregoing simulation experiments, in this section we assess the performance of Auto-PARM with the following process containing a short segment:

$$Y_t = \begin{cases} .75Y_{t-1} + \varepsilon_t & \text{if } 1 \le t \le 50 \\ -.50Y_{t-1} + \varepsilon_t & \text{if } 51 \le t \le 1{,}024, \end{cases} \tag{11}$$

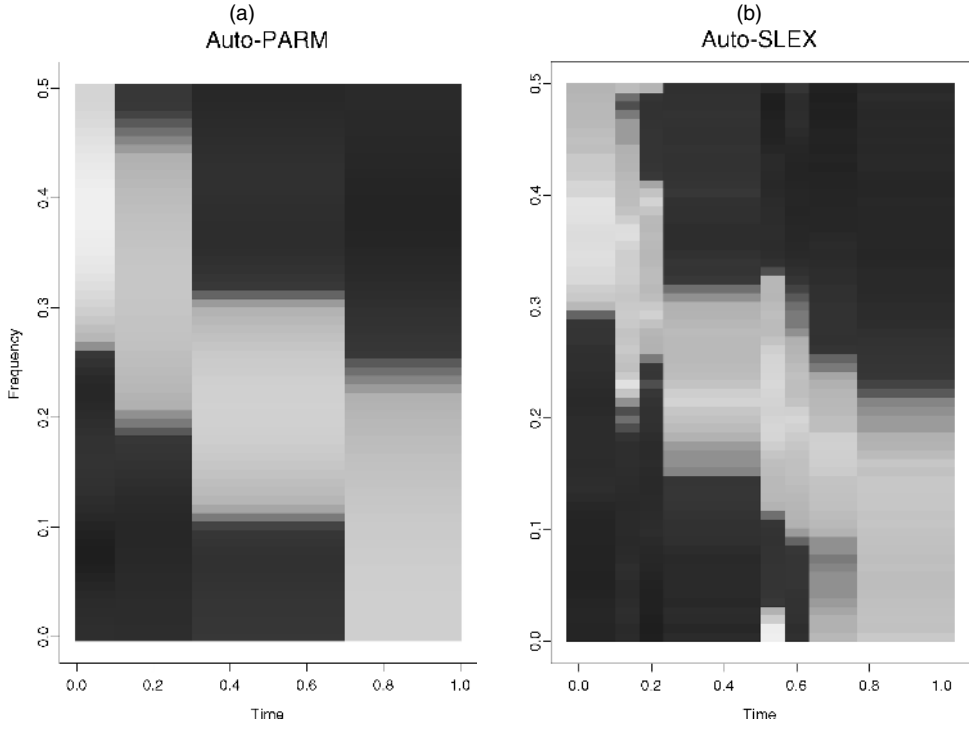

Figure 8. Auto-PARM (a) and Auto-SLEX (b) Estimates of Log-Spectrum of Process in (10) for the Realization From Figure 7.

Table 11. Summary of the Estimated Breakpoints From the Auto-SLEX and Auto-PARM Procedures for the Process (10)

Number of segments	Auto-SLEX		Auto-PARM			
	Breakpoints (%)	ASE	Breakpoints (%)	Mean	SE	ASE
2			3.0	.374	.040	.307 (.023)
3	3.5	.187 (.027)	89.0	.238 .548	.072 .089	.211 (.029)
4	6.5	.157 (.017)	8.0	.156 .391 .667	.045 .062 .093	.182 (.021)
5	15.5	.170 (.028)				
6	17.0	.163 (.025)				
7	20.0	.158 (.030)				
8	15.0	.180 (.029)				
9	11.5	.203 (.032)				
≥10	11.0	.223 (.035)				
All	100.0	.18 (.036)				.211 (.034)

NOTE: For Auto-PARM, the means and standard errors of the relative breakpoints are also reported. Numbers inside parentheses are standard errors of the ASE values.

where $\varepsilon_t \sim \text{iid}\, N(0, 1)$. A typical realization of this process is shown in Figure 10. For the realization in Figure 10, Auto-PARM gives a single breakpoint at $\hat{\tau}_1 = 51$, which is shown as the vertical dotted line in Figure 10. Both pieces are modeled as AR(1) processes. The run time for this realization was 2.70 seconds.

Table 12. Relative Frequencies of the AR Order Selected by Auto-PARM for the Realizations (with three segments) From the Process (10)

Order	1	2	3	4	5
p_1	10.0	40.0	20.0	20.0	
p_2		40.0	20.0	30.0	
p_3		10.0	10.0	70.0	10.0

We further applied the Auto-PARM procedure to 200 realizations of this process. For all of these realizations, Auto-PARM found one breakpoint. The mean of the relative position estimates of this changepoint is .042 (the true value is .049), with a standard error of .004. The minimum, median, and maximum of the breakpoints are 34, 51, and 70. Table 13 gives the relative frequency of the orders p_1 and p_2 of each of the two pieces selected by Auto-PARM. The Auto-PARM procedure correctly segmented 92.5% of the realizations (two AR pieces of order 1). This is exceptional performance for a process in which the break occurs near the beginning of the series.

4.6 Further Remarks on Estimated Breaks

As seen in the simulations in Sections 4.1 and 4.5, when the true unknown pieces are indeed AR processes, Auto-PARM can detect changes in order and in parameters. Consider, for example, the process in Section 4.1 where the first piece is an AR process of order 1 and the second piece is an AR process of order 2. In this case Auto-PARM detected the change of order reasonably well (see Table 3). But the second and third pieces of this process have the same order 2 with different parameter values. Moreover, the two pieces of the process in Section 4.5 have also the same order 1. Tables 3 and 13 show that Auto-PARM

