UNIVERSIDAD CARLOS III DE MADRID working papers

Working Paper 09-80
Statistics and Econometrics Series 25
December 2009

Departamento de Estadística
Universidad Carlos III de Madrid
Calle Madrid, 126
28903 Getafe (Spain)
Fax (34) 91 624-98-49

TIME SERIES SEGMENTATION BY CUSUM, AUTOSLEX AND AUTOPARM METHODS

Ana Badagián, Regina Kaiser and Daniel Peña*

Abstract

Time series segmentation has many applications in several disciplines as neurology, cardiology, speech, geology and others. Many time series in this fields do not behave as stationary and the usual transformations to linearity cannot be used. This paper describes and evaluates different methods for segmenting non-stationary time series.

We propose a modification of the algorithm in Lee et al. (2003) which is designed to searching for a unique change in the parameters of a time series, in order to find more than one change using an iterative procedure. We evaluate the performance of three approaches for segmenting time series: AutoSLEX (Ombao et al., 2002), AutoPARM (Davis et al., 2006) and the iterative cusum method mentioned above and referred as ICM. The evaluation of each methodology consists of two steps. First, we compute how many times each procedure fails in segmenting stationary processes properly. Second, we analyze the effect of different change patterns by counting how many times the corresponding methodology correctly segments a piecewise stationary process.

ICM method has a better performance than AutoSLEX for piecewise stationary processes. AutoPARM presents a very satisfactory behaviour. The performance of the three methods is illustrated with time series datasets of neurology and speech.

Keywords: Time Series Segmentation, AutoSLEX, AutoPARM, Cusum Methods

* Departamento de Estadística, Universidad Carlos III de Madrid, C/ Madrid 126, 28903 Getafe (Madrid), e-mail adresses: abadagia@est-econ.uc3m.es; kaiser@est-econ.uc3m.es; dpena@est-econ.uc3m.es.

1 Introduction

Time series segmentation has many applications in several disciplines as neurology, cardiology, speech, geology and others. Many series in these fields do not behave as stationary and the usual transformations to linearity can not be used. This paper describes and evaluates different methods for segmenting non-stationary time series.

The goal of the segmentation is to obtain intervals, partitions, blocks or segments in which the time series behaves as approximately stationary. Thus, the segmentation pursues: 1) to find the periods of stability and homogeneity in the behavior of the process, 2) to identify the moments of change, 3) to represent the regularities and features of each segment or block and, 4) to use this information in order to determine the pattern moving the non-stationary time series.

Two of the most recent methods are AutoSLEX (Ombao et al. (2002)) and AutoPARM (Davis et al. (2006)). Both of them have an important computational burden and are based on complex techniques. In the case of AutoSLEX, the use of non-parametrics, frequency domain and dyadic structures complicates the method. For AutoPARM, although it is based on parametric models, the need of a genetic algorithm makes difficult the process. In this paper we propose the use of cusum methods to obtain the stationary intervals, since they usually built into intuitive procedures. Cusum method have been referred in the literature of time series in order to find breakpoints and which, in general, intensive and complicated computer methods are not required. Following the initial idea in Lee et al. (2003) we propose a modification consisting in an iterative cusum method -in what follows ICM-, which is designed to search and identify multiple moments of parameters change. We also evaluate and compare the performance of AutoSLEX and AutoPARM to and with ICM.

The organization of the paper is as follows. In Section 2, AutoSLEX, AutoPARM procedures. In Section 3 we introduce cusum methods and propose some modifications to the hypothesis test presented in Lee et al. (2003). In Section 4 we apply ICM, AutoSLEX and AutoPARM to several stationary datasets to evaluate how each method performs when it should not segment the process. Moreover, we present the application of each method to piecewise stationary processes and evaluates their performance. In Section 5 we compare the results of applying ICM, AutoSLEX and AutoPARM to real datasets of different disciplines: a neurology dataset, EEGT3 (the recordings from the left temporal lobe during an epileptic seizure of a patient) and a linguistic dataset consisting of the speech recording of the word GREASY. Finally, Section 6 presents the conclusions.

2 Methodologies for segmenting a time series

Segmentation could be performed using different methods. In what follows we describe AutoSLEX (Ombao et al. (2002)) and AutoPARM (Davis et al. (2006)) procedures for segmenting time series.

