# STRUCTURAL BREAKS ESTIMATION FOR NON-STATIONARY TIME SERIES SIGNALS

*Richard A. Davis*[*], *Thomas C. M. Lee*[†] *& Gabriel A. Rodriguez-Yam*

Department of Statistics, Colorado State University

**ABSTRACT**

In this work we consider the problem of modeling a class of non-stationary time series signals using piecewise autoregressive (AR) processes. The number and locations of the piecewise autoregressive segments, as well as the orders of the respective AR processes, are assumed to be unknown. The minimum description length principle is applied to find the "best" combination of the number of the segments, the lengths of the segments, and the orders of the piecewise AR processes. A genetic algorithm is implemented to solve this difficult optimization problem. We term the resulting procedure Auto-PARM. Numerical results from both simulation experiments and real data analysis show that Auto-PARM enjoys excellent empirical properties. Consistency of Auto-PARM for break point estimation can also be shown.

KEY WORDS: Non-stationarity, change points, minimum description length principle, genetic algorithm

## 1. INTRODUCTION

In this work we consider the problem of modeling a non-stationary time series by segmenting the series into blocks of different autoregressive (AR) processes. The number of break points denoted by $m$, as well as their locations and the orders of the respective AR models are assumed to be unknown. We propose an automatic procedure for obtaining such a partition. We term the proposed procedure Auto-PARM, short for **auto**matic **p**iecewise **a**uto**r**egressive **m**odeling.

### 1.1. Piecewise Autoregressive Modeling

In order to describe the setup, for $j = 1, \ldots, m$, denote the location of the break point between the $j$-th and $(j+1)$-th AR processes as $\tau_j$, and set $\tau_0 = 1$ and $\tau_{m+1} = n+1$. Then the $j$-th piece of the series is modeled as an AR process

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

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

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

$\boldsymbol{\psi}_j := (\gamma_j, \phi_{j1}, \ldots, \phi_{j,p_j}, \sigma_j^2)$ 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 break point locations $\tau_1, \ldots, \tau_m$, and the AR orders $p_1, \ldots, p_{m+1}$. We propose an automatic procedure for obtaining such a partition. Note that once these parameters are specified, maximum likelihood estimates of the AR parameters $\boldsymbol{\psi}_j$'s are easily computed.

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), under certain conditions, argue that 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 $\omega$ is $|\mathrm{A}(t/n, \omega)|^2$, where $\mathrm{A}(u, \omega)$, $u \in [0,1]$, $\omega \in [-1/2, 1/2]$ is a continuous function in $u$. Since AR processes are dense in the class of weakly stationary (purely non-deterministic) processes, the piecewise AR process is *dense* in the class of locally stationary processes. This provides some motivation for considering models of the form in (1).

The above problem of finding a "best" combination of $m$, $\tau_j$'s and $p_j$'s can be treated as a statistical model selection problem, in which candidate models may have different numbers of parameters. In Section 2 below we solve this problem by applying the minimum description length (MDL) principle of Rissanen (1989) to *define* a best fitting model. The basic idea behind the MDL principle is that the best fitting model is the one that enables the maximum compression of the data.

As described below, the best fitted 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, as the search space (consisting of $m$, $\tau_j$'s and $p_j$'s) is enormous. For this problem, we propose using genetic algorithms (e.g., see Holland 1975). Genetic algorithms are becoming popular in statistical applications and seem particularly well suited for this MDL optimization problem as can be seen in our numerical studies. Section 3 presents the genetic algorithm that was developed for this MDL optimization problem.