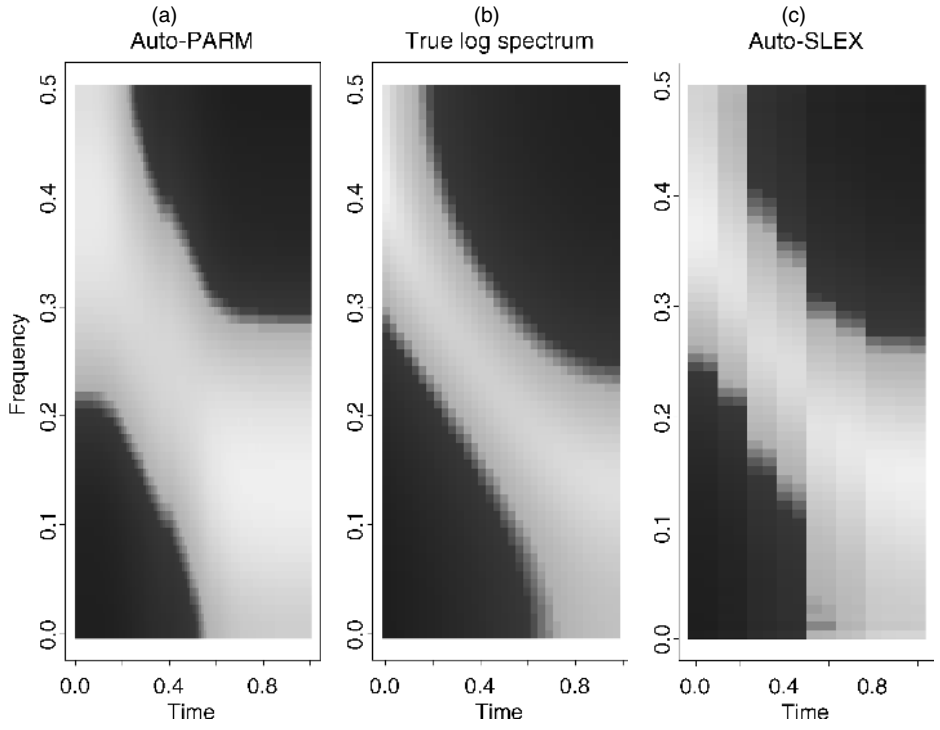

Figure 9. True Time-Varying Log-Spectrum of the Process in (10) (b) and Auto-PARM (a) and Auto-SLEX (c) Log-Spectrum Estimates. (Average of log-spectrum estimates obtained from 200 realizations.)

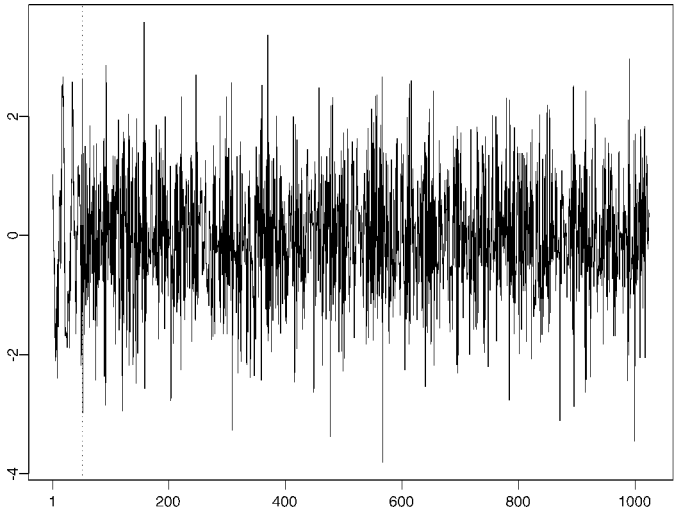

Figure 10. A Realization From the Piecewise Stationary Process in (11).

does a good job in detecting changes in parameter values. The parameter estimates of both processes, given in Tables 4 and 8, show how well Auto-PARM also performs for parameter estimation.

The simulation in Section 4.3 is an example of a process that is not a piecewise AR process. In this case the first piece is an ARMA(1, 1) process, and the third piece is a MA(1) process. Auto-PARM approximates both the ARMA and MA pieces with AR processes perhaps of a large order. The fact that it did exceptionally well in detecting the breaks of this process (see Table 9) is not surprising, because for general stationary process, its spectral density can be well approximated by the spectrum of an AR process under the assumption of continuity of the spectral density (see, e.g., Brockwell and Davis 1991, thm. 4.4.3). The Auto-PARM procedure can then be interpreted as a method for segmenting piecewise stationary processes. In this example, the breaks that Auto-PARM found are points where the spectrum has "large" changes.

5. APPLICATIONS

5.1 Seat Belt Legislation

In the hope of reducing the mean number of monthly deaths and serious injuries, seat-belt legislation was introduced in the United Kingdom in February 1983. Displayed in Figure 11(a) is a time series $\{y_t\}_{t=1}^{120}$, beginning in January 1975, showing the monthly number of deaths and serious injuries. To remove the seasonal component of $\{y_t\}$, Brockwell and Davis (2002) considered the differenced time series $x_t = y_t - y_{t-12}$, and analyzed $\{x_t\}$ with a regression model with errors following an ARMA model. The Auto-PARM procedure applied to the differenced series $\{x_t\}$ segmented the series into three pieces with breakpoints at $\hat{\tau}_1 = 86$ and $\hat{\tau}_2 = 98$. The first two pieces are iid, and the last piece is an AR process of order 1. Figure 11(b) shows the differenced time series $\{x_t\}$, along with the estimated means of each piece. From the Auto-PARM fit, one can conclude that there is a structural change in the time series $\{y_t\}$ after February 1983, which coincides with the time of introduction of the seat belt legislation.

Table 13. Relative Frequencies of the AR Order Selected by Auto-PARM for the Realizations From the Process (11)

Order	0	1	2	3	≥4
p_1	0	96.5	3.0	.5	0
p_2	0	96.0	4.0	0	0

Table 14. Summary of Parameter Estimates of the Realizations of the Process in (11) Segmented Correctly by Auto-PARM (92.5%) as Two Pieces, Where Each Piece Is an AR(1) Process

	First piece		Second piece	
Parameter	ϕ_1	σ^2	ϕ_1	σ^2
True	.75	1.00	−.50	1.00
Mean	.66	1.05	−.50	1.00
SE	(.11)	(.23)	(.03)	(.04)

NOTE: For each segment, the true parameters, their mean, and standard deviation (in parenthesis) are shown.

5.2 Speech Signal

We applied the Auto-PARM procedure to analyze a human speech signal that is the recording of the word "greasy." This signal contains 5,762 observations and is shown in Figure 12(a). This nonstationary time series was also analyzed by the Auto-SLEX procedure of Ombao et al. (2001). The Auto-PARM fit of this speech signal resulted in 15 segments. The total run time was 18.02 seconds. The time-varying log spectrum obtained with this fit is shown in Figure 12(b). This figure shows that the signal is roughly divided into segments corresponding to "G," "R," "EA," "S," and "Y." The information conveyed in this figure closely matches that provided by Ombao et al. (2001). The spectrum from those pieces that correspond to "G" have high power at the lowest frequencies. The pieces that correspond to "R" show power at frequencies slightly above that for "G." The pieces that correspond to "EA" show the evolution of power from lower to higher frequencies. The pieces that correspond to "S" have high power at high frequencies. Notice that the Auto-PARM procedure breaks this speech signal into a smaller number of pieces than the Auto-SLEX procedure while still capturing the important features in the spectrum.