2.1 AutoSLEX

Fourier vectors are perfectly localized in frequency and hence are ideal at representing stationary time series. However, they cannot adequately represent non stationary time series, i.e., the time series with spectra that change over time. SLEX vectors are simultaneously orthogonal and localized in time and frequency. They are constructed by applying a projection operator on the Fourier vectors. The action of a projection operator on any periodic vector is identical to applying two specially constructed smooth windows to the Fourier vectors. Then, a SLEX basis vector $\phi_{S,\omega}(t)$ for the time block $[\alpha_0, \alpha_1]$ and oscillating at frequency ω, has support on the discrete time block $S = \{\alpha_0 - \epsilon + 1, ..., \alpha_1 - \epsilon\}$ and has the form

$$\phi_{S,\omega}(t) = \Psi_{S,+}(t) \exp\left(i2\pi\omega\frac{t}{|S|}\right) + \Psi_{S,-}(t) \exp\left(-i2\pi\omega\frac{t}{|S|}\right) \tag{1}$$

where $\omega \in [-1/2, 1/2]$, $|S| = \alpha_1 - \alpha_0$, ϵ is a small overlap between two consecutive time blocks which ensures smoothness in the transition between them. In Huang et al. (2004), the windows $\Psi_{S,+}(t)$ and $\Psi_{S,-}(t)$ take the form

$$\Psi_{S,+}(t) = r^2\left(\frac{t-\alpha_0}{\epsilon}\right) r^2\left(\frac{\alpha_1 - t}{\epsilon}\right)$$
$$\Psi_{S,-}(t) = r\left(\frac{t-\alpha_0}{\epsilon}\right) r\left(\frac{\alpha_0 - t}{\epsilon}\right) - r\left(\frac{t-\alpha_1}{\epsilon}\right) r\left(\frac{\alpha_1 - t}{\epsilon}\right)$$

where $r(.)$ is called a "rising cut-off function". Huang et al. (2004) use the sine rising cut-off function

$$r(u) = \sin\left(\frac{\pi}{4}(1+u)\right), \qquad \text{where } u \in [-1, 1]. \tag{2}$$

Other types of rising cut-off functions may be used (see Wickerhauser (1994) for details).

The SLEX library is a collection of bases, each having orthogonal vectors with time support that is obtained by segmenting the time series, of length T, in a dyadic manner. The library is constructed by

first specifying the finest resolution level J or the length of the smallest time block $T/2^J$. At resolution level j, with $j = 0, ..., J$, time series is divided into 2^j overlapping blocks. The amount of overlap ϵ is the same for all levels j, and is equal to $\epsilon = T/2^{J+1}$. With this restriction the SLEX vectors remain orthogonal despite the overlap. Let $S(j,b)$ the block b on level j and $M_j = T/2^j$ the length of the block j. The SLEX vectors on block $S(j,b)$ are allowed to oscillate at different fundamental frequencies $\omega_k = k/M_j$ where $k = -M_j/2 + 1, ..., M_j/2$. For example, if $J = 2$, the SLEX library consists of 5 orthogonal bases: i) $S(0,0)$; ii) $S(1,0) \cup S(1,1)$; iii) $S(2,0) \cup S(2,1) \cup S(2,2) \cup S(2,3)$; iv) $S(1,0) \cup S(2,2) \cup S(2,3)$; v) $S(2,0) \cup S(2,1) \cup S(1,1)$. Therefor, the SLEX basis vectors are allowed to have different lengths of support (different time and frequency resolutions).

The SLEX transform consists of the set of coefficients corresponding to all the SLEX vectors defined in the library. The SLEX coefficients on block $S = S(j,b)$ are defined by

$$\hat{\theta}_{S,k} = \frac{1}{\sqrt{M_j}} \sum_t X_{t,T} \bar{\phi}_{S,\omega_k}(t), \tag{3}$$

where the fundamental frequency is $\omega_k = k/M_j$ and $k = -M_j/2 + 1, ..., M_j/2$. The SLEX periodogram, an analogue of the Fourier periodogram for a stationary process is defined to be

$$\hat{\alpha}_{S,k} = \left|\hat{\theta}_{S,k}\right|^2. \tag{4}$$

After computing the SLEX transform a well-defined cost is computed at each of the blocks. For example, the cost function of the block $S(j,b)$ could be

$$\text{Cost}(j,b) = \sum_{k=-M_j/2+1}^{M_j/2} \log\hat{\alpha}_{S,k} + \beta\sqrt{M_j}, \tag{5}$$

where β is a complexity penalty parameter. The penalty term $\beta\sqrt{M_j}$ safeguards the procedure from obtaining a segmentation that has too many or too few blocks. A small value of β leads to a procedure that tends to select a segmentation with too many small blocks, and this favors the existence of less bias due to the non stationarity. However, having less observations within each block leads to inflated variances of the estimates. A large value of β, on the other hand, leads to a procedure that tends to select a segmentation with very few blocks. Although variance of the estimates is reduced, having too few blocks may lead to bias due to non stationarity (i.e. error due to not splitting a non stationary block). The penalty parameter β can be either approximated or computed via a data-driven procedure. Ombao et al. (2002) set $\beta = 1$ motivated by Donoho et al. (1998).