### 1.2. Previous Work

Various versions of the above break point detection problem have been considered in the literature. For example, Bai and Perron (1998, 2003) examine the multiple change-point modeling for the case of multiple linear regression, Inclan and Tiao (1994) and Chen and Gupta (1997) consider the problem of detecting multiple variance change-points in a sequence of independent Gaussian random variables, and Kim and Nelson (1999) provide a summary

[*]Supported in part by NSF grant DMS-0308109

[†]Supported in part by NSF grant DMS-0203901

of various applications of the hidden Markov approach to econometrics. Kitagawa and Akaike (1978) implemented an "on-line" procedure based on AIC to determine segments. To implement their method, suppose that an autoregressive model AR($p_0$) 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($p_1$) autoregressive 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 autoregressive that results when the dataset $\{y_1, \ldots, y_{n_0+n_1}\}$ is modeled as a single autoregressive model of order $p_2$. Each $p_j, j = 0, 1, 2$ is selected among the values $0, 1, \ldots, K$ ($K$ is a predefined value) that minimizes the AIC criterion. The iteration is continued until no more data are available. Like $K$, $n_1$ is a predefined value. Ombao et al. (2001) implement 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 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, they assume that the length of the segments follow a dyadic structure; i.e., an integer power of 2. Bayesian approaches have also been studied; e.g., see Lavielle (1998) and Punskaya et al. (2002). Both procedures choose the final optimal segmentation as the one that maximizes the posterior distribution of the observed series. Numerical results suggest that both procedures enjoy excellent empirical properties. However, theoretical results supporting these procedures are lacking.

## 2. MODEL SELECTION USING MINIMUM DESCRIPTION LENGTH

### 2.1. The MDL Criterion

This section presents our results of applying 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 as $\mathcal{M}$ and any model from this class as $\mathcal{F} \in \mathcal{M}$. 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 $\mathbf{y} = (y_1, y_2, \ldots, y_n)$.

To proceed let $n_j := \tau_j - \tau_{j-1}$ be the length of the $j$th segment. Also denote the covariance matrix of $\mathbf{y}_j$ as $\mathbf{V}_j^{-1}$, and let $\hat{\mathbf{V}}_j$ be an estimate for $\mathbf{V}_j$. Then in Davis, Lee and Rodriguez-Yam (2005) it is shown that the best fitting MDL model $\hat{\mathcal{F}}$ is well approximated as the minimizer of

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

We propose selecting the best fitting model for $\mathbf{y}$ as the model $\mathcal{F} \in \mathcal{M}$ that minimizes $\mathrm{MDL}(\mathcal{F})$.

### 2.2. Consistency

Assume that there exist true values $m_0$ and $\lambda_j^0$, $j = 1, \ldots, m_0$, 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 $x$. For $j = 1, 2, \ldots, m_0$, let $\hat{\lambda}_j = \hat{\tau}_j/n$, where $\hat{\tau}_j$ is the $j$th estimated break point of the best fitting model suggested by $\mathrm{MDL}(\mathcal{F})$. In Davis et al. (2005) the following consistency result is established.

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

## 3. OPTIMIZATION USING GENETIC ALGORITHMS

As the search space is enormous, optimization of $\mathrm{MDL}(\mathcal{F})$ is a nontrivial task. In this section we propose using a genetic algorithm (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 are often called *chromosomes* and 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 chance of being chosen. Then offspring are produced by applying a *crossover* or a *mutation* operation to the chosen parents. Once a sufficient number of such second generation offspring are produced, third generation offspring are further 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*, 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 that the child receives 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 genetic algorithms different from other optimization methods. For possible variants of the crossover operation, consult 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 where randomness is introduced to alter the types of genes. Such a mutation operation prevents the algorithm from being trapped in local optima.

In order 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. Inclusion of this elitist step guarantees the monotonicity of the algorithm.

There are many variations of the above canonical GA. For example, parallel implementations can be applied to speed up the convergence rate as well as to reduce the chance of converging to sub-optimal solutions (Forrest 1991; Alba and Troya 1999). In this paper we implement the *Island Model*. Instead of running only one search in one giant population, the island model simultaneously runs $NI$ (Number-of-Islands) canonical GAs in $NI$ different sub-populations. The key feature is, periodically, a number

of individuals are migrated amongst 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 paper, we adopt the following migration policy: after every $M_i$ generations, the worst $M_N$ chromosomes from the $j$-th island are replaced by the best $M_N$ chromosomes from the $(j-1)$-th island, $j = 1, \ldots, NI$. For $j = 1$ the best $M_N$ chromosomes are migrated from the $NI$-th island. In our simulations we used $NI = 40$, $M_i = 5$, $M_N = 2$ and a sub-population size of 40.