6. MULTIVARIATE TIME SERIES

In this section we demonstrate how Auto-PARM can be extended to model multivariate time series. In Section 6.1 the MDL of a piecewise multivariate AR process is obtained, and in Section 6.2 Auto-PARM is exemplified to a bivariate time series.

6.1 Minimum Description Length

Let $\{\mathbf{Y}_t\}$ be a multivariate time series with r components, and assume that there are breakpoints $\tau_0 := 1 < \tau_1 < \cdots < \tau_m < n+1$ for which the jth piece $\mathbf{Y}_t = \mathbf{X}_{t,j}$, $\tau_{j-1} \le t < \tau_j$, $j = 1, 2, \ldots, m+1$, is modeled by a multivariate AR(p_j) process,

$$\mathbf{X}_{t,j} = \boldsymbol{\gamma}_j + \Phi_{j1}\mathbf{X}_{t-1,j} + \cdots + \Phi_{j,p_j}\mathbf{X}_{t-p_j,j} + \boldsymbol{\Sigma}_j^{1/2}\mathbf{Z}_t, \quad \tau_{j-1} \le t < \tau_j, \quad (12)$$

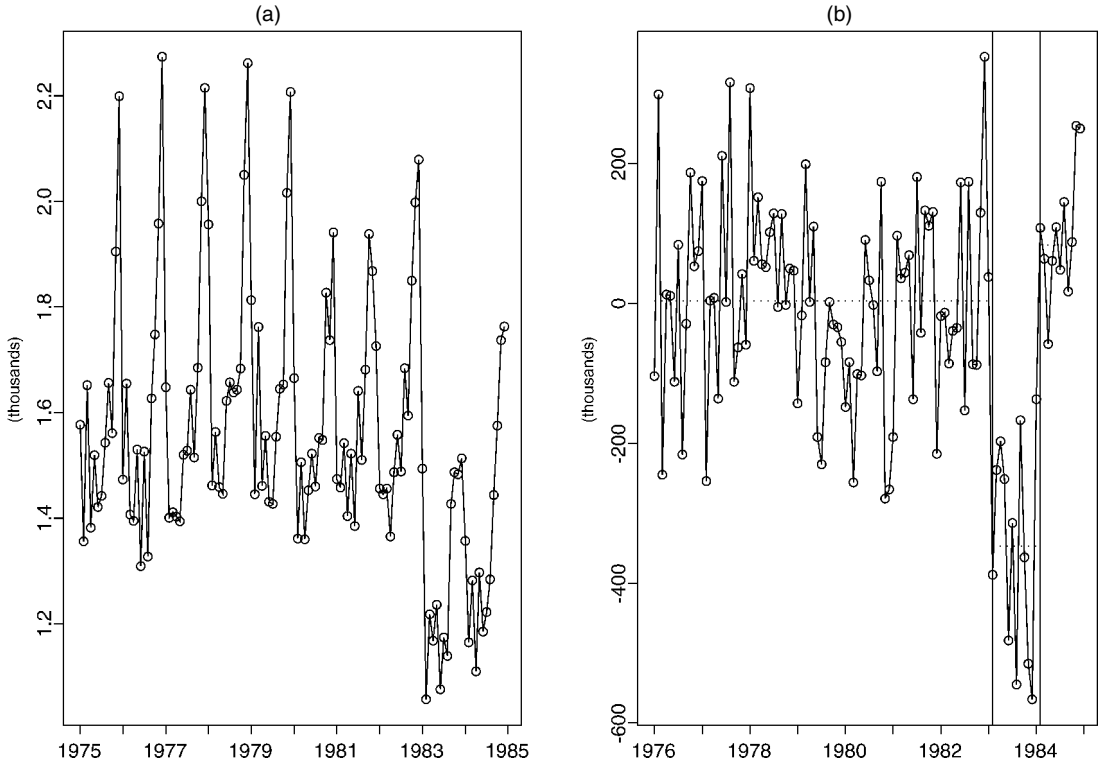

Figure 11. (a) Monthly Deaths and Serious Injuries on U.K. Roads and (b) Transformed Seat Belt Legislation Time Series. The vertical lines are $\hat{\tau}_1$ and $\hat{\tau}_2$. The dotted horizontal line is the estimated mean of the ith segment.

where the noise sequence $\{\mathbf{Z}_t\}$ is iid with mean $\mathbf{0}$ and covariance matrix $\mathbf{I}$. The (unknown) AR matrix coefficients and covariance matrices are of dimension $r \times r$. Let $\mathcal{M}$ be the class of all piecewise multivariate AR models as described above. Let $\mathbf{y}_1, \ldots, \mathbf{y}_n$ be a realization of $\{\mathbf{Y}_t\}$. Parameter estimates in model (12) can be obtained using Whittle's algorithm (see Brockwell and Davis

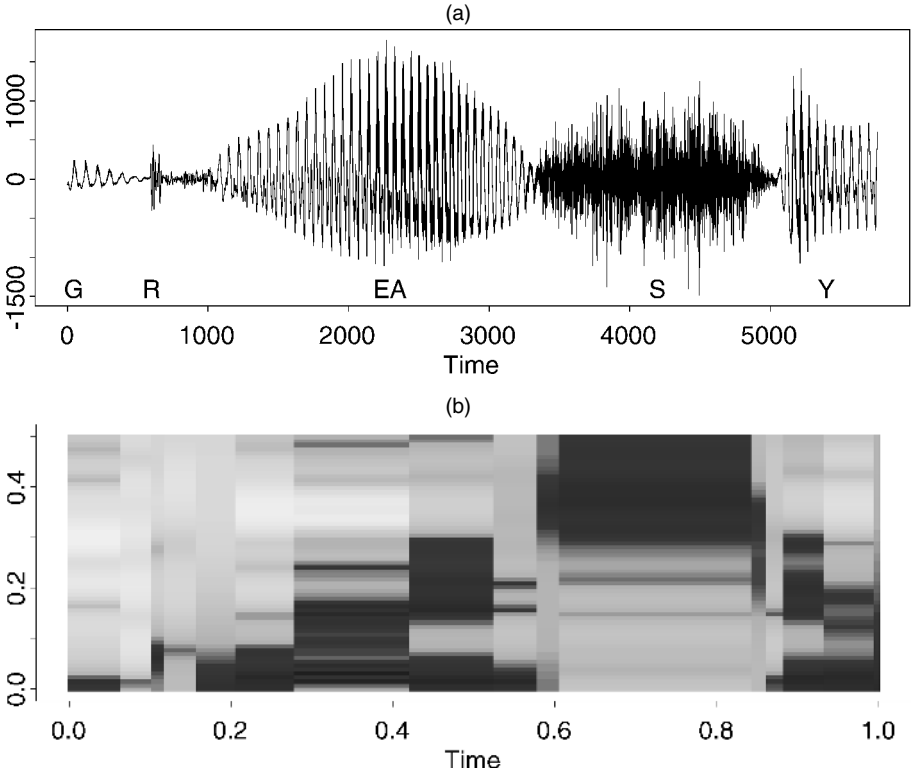

Figure 12. Speech Signal (a) and GA Estimate of the Time-Varying Log Spectrum (b).