The cost for a particular segmentation of the time series is the sum of the costs at all the blocks defining that segmentation. The Best Basis Algorithm is applied to the SLEX transform to obtain the unique orthonormal transform in the SLEX library that has the smallest cost. So, the Best Basis in the SLEX library is the segmentation having the smallest cost.

Let B_T the best basis selected from the SLEX library and $\cup S_i$ be the blocks in B_T (a particular dyadic segmentation of the time series). Define M_i to be the numbers of points on the block S_i. Let J_T to be the highest time resolution level in B_T, i.e., the smallest time block in B_T has length $T/2^{J_T}$. The frequencies defined on S_i are the grid frequencies $\omega_{k_i} = k_i/M_i$ for $k_i = -M_i/2+1, ..., M_i/2$. The spectral representation of $X_{t,T}$ is

$$X_{t,T} = \sum_{\cup S_i \sim B_T} \frac{1}{\sqrt{M_i}} \sum_{k=-M_i/2+1}^{M_i/2} \theta_{i,k,T} \phi_{i,k}(t) z_{i,k} \tag{6}$$

where $\theta_{i,k,T}$ is the transfer function on time block S_i and frequency k; $\phi_{i,k}$ is the SLEX basis vector oscillating at frequency k and having support at block S_i; and $z_{i,k}$ is a orthonormal random process with finite fourth moment.

The SLEX spectrum is defined analogously to the spectrum of a stationary process. It is the square of the modulus of the time varying transfer function. It is defined on rescaled time $[0, 1]$. Let u be in an interval $I \in [0, 1]$ such that $[uT]$ is in some time block S_i on B_T. The SLEX spectrum is $f_T(u, \omega_k) = |\theta_{i,k,T}|^2 \Leftrightarrow [uT] \in S_i$. Note that for a fixed frequency ω_k, $f_T(u, \omega_k)$ is constant within each time block. This is because for each fixed T the SLEX model gives an explicit partitioning of the time-frequency plane, as determined by the blocks $\cup_i S_i$ in the basis B_T.

2.2 AutoPARM

Davis et al. (2006) proposed an automatic procedure called AutoPARM for modelling a non stationary time series by segmenting the series into blocks of different autoregressive processes. Let τ_j the breakpoint between the j-th and the $(j+1)$st AR processes, with $j = 1, ..., m$, $\tau_0 = 1$ and $\tau_{m+1} = n+1$. Thus, the j-th piece of the series is modelled as an AR process,

$$Y_t = X_{t,j}, \qquad \tau_{j-1} \leq t < \tau_j, \tag{7}$$

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

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

where $\psi_j := \left(\gamma_j, \phi_{j1}, ..., \phi_{j,p_j}, \sigma_j^2\right)$ is the parameter vector corresponding to this AR(p_j) process and the sequence $\{\epsilon_t\}$ is iid with mean 0 and variance 1. Under this model equation, it is assumed that some aspect of the behavior of the time series is changing at various times. Such a change might be a shift in the mean, a change in the variance and/or a change in the dependence structure of the process.

The idea is, given the time series $\{y_i\}_{i=1}^n$, the objective is to obtain the best-fitting model from this class of piecewise AR processes. In other words, the proposal is to find the best combination of the number of pieces, $m+1$, the location of the breakpoints $\tau_1, ..., \tau_m$ and the AR orders in each piece $p_1, ..., p_{m+1}$. Once these parameters are estimated, the estimate of the evolutive spectrum is obtained by substitution.

To solve the problem of selecting the appropriate model is applied the minimum description length (MDL) principle of Rissanen (1989). The basic idea behind this principle is that the best-fitting model is the one that makes the maximum compression of the data possible.