### 3.2. Implementation Details

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

*Chromosome Representation*: The performance of a genetic algorithm 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 $\mathcal{F} \in \mathcal{M}$. That is, the break points $\tau_j$'s as well as the AR orders $p_j$'s. 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 $\delta_t$ defined as

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

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 this piece is made to have at least $m_p$ observations. This predefined integer $m_p$ is chosen to guarantee that there are enough observations for obtaining 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 $P_0$, as for complicated series one needs a large $P_0$ to capture for example seasonality, while for small series $P_0$ cannot be larger than the number of observations $n$. For all our numerical works we set $P_0 = 20$, and the corresponding minimum span $m_p$'s are listed in Table 1.

**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 |

Our empirical experience suggests that the above representation scheme, together with the minimum span constraint, is extremely effective for the purpose of using GAs to minimize $\mathrm{MDL}(\mathcal{F})$. It is most likely due to the fact that the location information of the break points 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 $\pi_B$, the probability that the "$j$-th location" of the chromosome being generated be a break point is needed. A large value of $\pi_B$ makes the initial chromosomes to have a large number of break points, thus a small value is preferred. We use $\pi_B = \min(m_p)/n = 10/n$ (in Section 4 a sensitivity analysis for this parameter is given). Once a location is declared to be a break, an AR order is selected from the uniform distribution with values 0, 1,. . ., $P_0$. The following strategy was used to generate each initial chromosome. First, select a value for $p_1$ from $\{0, \ldots, P_0\}$ with equal probabilities and set $\delta_1 = p_1$; i.e., the first AR piece is of order $p_1$. Then the next $m_{p_1} - 1$ genes $\delta_i$'s (i.e., $\delta_2$ to $\delta_{m_{p_1}}$) are set to $-1$, so that the above minimum span constraint is imposed for this first piece. Now for the next gene $\delta_{m_{p_1}+1}$ in line. It will either be initialized as a break point (i.e., assigned a non-negative integer $p_2$) with probability $\pi_B$, or it will be assigned $-1$ with probability $1 - \pi_B$. If it is to be initialized as a break point, 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 the next $m_{p_2} - 1$ $\delta_i$'s will be assigned $-1$ so that the minimum span constraint is guaranteed. On the other hand, if $\delta_{m_{p_1}+1}$ is to be assigned with $-1$, the initialization process will move to the next gene in line and decide if this gene should be a break point 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 $\delta_n$.

*Crossover and Mutation*: Once a set of initial random chromosomes is generated, new chromosomes are generated by either a crossover 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 that have smaller MDL values will have higher chances to be selected. From these two parents, the gene values $\delta_i$'s of the child chromosome will be inherited in the following manner. Firstly for $t = 1$, $\delta_t$ will take on the corresponding $\delta_t$ value from either the first or the second parent with equal probabilities. If this value is $-1$, then the same gene–inheriting process will be repeated for the next gene in line (i.e., $\delta_{t+1}$). If this value is not $-1$, then it is a non-negative integer $p_j$ denoting the AR order of the current piece. In this case the minimum span constraint will be imposed (i.e., the next $m_{p_j} - 1$ $\delta_t$'s will be set to $-1$), and the same gene–inheriting process will be applied to the next available $\delta_t$.

For mutation one child is reproduced from one parent. Again, this process starts with $t = 1$, and every $\delta_t$ (subject to the minimum span constraint) can take on one of the following three possible values: (i) with probability $\pi_P$ it will take the corresponding $\delta_t$ value from the parent, (ii) with probability $\pi_N$ it will take the value $-1$, and (iii) with probability $1 - \pi_P - \pi_N$, it will take the a new randomly generated AR order $p_j$. In this paper we set $\pi_P = 0.3$ and $\pi_N = 0.3$.