1991). From (5), we have

$$\begin{aligned}\mathrm{MDL}(m, \tau_1, \ldots, \tau_m, p_1, \ldots, p_{m+1}) &= \log m + (m+1)\log n \\ &\quad + \sum_{j=1}^{m+1} \log p_j + \sum_{j=1}^{m+1} \frac{3r + 2p_j r^2 + r^2}{4} \log n_j \\ &\quad - \sum_{j=1}^{m+1} \log L(\hat{\Phi}_{j,1}, \ldots, \hat{\Phi}_{j,p_j}, \hat{\boldsymbol{\Sigma}}_j),\end{aligned}$$

where $L(\hat{\Phi}_{j,1}, \ldots, \hat{\Phi}_{j,p_j}, \hat{\boldsymbol{\Sigma}}_j)$ is the likelihood of the jth piece evaluated at the parameter estimates. As in the univariate case, the best segmentation of the realization $\mathbf{y}_1, \ldots, \mathbf{y}_n$ of $\{\mathbf{Y}_t\}$ is defined as the minimizer of $\mathrm{MDL}(m, \tau_1, \ldots, \tau_m, p_1, \ldots, p_{m+1})$. A similar GA can be developed for the practical minimization of $\mathrm{MDL}(m, \tau_1, \ldots, \tau_m, p_1, \ldots, p_{m+1})$.

6.2 Electroencephalogram Analysis

Figure 13 displays two electroencephalograms (EEGs) each of length $n = 32{,}768$ recorded from a female patient who was diagnosed with left temporal lobe epilepsy. This dataset is courtesy of Dr. Beth Malow (formerly from the Department of Neurology at the University of Michigan). Panel (a) shows the EEG from the left temporal lobe (T3 channel), and (b) shows the EEG from the left parietal lobe (P3 channel). Each EEG was recorded for a total of 5 minutes and 28 seconds, with a sampling rate of 100 Hz. Of primary interest is the estimation of the power spectra of both EEGs and the coherence between them. One way to solve this problem is by segmenting the time series into stationary AR pieces (e.g., Gersch 1970; Jansen, Hasman, Lenten, and Visser 1979; Ombao et al. 2001; Melkonian, Blumenthal, and Meares 2003). We applied the multivariate Auto-PARM procedure to this bivariate time series, and the breakpoint locations and the AR orders of the resulting fit are given in Table 15. Notice that the multivariate implementation of Auto-PARM estimated the starting time for seizure for this epileptic episode at $t = 185.8$ seconds, which is in extremely close agreement with the neurologist's estimate of 185 seconds. Figure 14 shows the estimated spectrums for channel T3 (a) and channel P3 (b) based on the Auto-PARM fit in Table 15. The estimates are close to those obtained by Ombao et al. (2001), and conclusions similar to theirs can be drawn. For example, before seizure, power was concentrated at lower frequencies. During seizure, power was spread to all frequencies, whereas toward the end of seizure, the power concentration was slowly restored to the lower frequencies.

Figure 15 shows the Auto-PARM estimate of the coherence between the T3 and P3 time series channels. Again, this estimate is close to the estimate obtained by Ombao et al. (2001).

7. CONCLUSIONS

In this article we have provided a procedure for analyzing a nonstationary time series by breaking it in pieces that are modeled as AR processes. The best segmentation is obtained by minimizing a MDL criterion of the set of possible solutions via the GA. (Our procedure does not make any restrictive assumptions on this set.) The order of the AR process and the estimates of the parameters of this process is a byproduct of

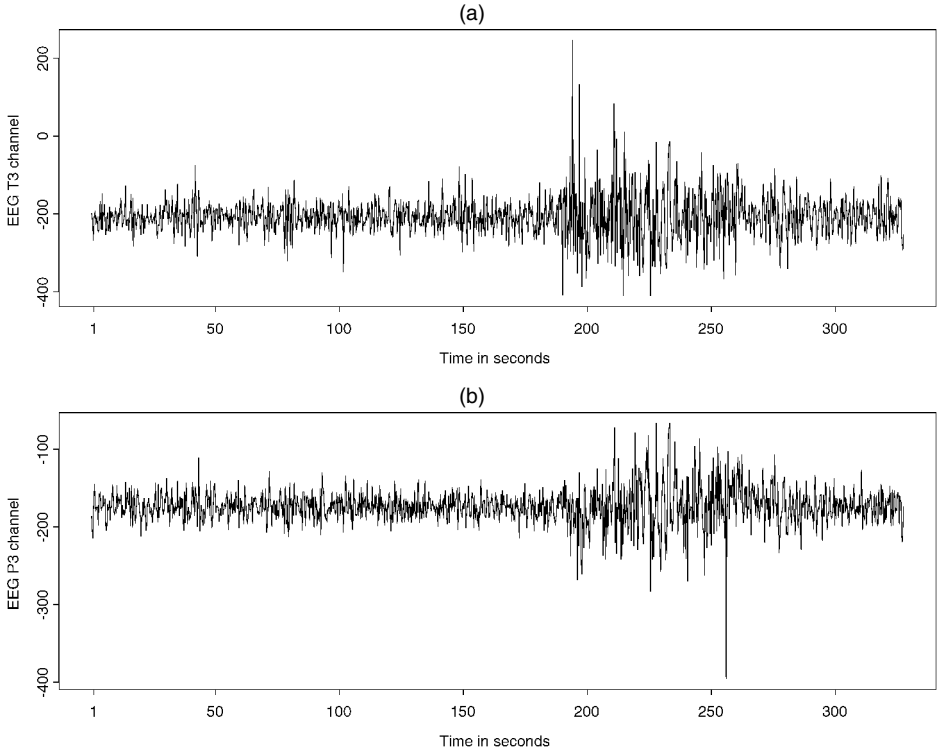

Figure 13. Bivariate EEGs of Length n = 32,768 at Channels T3 (a) and P3 (b) From a Patient Diagnosed With Left Temporal Lobe Epilepsy. (Courtesy of Dr. Beth Malow, formerly from the Department of Neurology at the University of Michigan.)