Let $\mathcal{M}$ the complete class of piecewise autoregressive models and $\mathcal{F}$ any model corresponding to this class $\mathcal{M}$. The MDL principle defines as the best model of $\mathcal{M}$ as the one that produces the shortest code length that completely describes the observed data $\mathbf{y} = (y_1, y_2, ..., y_n)$. The code length of an object is defined as the memory space required to store that object. In the applications of MDL principle, a classical way to store $\mathbf{y}$ is to split $\mathbf{y}$ in 2 components: the adjusted model $\hat{\mathcal{F}}$ and the portion of $\mathbf{y}$ not explained by the model, 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 the object z using model $\mathcal{F}$, then is obtained the following decomposition:

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

where $CL_{\mathcal{F}}\left(\hat{\mathcal{F}}\right)$ represent the code length of the fitted model and $CL_{\mathcal{F}}\left(\hat{\mathbf{e}}/\hat{\mathcal{F}}\right)$ and is the code length of the corresponding residuals conditional on the fitted model $\hat{\mathcal{F}}$. Very briefly, the MDL principle suggests that the best piecewise AR model $\hat{\mathcal{F}}$ is the minimizer of $CL_{\mathcal{F}}(\mathbf{y})$. The authors decompose $CL_{\mathcal{F}}\left(\hat{\mathcal{F}}\right)$ in:

$$CL_{\mathcal{F}}(m) + CL_{\mathcal{F}}(\tau_1, ..., \tau_m) + CL_{\mathcal{F}}(p_1, ..., p_{m+1}) + CL_{\mathcal{F}}\left(\hat{\psi}_1, ..., \hat{\psi}_{m+1}\right)$$

$$= CL_{\mathcal{F}}(m) + CL_{\mathcal{F}}(n_1, ..., n_{m+1}) + CL_{\mathcal{F}}(p_1, ..., p_{m+1}) + CL_{\mathcal{F}}\left(\hat{\psi}_1, ..., \hat{\psi}_{m+1}\right).$$

Behind the last equation is the idea that complete knowledge of $(\tau_1, ..., \tau_m)$ implies the complete knowledge of $(n_1, ..., n_{m+1})$ and *vice versa*. In general, to store a not bounded integer I, is required approximately $\log_2 I$ bits. Then, $CL_{\mathcal{F}}(m) = \log_2 m$ and $CL_{\mathcal{F}}(p_j) = \log_2 p_j$. If the object I has a known bound, I_U, is required approximately $\log_2 I_u$ bits. Since all n_j are bounded by n, $CL_{\mathcal{F}}(n_j) = \log_2 n$ for all j. To calculate $CL_{\mathcal{F}}\left(\hat{\psi}_j\right)$ a result of Rissanen is used. It says: A maximum likelihood estimator of a real parameter computed using N observations can be encoded with $\frac{1}{2}\log_2 N$ bits. Since each of the $p_j + 2$ parameters of $\hat{\psi}_j$ is computed with n_j observations,

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

Combining these results is obtained the equation (8):

$$CL_{\mathcal{F}}\left(\hat{\mathcal{F}}\right) = \log_2 m + (m+1)\log_2 n + \sum_{j=1}^{m+1}\log_2 p_j + \sum_{j=1}^{m+1}\frac{p_j + 2}{2}\log_2 n_j. \tag{8}$$

The code length for the residuals, $CL_{\mathcal{F}}\left(\hat{\mathbf{e}}/\hat{\mathcal{F}}\right)$ is obtained using a classical result of Rissanen, who demonstrated that the code length of $\hat{e}$ is equal to the negative of the log-likelihood of the fitted model $\hat{\mathcal{F}}$. Let $\mathbf{y}_j := \left(y_{\tau_{j-1}}, ..., y_{\tau_j - 1}\right)$ the vector of observations of the piece j in (7). For simplicity, is assumed that μ_j, the mean of the piece j in (7) is $\mathbf{0}$ and the covariance matrix is denoted by $\mathbf{V}_j^{-1} = \mathrm{cov}\{\mathbf{y}_j\}$, where $\hat{\mathbf{V}}_j$ is an estimator of $\mathbf{V}_j$. Even the ϵ_j's are not assumed to be normal, the inference is based on a Gaussian likelihood (quasi-likelihood procedure). Assuming the independence of the pieces, the Gaussian likelihood of a piecewise process is given by

$$L(m, \tau_0, \tau_1, ..., \tau_m, p_1, ..., p_{m+1}, \psi_1, ..., \psi_{m+1}; \mathbf{y}) = \prod_{j=1}^{m+1}(2\pi)^{-n_j/2}\left|\mathbf{V}_j\right|^{1/2}\exp\left\{-\frac{1}{2}\mathbf{y}_j^T\mathbf{V}_j\mathbf{y}_j\right\},$$