*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 the total number of migrations exceeds 20, this best chromosome is taken as the solution to this optimization problem.

## 4. SIMULATIONS

Three sets of simulation experiments were conducted to evaluate the practical performances of Auto-PARM. The experimental setups of the first two simulations are taken from Ombao et al. (2001), which were used to test their Auto-SLEX procedure. In the first simulation, the pieces of the true process follow a dyadic structure; i.e., the length of each segment is a integer power of 2. In the second simulation the true process does not contain any struc-

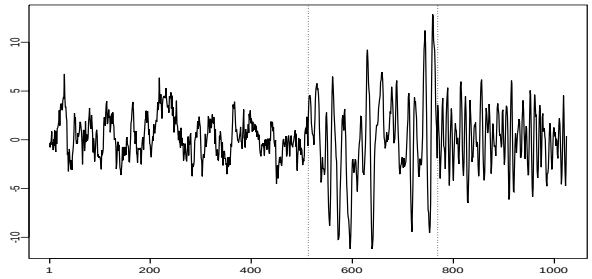

**Fig. 1**. *A realization from the piecewise stationary process in (3).*

tural breaks, but its time-varying spectrum changes very slowly over time. In the last simulation the process contains three pieces, one of which is an ARMA(1,1) process and another is a MA(1) process.

### 4.1. Piecewise Stationary Process with Dyadic Structure

In this simulation example, the target non-stationary series is generated with the following model

$$Y_t = \begin{cases} 0.9Y_{t-1} + \varepsilon_t, & \text{if } 1 \le t \le 512, \\ 1.69Y_{t-1} - 0.81Y_{t-2} + \varepsilon_t, & \text{if } 513 \le t \le 768, \\ 1.32Y_{t-1} - 0.81Y_{t-2} + \varepsilon_t, & \text{if } 769 \le t \le 1024, \end{cases} \quad (3)$$

where $\varepsilon_t \sim$ iid $\mathrm{N}(0,1)$. The main feature of this model is that the lengths of the pieces are a power of 2. This is in fact ideally suited for the Auto-SLEX procedure of Ombao et al. (2001). A typical realization of this process is shown in Figure 1. We applied Auto-PARM to this realization and obtained two break points located at $\hat{\tau}_1 = 512$ and $\hat{\tau}_2 = 769$, indicated by the dotted vertical lines in the figure. The Auto-PARM correctly identified the AR orders ($\hat{p_1}$=1, $\hat{p_2}$=2 and $\hat{p_3}$=3) for this realization. Our implementation of Auto-PARM, which is written in Compaq Visual Fortran, took 2.34 seconds on a 1.6 Ghz intel pentium M processor to obtain the above estimates.

Next, 200 realizations of the process in (3) were simulated and Auto-PARM was applied to segment each of these realizations. Table 2 lists the percentages of the fitted number of segments. For comparative purposes, the corresponding values of the Auto-SLEX method, taken from Table 2 of Ombao et al. (2001), are also listed. Notice that for all 200 realizations Auto-PARM gave the correct number of segments, while Auto-SLEX gave the correct segmentation for only 60% of the realizations. Table 2 also reports, for each $\hat{m}$, the mean and standard deviation of $\hat{\lambda}_j := (\hat{\tau}_j - 1)/n$, $j = 2, \ldots, \hat{m}$, where $\hat{\tau}_j$ is the Auto-PARM estimate of $\tau_j$. For convenience we will refer to $\hat{\lambda}_j$ as a *relative* break point. From Table 2 one can see that the practical performance of Auto-PARM for the above piecewise stationary process is extremely good, especially for locating the break points.