Table 15. GA Segmentation of the Bivariate Time Series From Figure 13

						j						
	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*
$\hat{\tau}_j$	1	185.8	189.6	206.1	220.9	233.0	249.0	261.6	274.6	306.0	308.4	325.8
$\hat{p}_j$	17	14	5	8	7	3	3	4	10	4	1	1

NOTE: $\hat{\tau}_j$ is given in seconds.

this procedure. As seen in the simulation experiments, the rate at which this procedure correctly segments a piecewise stationary process is high. In addition, the "quality" of the estimated time-varying spectrum obtained with the results of our method is quite good.

APPENDIX: TECHNICAL DETAILS

In this appendix we show the consistency of $\hat{\tau}_j/n$, $j = 1, \ldots, m$, when m, the number of breaks, is known. Throughout this section we denote the true value of a parameter with a "0" superscript (except for σ_j^2). Preliminary results are given in Propositions A.1–A.3, and consistency is established in Proposition A.4.

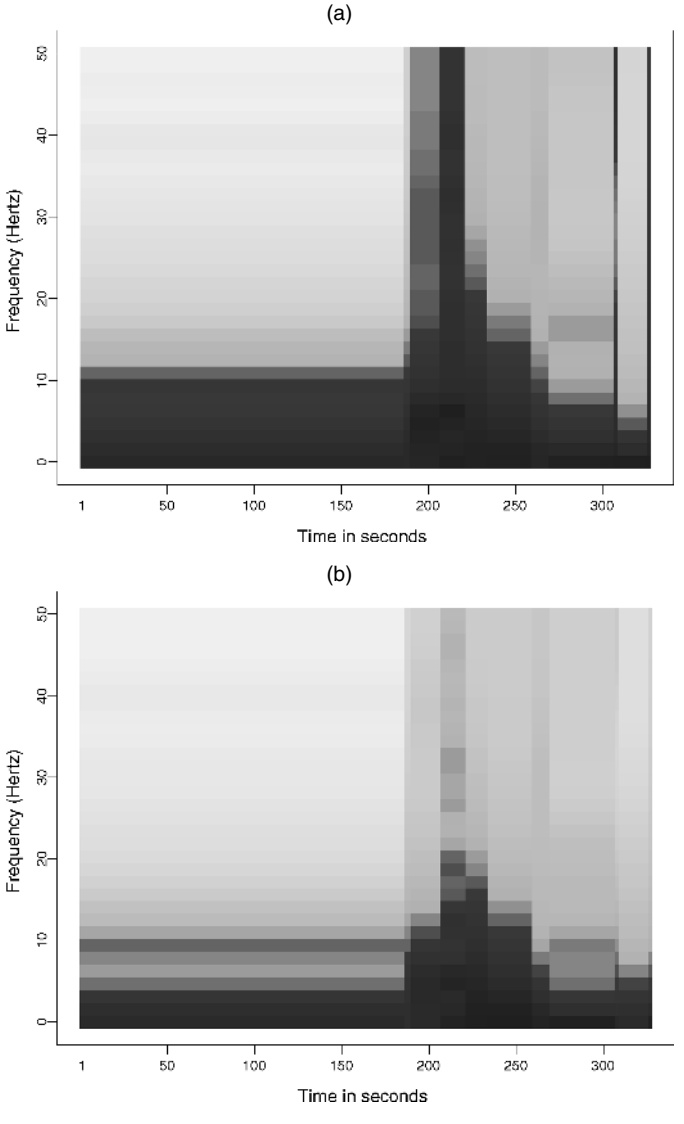

Figure 14. Estimate of the Time-Varying Log Spectra of the EEGs From Figure 13. (a) T3 channel; (b) P3 channel.

Set $\boldsymbol{\lambda} := (\lambda_1, \ldots, \lambda_m)$ and $\mathbf{p} = (p_1, \ldots, p_{m+1})$. Because m is assumed known for our asymptotic results, equation (5) can be rewritten in the compact form

$$\frac{2}{n}\mathrm{MDL}(\boldsymbol{\lambda}, \mathbf{p}) = \frac{2(m+1)}{n}\log(n) + \sum_{j=1}^{m+1}\frac{p_j+2}{n}\log n_j + \sum_{j=1}^{m+1}\frac{n_j}{n}\log(\hat{\sigma}_j^2) + o(1).$$

Proposition A.1. Suppose that $\{X_t\}$ is a stationary ergodic process with $E|X_t| < \infty$. Then, with probability 1, the process

$$S_n(s) = \frac{1}{n}\sum_{t=1}^{[ns]} X_t$$

converges to the process sEX_1 on the space $D[0, 1]$.

Proof. The argument relies on repeated application of the ergodic theorem. Let $\mathbb{Q}_{[0,1]}$ be the set of rational numbers in $[0, 1]$. For $r \in \mathbb{Q}_{[0,1]}$,

$$\frac{1}{n}\sum_{t=1}^{[nr]} X_t \to rEX_1, \quad \text{a.s.} \tag{A.1}$$

If B_r is the set of ω's for which (A.1) holds, then set

$$B = \bigcap_{r\in\mathbb{Q}_{[0,1]}} B_r,$$

and note that $P(B) = 1$. Moreover, for $\omega \in B$ and any $s \in [0, 1]$, choose $r_1, r_2 \in \mathbb{Q}_{[0,1]}$, such that $r_1 \le s \le r_2$. Hence

$$\left|\frac{1}{n}\sum_{t=1}^{[ns]} X_t - \frac{1}{n}\sum_{t=1}^{[nr_1]} X_t\right| \le \frac{1}{n}\sum_{t=[nr_1]}^{[nr_2]} |X_t| \to (r_2 - r_1)E|X_1|.$$

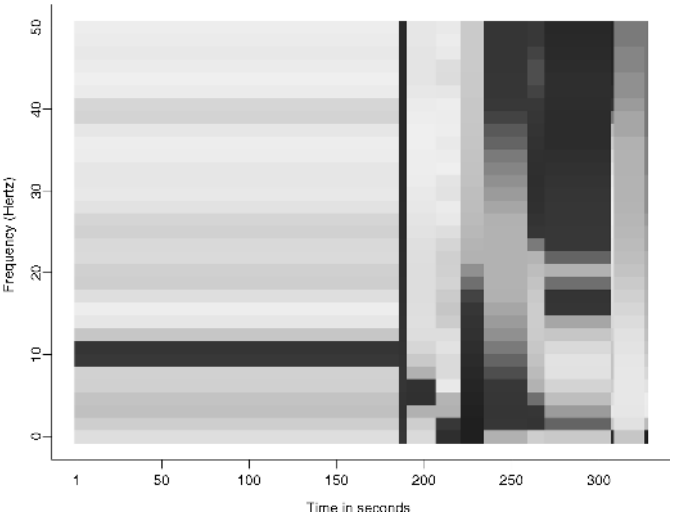

Figure 15. Estimated Coherence Between the EEGs Shown in Figure 13.