and then, the code length of $\hat{e}$ given the model $\hat{\mathcal{F}}$ is

$$-\log_2 L\left(m, \tau_0, \tau_1, ..., \tau_m, p_1, ..., p_{m+1}, \hat{\psi}_1, ..., \hat{\psi}_{m+1}; \mathbf{y}\right) = \tag{9}$$
$$\sum_{j=1}^{m+1}\left\{\frac{n_j}{2}\log(2\pi) - \frac{1}{2}\log\left|\hat{\mathbf{V}}_j\right| + \frac{1}{2}\mathbf{y}_j^T\mathbf{V}_j\mathbf{y}_j\right\}\log_2 e.$$

Combining (8) and (9) and using logarithm base e rather than 2, is obtained the following approximation of $CL_{\mathcal{F}}(\mathbf{y})$:

$$\log m + (m+1)\log n + \sum_{j=1}^{m+1} \log p_j + \sum_{j=1}^{m+1} \frac{p_j+2}{2} \log n_j + \tag{10}$$
$$+ \sum_{j=1}^{m+1} \left\{ \frac{n_j}{2} \log(2\pi) - \frac{1}{2} \log \left| \hat{\mathbf{V}}_j \right| + \frac{1}{2} \mathbf{y}_j^T \mathbf{V}_j \mathbf{y}_j \right\}.$$

Using the approximation of the likelihood for the autoregressive models $-2\log$ (likelihood) by $n_j \log \hat{\sigma}_j^2$, where $\hat{\sigma}_j^2$ is the Yule Walker estimator of σ_j^2 (Brockwell and Davis (1991)), MDL is defined as:

$$MDL(m, \tau_1, ..., \tau_m, p_1, ..., p_{m+1}) =$$
$$\log m + (m+1)\log n + \sum_{j=1}^{m+1} \log p_j + \sum_{j=1}^{m+1} \frac{p_j+2}{2} \log n_j + \sum_{j=1}^{m+1} \frac{n_j}{2} \log\left(2\pi\hat{\sigma}_j^2\right). \tag{11}$$

Davis et al. (2006) demonstrated that the best-fitted model obtained by the minimization of the MDL principle is a non trivial issue because the search space composed by m, τ_j's and p_j's has a enormous dimension. To solve this problem, they use an genetic algorithm. These algorithms make a population of individuals "to evolve" subject to random actions similar to those that characterize the biologic evolution (i.e. crossover and genetic mutation), as well as a selection process following a certain criteria which determines the most adapted or best individuals that survive the process, and the less adapted or the "worst" one, who are ruled out.

The genetic algorithm in its canonical version has the following idea: an initial set or population of candidate solutions to one optimization problem is represented by vectors called chromosomes. The chromosomes "parents" are randomly selected from the initial population with a probability inversely proportional to their MDL. This mean that a chromosome with a low MDL will have a greater likelihood to be selected. The second generation (the first "child" chromosomes) are obtained under the operations of *crossover* or *mutation* of the selected parents. Once enough members of the second generation are obtained, it begins the production of the children of the third generation. This process continues producing new generations, with the expectation of the gradual improvement of the values of the objective function moving closer to the optimal value.

The crossover operation is the feature that distinguish the genetic algorithms from the other optimization procedures. The chromosome child is created by the mixture of two parents. The new solution created typically shares many of the best characteristics of its parents. One typical strategy for the mixture is to assign to each location of the child's gen the same probability of receipting the

corresponding father's or mother's gen.

In the mutation, one child chromosome is created from only one parent chromosome. The child is very similar to the parent, except for a small number of gens in which is introduced randomness to reach the changes. The mutation operation prevents the algorithm to be trapped in local optima.

To preserve the best chromosome of the current generation, there exists the elitist stage. The worst chromosome of the next generation is replaced with the best chromosome of the current generation. This procedure guarantees the monotonicity of the algorithm.

There exist a lot of variations of the canonical genetic algorithm, pursuing the goal of the improvement the convergence rates and to reduce obtaining suboptimal solutions. Davis et al. (2006) implement the island model, which runs NI searchs (number of islands) simultaneously applying canonical genetic algorithms in NI different subpopulations rather than performing the search in only one enormous population. The key feature is that periodically a number of individuals emigrate between islands according a certain migration rule. In Davis *et al.* (2006) after M_i generations, the worst M_N chromosomes of the jth island are replaced with the best M_N chromosomes of the $(j-1)$st island, with $j = 2, ..., NI$. For $j = 1$, the best M_N chromosomes emigrate from the NIth island.