### 4.2. Slowly Varying AR(2) Process

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

$$Y_t = a_t Y_{t-1} - 0.81 Y_{t-2} + \varepsilon_t, \quad t = 1, 2, \ldots, 1024, \quad (4)$$

| Number of segments | (%) | Auto-SLEX break points | (%) | Auto-PARM break points mean | std |
|---|---|---|---|---|---|
| 2 | 0 | 1/2 | 0 | | |
| 3 | 60.0 | 2/4, 3/4 | 100.0 | 0.500 | 0.002 |
| | | | | 0.742 | 0.007 |
| 4 | 34.0 | 1/4, 2/4, 3/4 | 0 | | |
| 5 | 5.0 | 2/8, 4/8, 5/8, 6/8, 7/8 | 0 | | |
| $\geq 6$ | 1.0 | | 0 | | |

**Table 2**. *Summary of the estimated break points from both the Auto-SLEX and Auto-PARM procedures for process (3). Numbers from Auto-SLEX are taken from Table 2 of Ombao et al. (2001). For Auto-PARM the means and standard errors of the relative break points are also reported.*

where $a_t = 0.8[1 - 0.5\cos(\pi t/1024)]$ and $\varepsilon_t \sim$ iid $\mathrm{N}(0,1)$. A typical realization of this process is shown in Figure 2, while the spectrum of this process is shown on the left panel of Figure 3 (darker shades represent higher power).

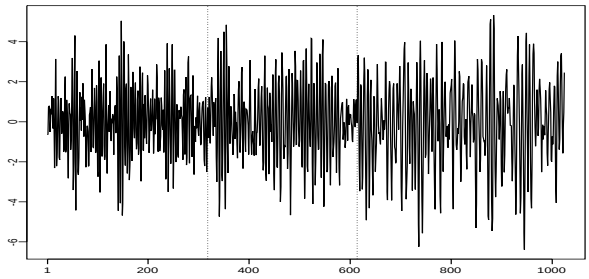

**Fig. 2**. *Realization from the process in (4).*

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

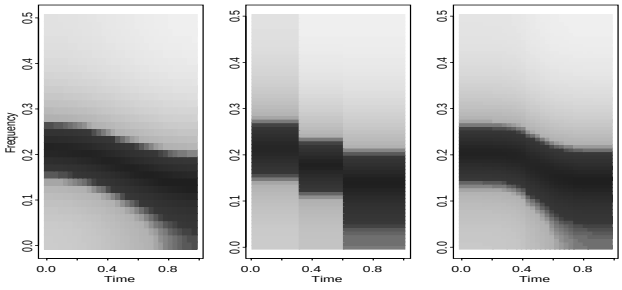


**Fig. 3**. *Left: True time-varying log-spectrum of process in (4). Center: Auto-PARM log-spectrum estimate for the realization from Figure 2. Right: Average of the Auto-PARM log-spectrum estimates obtained from 100 realizations.*

Next we generated 100 realizations of the above process, and

| Number of segments | Auto-SLEX (%) | Auto-SLEX ASE | Number of segments | Auto-PARM (%) | Auto-PARM break points mean | Auto-PARM break points std | Auto-PARM ASE |
|---|---|---|---|---|---|---|---|
| $\leq 4$ | 14.0 | 0.238 | 1 | 0 | | | - |
| | | (0.030) | | | | | - |
| 5 | 27.0 | 0.228 | 2 | 44.0 | 0.493 | 0.057 | 0.131 |
| | | (0.025) | | | | | (0.015) |
| 6 | 35.0 | 0.232 | 3 | 56.0 | 0.372 | 0.078 | 0.080 |
| | | (0.029) | | | 0.652 | 0.078 | (0.016) |
| 7 | 18.0 | 0.243 | $\geq 4$ | 0 | | | |
| | | (0.033) | | | | | |
| 8 | 15.0 | 0.269 | | | | | |
| | | (0.040) | | | | | |
| $\geq 9$ | 1.0 | 0.308 | | | | | |
| | | - | | | | | |
| All | 100.0 | 0.239 | All | 100.0 | | | 0.102 |
| | | (0.033) | | | | | (0.030) |