By making $|r_2 - r_1|$ arbitrarily small, it follows from the ergodic theorem that

$$\frac{1}{n}\sum_{t=1}^{[ns]} X_t \to sEX_1.$$

To establish convergence on $D[0, 1]$, it suffices to show that for $\omega \in B$,

$$\frac{1}{n}\sum_{t=1}^{[ns]} X_t \to sEX_1 \quad \text{uniformly on } [0, 1].$$

Given $\epsilon > 0$, choose $r_1, \ldots, r_m \in \mathbb{Q}_{[0,1]}$ such that $0 = r_0 < r_1 < \cdots < r_m = 1$, with $r_i - r_{i-1} < \epsilon$. Then for any $s \in [0, 1]$, $r_{i-1} < s \le r_i$ and

$$\begin{aligned}&\left|\frac{1}{n}\sum_{t=1}^{[ns]} X_t - sEX_1\right| \\ &\quad\le \left|\frac{1}{n}\sum_{t=1}^{[ns]} X_t - \frac{1}{n}\sum_{t=1}^{[nr_{i-1}]} X_t\right| + \left|\frac{1}{n}\sum_{t=1}^{[nr_{i-1}]} X_t - r_{i-1}EX_1\right| \\ &\qquad + |r_{i-1}EX_1 - sEX_1|.\end{aligned}$$

The first term is bounded by

$$\frac{1}{n}\sum_{t=[nr_{i-1}]}^{[nr_i]} |X_t| \to (r_i - r_{i-1})E|X_1| < \epsilon E|X_1|.$$

Choose n so large that this term is less than $\epsilon E|X_1|$ for $i = 1, \ldots, m$. It follows that

$$\sup_s \left|\frac{1}{n}\sum_{t=1}^{[ns]} X_t - sEX_1\right| < \epsilon E|X_1| + \epsilon + \epsilon E|X_1|,$$

for n large.

Proposition A.2. Suppose that $\{X_t\}$ is the AR(p_0) process

$$X_t = \phi_0 + \phi_1 X_{t-1} + \cdots + \phi_{t-p_0} X_{t-p_0} + \sigma\varepsilon_t, \qquad \varepsilon_t \sim \text{iid N}(0, 1).$$

For $r, s \in [0, 1]$ $(r < s)$ and $p = 0, 1, \ldots, P_0$, let $\hat{\phi}(r, s, p)$ be the Y–W estimate of the AR(p) parameter vector $\phi(p)$ based on fitting an AR(p) to the data $X_{[rn]+1}, \ldots, X_{[sn]}$. Then, with probability 1,

$$\hat{\phi}(r, s, p) \to \phi(p) \qquad \text{and} \qquad \hat{\sigma}^2(r, s, p) \to \sigma^2(p).$$

Proof. Because $\{X_t\}$ is a stationary ergodic process, $\{|X_t|\}$, $\{X_{t-i}X_{t-j}\}$ and $\{|X_{t-i}X_{t-j}|\}$ are stationary ergodic processes. By Proposition A.1, the partial sum processes for each of these processes converge to their respective limit a.s. Let B be the probability 1 set on which these partial sum processes converge. Since $\hat{\phi}(r, s, p)$ and $\hat{\sigma}^2(r, s, p)$ are continuous functions of these processes, the result follows.

Proposition A.3. Let $\{Y_t\}$ be the process defined in (1) with $\phi_{0j} = 0$. For $r, s \in [0, 1]$ $(r < s)$ and $p = 0, 1, \ldots, P_0$, let $\hat{\phi}_Y(r, s, p)$ be the Y–W estimates in fitting an AR(p) model to $Y_{[rn]+1}, \ldots, Y_{[sn]}$. Then, with probability 1,

$$\hat{\phi}_Y(r, s, p) \to \phi_Y^*(r, s, p), \qquad \hat{\sigma}_Y^2(r, s, p) \to \sigma_Y^{*2}(r, s, p),$$

where $\phi_Y^*(r, s, p)$ and $\sigma_Y^{*2}(r, s, p)$ are defined in the proof.

Proof. Let B_k^* be the probability 1 set on which

$$\frac{1}{n}\sum_{t=1}^{[ns]} X_{t,k}, \qquad \frac{1}{n}\sum_{t=1}^{[ns]} |X_{t,k}|, \qquad \frac{1}{n}\sum_{t=1}^{[ns]} X_{t-i,k}X_{t-j,k}, \qquad \text{and}$$

$$\frac{1}{n}\sum_{t=1}^{[ns]} |X_{t-i,k}X_{t-j,k}| \qquad i, j = 1, \ldots, P_0,$$

converge, $k = 1, 2, \ldots, m + 1$, and set

$$B^* = \bigcap_{k=1}^{m+1} B_k^*.$$

Let $r, s \in [0, 1]$, $r < s$. Then $r \in [\lambda_{i-1}^0, \lambda_i^0)$ and $s \in (\lambda_{i-1+k}^0, \lambda_{i+k}^0]$, $k \ge 0$. Assuming that the mean of the process $\{Y_t\}$ is 0, we have

$$\begin{aligned}\hat{\gamma}_Y(h) &:= \frac{1}{[sn] - [rn]} \sum_{t=[rn]+1}^{[sn]-h} Y_{t+h}Y_t \\ &= \frac{n}{[sn] - [rn]} \\ &\quad \times \left\{\frac{1}{n}\sum_{t=[rn]+1}^{[\lambda_i^0 n]-h} X_{t+h,i}X_{t,i} + \frac{1}{n}\sum_{t=[\lambda_i^0 n]+1}^{[\lambda_{i+1}^0 n]-h} X_{t+h,i+1}X_{t,i+1}\right. \\ &\qquad \left. + \cdots + \frac{1}{n}\sum_{t=[\lambda_{i-1+k}^0 n]+1}^{[sn]-h} X_{t+h,i+k}X_{t,i+k} + o(1)\right\}.\end{aligned}$$

Let $\gamma_i(h) := \text{cov}\{X_{t+h,i}, X_{t,i}\}$. For $\omega \in B^*$, it follows from Proposition A.2 that

$$\begin{aligned}\hat{\gamma}_Y(h) &\to \frac{\lambda_i^0 - r}{s - r}\gamma_i(h) + \frac{\lambda_{i+1}^0 - \lambda_i}{s - r}\gamma_{i+1}(h) + \cdots \\ &\qquad + \frac{s - \lambda_{i-1+k}^0}{s - r}\gamma_{i+k}(h), \\ &= a_i\gamma_i(h) + \cdots + a_{i+k}\gamma_{i+k}(h).\end{aligned}$$

Then

$$\begin{aligned}\hat{\phi}_Y(r, s, p) &= \hat{\mathbf{\Gamma}}_Y^{-1}(p)\hat{\boldsymbol{\gamma}}_Y(p) \to \left(\sum_{j=i}^{i+k} a_j\mathbf{\Gamma}_j(p)\right)^{-1}\sum_{j=i}^{i+k} a_j\boldsymbol{\gamma}_j(p) \\ &=: \phi_Y^*(r, s, p),\end{aligned}$$

where $\mathbf{\Gamma}_j(p) = \{\gamma_j(i_1 - i_2)\}_{i_1,i_2=1}^p$ and $\boldsymbol{\gamma}_j(p) = [\gamma_j(1), \ldots, \gamma_j(p)]^T$. This establishes the desired convergence for $\hat{\phi}_Y(r, s, p)$. Note that if $k = 0$, then $\phi_Y^*(r, s, p) = \phi_i(p)$. The proof of the convergence for $\hat{\sigma}_Y^2(r, s, p)$ is similar.

Proposition A.4. For the piecewise process in (1), choose $\epsilon > 0$ small such that

$$\epsilon \ll \min_{i=1,\ldots,m+1}(\lambda_i^0 - \lambda_{i-1}^0)$$

and set

$$\begin{aligned}A_\epsilon = \big\{&\boldsymbol{\lambda} \in [0, 1]^m, 0 = \lambda_0 < \lambda_1 < \lambda_2 < \cdots < \lambda_m < \lambda_{m+1} = 1, \\ &\lambda_i - \lambda_{i-1} \ge \epsilon, i = 1, 2, \ldots, m + 1\big\},\end{aligned}$$

where $m = m^0$. If

$$\hat{\boldsymbol{\lambda}}, \hat{\mathbf{p}} = \arg \min_{\substack{\boldsymbol{\lambda}\in A_\epsilon \\ 0\le \mathbf{p}\le P_0}} \frac{2}{n}\text{MDL}(\boldsymbol{\lambda}, \mathbf{p}),$$

then $\hat{\boldsymbol{\lambda}} \to \boldsymbol{\lambda}^0$ a.s.

Proof. Let B^* be the event described in the proof of Proposition A.4. We show that for each $\omega \in B^*$, $\hat{\boldsymbol{\lambda}} \to \boldsymbol{\lambda}^0$. For $\omega \in B^*$, suppose that $\hat{\boldsymbol{\lambda}} \not\to \boldsymbol{\lambda}^0$. Because the sequences are bounded, there exists a subsequence $\{n_k'\}$ such that $\hat{\boldsymbol{\lambda}} \to \boldsymbol{\lambda}^*$ and $\hat{p}_j \to p_j^*$ on the subsequence. Note that $\boldsymbol{\lambda}^* \in A_\epsilon$, because $\hat{\boldsymbol{\lambda}} \in A_\epsilon$ for all n. It follows that

$$\frac{2}{n}\text{MDL}(\hat{\boldsymbol{\lambda}}, \hat{\mathbf{p}}) \to \sum_{j=1}^{m+1}(\lambda_j^* - \lambda_{j-1}^*)\log\sigma_Y^{*2}(\lambda_{j-1}^*, \lambda_j^*, p_j^*).$$

If $\lambda_i^0 \le \lambda_{j-1}^* < \lambda_j^* \le \lambda_{i+1}^0$, then

$$\sigma_Y^{*2}(\lambda_{j-1}^*, \lambda_j^*, p_j^*) = \sigma_{i+1}^2(p_j^*) \ge \sigma_{i+1}^2, \tag{A.2}$$

with equality if and only if $p_j^* \ge p_{i+1}$. If $\lambda_{i-1}^0 \le \lambda_{j-1}^* < \lambda_i^0 < \cdots < \lambda_{i+k}^0 < \lambda_j^* \le \lambda_{i+k+1}^0$, then

$$\sigma_Y^{*2}(\lambda_{j-1}^*, \lambda_j^*, p_j^*) \ge \frac{\lambda_i^0 - \lambda_{j-1}^*}{\lambda_j^* - \lambda_{j-1}^*}\sigma_i^2 + \frac{\lambda_{i+1}^0 - \lambda_i^0}{\lambda_j^* - \lambda_{j-1}^*}\sigma_{i+1}^2 + \cdots + \frac{\lambda_j^* - \lambda_{i+k}^0}{\lambda_j^* - \lambda_{j-1}^*}\sigma_{i+k+1}^2.$$

By the concavity of the log function,

$$\begin{aligned}
&(\lambda_j^* - \lambda_{j-1}^*)\log\sigma_Y^{*2}(\lambda_{j-1}^*, \lambda_j^*, p_j^*) \\
&\quad \ge (\lambda_j^* - \lambda_{j-1}^*)\left[\frac{\lambda_i^0 - \lambda_{j-1}^*}{\lambda_j^* - \lambda_{j-1}^*}\log\sigma_i^2 + \frac{\lambda_{i+1}^0 - \lambda_i^0}{\lambda_j^* - \lambda_{j-1}^*}\log\sigma_{i+1}^2 \right. \\
&\qquad\qquad \left. + \cdots + \frac{\lambda_j^* - \lambda_{i+k}^0}{\lambda_j^* - \lambda_{j-1}^*}\log\sigma_{i+k+1}^2\right] \\
&\quad = (\lambda_i^0 - \lambda_{j-1}^*)\log\sigma_i^2 + (\lambda_{i+1}^0 - \lambda_i^0)\log\sigma_{i+1}^2 \\
&\qquad + \cdots + (\lambda_j^* - \lambda_{i+k}^0)\log\sigma_{i+k+1}^2.
\end{aligned}$$

It follows that

$$\begin{aligned}
\lim_{n\to\infty}\frac{2}{n}\mathrm{MDL}(\hat{\boldsymbol{\lambda}}, \hat{\mathbf{p}}) &> \sum_{i=1}^{m+1}(\lambda_i^0 - \lambda_{i-1}^0)\log\sigma_i^2 \\
&= \lim_{n\to\infty}\frac{2}{n}\mathrm{MDL}(\boldsymbol{\lambda}^0, \mathbf{p}^0) \\
&\ge \lim_{n\to\infty}\frac{2}{n}\mathrm{MDL}(\hat{\boldsymbol{\lambda}}, \hat{\mathbf{p}}),
\end{aligned} \tag{A.3}$$

which is a contradiction. Hence $\hat{\boldsymbol{\lambda}} \to \boldsymbol{\lambda}$ for all $\omega \in B^*$.