**Table 3**. *The ASEs values from the Auto-PARM and the Auto-SLEX estimates computed from realizations of (4). Numbers inside parentheses are standard errors of the ASE values.*

the corresponding Auto-PARM estimates were obtained. Since 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

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

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

The number of segments, locations of the break points and the ASEs of the Auto-PARM estimates are summarized in Table 3. Also listed in Table 3 are the ASE values of the Auto-SLEX procedure copied from Table 3 of Ombao et al. (2001). From Table 3 the following two main observations can be made. First, for each of the simulated processes, Auto-PARM produces either two or three segments that are of roughly the same length, while the Auto-SLEX procedure tends to split the process into a larger number of segments. Second, the ASE values of Auto-PARM are smaller than those from Auto-SLEX. In order to show there is a kind of "consistency" property of the Auto-PARM estimates, we computed the average of all the time-varying spectra of the 100 Auto-PARM estimates, the averaged spectrum is displayed in the right panel of Figure 3 and looks remarkably similar to the true time varying spectrum. Lastly in Table 4 we summarize the Auto-PARM estimates of the AR orders for the above process. Notice that most of the segments were modeled as AR(2) processes.

| Order | 0 | 1 | 2 | 3 | 4 | $\geq 5$ |
|---|---|---|---|---|---|---|
| 2-segment realizations | | | | | | |
| $p_1$ | 0 | 0 | 95.5 | 2.3 | 2.3 | 0 |
| $p_2$ | 0 | 0 | 90.9 | 6.8 | 2.3 | 0 |
| 3-segment realizations | | | | | | |
| $p_1$ | 0 | 0 | 100.0 | 0 | 0 | 0 |
| $p_2$ | 0 | 0 | 92.9 | 5.4 | 1.8 | 0 |
| $p_3$ | 0 | 0 | 83.9 | 14.3 | 1.8 | 0 |

**Table 4**. *Relative frequencies of the AR order selected by Auto-PARM for the realizations from the process (4).*

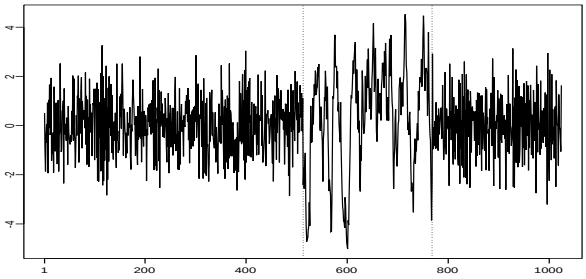


**Fig. 4**. *A realization from the piecewise stationary process in (5).*

### 4.3. Piecewise ARMA process

Recall that the Auto-PARM procedure assumes 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 data generating model given by

$$Y_t = \begin{cases} -0.9Y_{t-1} + \varepsilon_t + 0.7\varepsilon_{t-1}, & \text{if } 1 \leq t \leq 512, \\ 0.9Y_{t-1} + \varepsilon_t, & \text{if } 513 \leq t \leq 768, \\ +\varepsilon_t - 0.7\varepsilon_{t-1}, & \text{if } 769 \leq t \leq 1024, \end{cases} \quad (5)$$

where $\varepsilon_t \sim$ iid $\text{N}(0,1)$. Notice that the first piece is an ARMA(1,1) process while the last piece is a MA(1) process. A typical realization of this process is shown in Figure 4. The Auto-PARM procedure was applied to this realization. Three pieces were obtained. The break points are at $\hat{\tau}_1 = 513$ and $\hat{\tau}_2 = 769$ (dotted vertical lines in this figure), while the order of the AR processes are 4, 1 and 2 respectively. The total run time for this fit was 1.53 seconds. The time-varying spectrum based on the model found by Auto-PARM is reasonable close to the true spectrum even though two of the segments are not AR processes.

To assess the large sample behavior of Auto-PARM, 200 realizations from (5) were generated, and the corresponding Auto-PARM estimates were obtained. An encouraging result is that for all 200 realizations, Auto-PARM always gave the correct number of stationary segments. The estimates of the break point locations are summarized in Table 5. In Table 6 we show 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.

| Number of segments | % | relative break points mean | relative break points std |
|---|---|---|---|
| 3 | 100.0 | 0.50 | 0.005 |
| | | 0.75 | 0.003 |

**Table 5**. *Summary of Auto-PARM estimated break points obtained from 200 realizations from the process in (5).*

| Order | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | $\geq 8$ |
|---|---|---|---|---|---|---|---|---|---|
| $p_1$ | 0 | 4.0 | 22.5 | 40.0 | 23.5 | 8.5 | 1.0 | 0.5 | 0 |
| $p_2$ | 0 | 89.5 | 8.5 | 1.5 | 0.5 | 0 | 0 | 0 | 0 |
| $p_3$ | 0 | 0.5 | 22.0 | 45.0 | 19.5 | 7.5 | 4.5 | 1.0 | 0 |

**Table 6**. *Relative frequencies of the AR order selected by Auto-PARM for the realizations from the process (5).*

## 5. APPLICATION: SPEECH SIGNAL SEGMENTATION

The Auto-PARM procedure was applied to analyze a human speech signal which is the recording of the word "greasy". This signal contains 5762 observations and is shown at the top panel of Figure 5. This non-stationary time series was also analyzed by the Auto-SLEX procedure of Ombao et al. (2001). The Auto-PARM

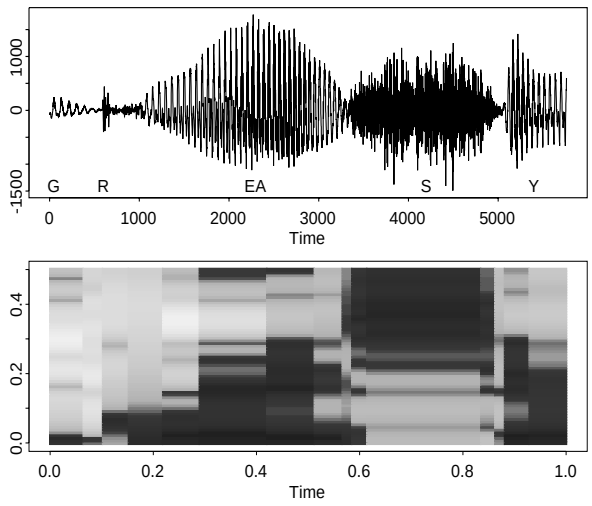


**Fig. 5**. *Top panel: Speech signal. Bottom panel: GA estimate of the time-varying log spectrum.*

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 at the bottom panel of Figure 5. From this figure, one can see that the signal is roughly divided in segments that correspond to "G", "R", "EA", "S", and "Y". The information conveyed in this figure closely matches that from 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. CONCLUSIONS

In this work we provided a procedure, termed Auto-PARM, to analyze a non-stationary time series signal by breaking it into pieces that are modeled as autoregressive processes. The best segmentation is obtained by minimizing the MDL (Risannen, 1989) of the set of possible solutions via the genetic algorithm. The order of the autoregressive process in which each piece is modeled and the estimates of the parameters of this process is a byproduct of this procedure. As seen in the simulation experiments, the rate in which this procedure segments correctly a piece-wise stationary process is high. Also, the quality of the estimated time-varying spectra produced by the procedure is very reasonable.

# 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 Algorithm," *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.

Chen, J., and Gupta, A. K. (1997), "Testing and Locating Variance Changepoints 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.

Davis, R. A., Lee, T. C. M., and Rodriguez-Yam, G. A. (2005), "Structural Breaks Estimation for Non-stationary Time Series Models", unpublished manuscript.

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.

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.

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 Non-stationary 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.

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, 101-124. Philadelphia: Bristol.

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.

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.