Notice that with probability 1, $\hat{p}_j$ cannot underestimate p_j^0. To see this, let p_j^* as in the proof of Proposition A.4, if for some j, $p_j^* < p_j^0$, then the contradiction in (A.3) is obtained again because of (A.2).

[Received November 2004. Revised June 2005.]

REFERENCES

Adak, S. (1998), "Time-Dependent Spectral Analysis of Nonstationary Time Series," *Journal of the American Statistical Association*, 93, 1488–1501.

Alba, E., and Troya, J. M. (1999), "A Survey of Parallel-Distributed Genetic Algorithms," *Complexity*, 4, 31–52.

——— (2002), "Improving Flexibility and Efficiency by Adding Parallelism to Genetic Algorithms," *Statistics and Computing*, 12, 91–114.

Bai, J., and Perron, P. (1998), "Estimating and Testing Linear Models With Multiple Structural Changes," *Econometrica*, 66, 47–78.

——— (2003), "Computation and Analysis of Multiple Structural Change Models," *Journal of Applied Econometrics*, 18, 1–22.

Brockwell, P. J., and Davis, R. A. (1991), *Time Series: Theory and Methods* (2nd ed.), New York: Springer-Verlag.

——— (2002), *Introduction to Time Series and Forecasting* (2nd ed.), New York: Springer-Verlag.

Chen, J., and Gupta, A. K. (1997), "Testing and Locating Variance Change-points With Application to Stock Prices," *Journal of the American Statistical Association*, 92, 739–747.

Davis, L. D. (1991), *Handbook of Genetic Algorithms*, New York: Van Nostrand Reinhold.

Dahlhaus, R. (1997), "Fitting Time Series Models to Nonstationary Processes," *The Annals of Statistics*, 25, 1–37.

Forrest, S. (1991), *Emergent Computation*, Cambridge, MA: MIT Press.

Gaetan, C. (2000), "Subset ARMA Model Identification Using Genetic Algorithms," *Journal of Time Series Analysis*, 21, 559–570.

Gersch, W. (1970), "Spectral Analysis of EEG's by Autoregressive Decomposition of Time Series," *Mathematical Biosciences*, 7, 205–222.

Hansen, M. H., and Yu, B. (2000), "Wavelet Thresholding via MDL for Natural Images," *IEEE Transactions on Information Theory*, 46, 1778–1788.

——— (2001), "Model Selection and the Principle of Minimum Description Length," *Journal of the American Statistical Association*, 96, 746–774.

Holland, J. (1975), *Adaptation in Natural and Artificial Systems*, Ann Arbor, MI: University of Michigan Press.

Inclan, C., and Tiao, G. C. (1994), "Use of Cumulative Sums of Squares for Retrospective Detection of Changes of Variance," *Journal of the American Statistical Association*, 89, 913–923.

Jansen, B. H., Hasman, A., Lenten, R., and Visser, S. L. (1979), "Usefulness of Autoregressive Models to Classify EEG Segments," *Biomedizinische Technik*, 24, 216–223.

Jornsten, R., and Yu, B. (2003), "Simultaneous Gene Clustering and Subset Selection for Classification via MDL," *Bioinformatics*, 19, 1100–1109.

Kim, C.-J., and Nelson, C. R. (1999), *State-Space Models With Regime Switching*, Boston: MIT Press.

Kitagawa, G., and Akaike, H. (1978), "A Procedure for the Modeling of Nonstationary Time Series," *Annals of the Institute of Statistical Mathematics*, 30, 351–363.

Lavielle, M. (1998), "Optimal Segmentation of Random Processes," *IEEE Transactions on Signal Processing*, 46, 1365–1373.

Lee, C.-B. (1997), "Estimating the Number of Change Points in Exponential Families Distributions," *Scandinavian Journal of Statistics*, 24, 201–210.

Lee, T. C. M. (2000), "A Minimum Description Length–Based Image Segmentation Procedure, and Its Comparison With a Cross–Validation Based Segmentation Procedure," *Journal of the American Statistical Association*, 95, 259–270.

Lee, T. C. M., and Wong, T. F. (2003), "Nonparametric Log-Spectrum Estimation Using Disconnected Regression Splines and Genetic Algorithms," *Signal Processing*, 83, 79–90.

Martin, W. N., Lienig, J., and Cohoon, J. P. (2000), "Island (Migration) Models: Evolutionary Algorithm Based on Punctuated Equilibria," in *Evolutionary Computation*, Vol. 2, *Advanced Algorithms and Operators*, ed. D. B. Fogel, Philadelphia: Bristol, pp. 101–124.

Melkonian, D., Blumenthal, T. D., and Meares, R. (2003), "High-Resolution Fragmentary Decomposition: A Model-Based Method of Nonstationary Electrophysiological Signal Analysis," *Journal of Neuroscience Methods*, 131, 149–159.

Ombao, H. C., Raz, J. A., Von Sachs, R., and Malow, B. A. (2001), "Automatic Statistical Analysis of Bivariate Nonstationary Time Series," *Journal of the American Statistical Association*, 96, 543–560.

Pittman, J. (2002), "Adaptive Splines and Genetic Algorithms," *Journal of Computational and Graphical Statistics*, 11, 1–24.

Punskaya, E., Andrieu, C., Doucet, A., and Fitzgerald, W. J. (2002), "Bayesian Curve Fitting Using MCMC With Applications to Signal Segmentation," *IEEE Transactions on Signal Processing*, 50, 747–758.

Rissanen, J. (1989), *Stochastic Complexity in Statistical Inquiry*, Singapore: World Scientific.

Saito, N. (1994), "Simultaneous Noise Suppression and Signal Compression Using a Library of Ortho-Normal Bases and the Minimum Description Length Criterion," in *Wavelets and Gheophysics*, eds. E. Foufoula-Georgiou and P. Kumar, New York: Academic Press, pp. 299–324.

Yao, Y.-C. (1988), "Estimating the Number of Change-Points via Schwarz's Criterion," *Statistics & Probability Letters*, 6, 181–189.