3 Cusum methods for detecting changes in the data generating process of a time series

Cusum Methods are useful for detecting the locations of change points (Inclán and Tiao (1994)). They have been utilized for testing for a change in mean, variance and distribution function. In this paper we propose to use cusum methods to obtain the approximately stationary intervals, since they are an intuitive procedure referred in the literature of time series to find breakpoints and intensive computer methods are not required. Thus, we modify the algorithm in Lee et al. (2003) which is designed to searching for an unique change in the parameters of a time series when the underlying distribution is completely unknown, in order to find more than one change using an iterative cusum procedure. The fact that the procedure is not based on a known distribution allows to deal with asymetric or non-constant variances data sets. In following subsections we describe the basic method developed by Lee et al. (2003) and develop one iterative algorithm to search for multiples parameters changes in time series.

3.1 Cusum method for detecting an unique change in the parameters of the generating process

The basic idea in Lee et al. (2003) is the following: consider the stationary time series $\{x_t; t = 0, \pm 1, \pm 2, ...\}$, and let $\theta = (\theta_1, ..., \theta_J)$ the parameter vector, which will be examined for constancy, e.g. the mean, variance, autocovariances, etc. The hypotheses to test are:

$H_0 : \theta$ does not change for $x_1, ..., x_n$ versus H_1: not H_0.

Let $\hat{\theta}_k$ be the estimator of θ based on $x_1, ..., x_k$. Lee et al. (2003) investigate the differences $\hat{\theta}_k - \hat{\theta}_n$, for constructing a cusum test. They assume that $\hat{\theta_k}$ obtained from $x_1, ..., x_n$ satisfies the following

$$\sqrt{k}\left(\hat{\theta}_k - \theta\right) = \frac{1}{\sqrt{k}} \sum_{1}^{k} I_t + \Delta_k,$$

where $I_t : I_t(\theta) = (I_{1,t}, ..., I_{J,t})'$ forms stationary martingale differences with respect to a filtration $\{\mathcal{F}_t\}$, namely for every t,

$$E\left(I_t / \mathcal{F}_{t-1}\right) = 0 \quad \text{a.s.},$$

and $\Delta_k = (\Delta_{1,t}, ..., \Delta_{J,t})'$. Let $\Gamma = \text{Var}(I_t)$ be the covariance matrix of I_t. Lee et al. (2003) define the statistic Tn by computing

$$T_k = \frac{k^2}{n}\left(\hat{\theta}_k - \hat{\theta}_n\right)\Gamma^{-1}\left(\hat{\theta}_k - \hat{\theta}_n\right) \tag{12}$$

and taking the maximum value for $k = J, ..., n$.

$$T_n = \max_{J \leq k \leq n} T_k \tag{13}$$

which in some regular conditions, and, under H_0 holds:

$$T_n \rightarrow^d \sup_{0 \leq s \leq 1} \sum_{j=1}^{J} \left(W_j^o(s)\right)^2. \tag{14}$$

where $\mathbf{W}_J^o(s) = (W_1^o(s), ..., W_J^o(s))'$ is a J-dimensional standard Brownian bridge. We reject H_0 if T_n is large. To calculate the critical values of the distribution they provide the tables through a Monte Carlo simulation, since it is not easy to calculate the critical values analytically. For this task,

they generate the random numbers ϵ_t following the standard normal distribution and compute the empirical quantiles based on the random variables

$$\mathcal{U}_{n,J} = \max_{1\leq k\leq n} \sum_{j=1}^{J} \left\{ n^{-1/2} \sum_{i=1}^{k} \epsilon_{i,j} - n^{-1/2} \left(\frac{k}{n} \sum_{i=1}^{n} \epsilon_{i,j} \right) \right\}^2 .$$

Lee et al. (2003) provide the critical values for the significance levels $\alpha = 0.01, 0.05, 0.1$ and $J = 1, ..., 10$, which are obtained by replicating 10000 simulated $\mathcal{U}_{1000,J}$.

Lee et al. (2003) exposed the RCA(1) model, to analyze the existence of changes in the coefficient of an AR(1) process, in its variance and in the variance of the innovation term. RCA models have been studied to investigate the effects of random perturbations of a dynamical system (Tong (1990)) in the fields of biology, enginnering, finance and economics.

Let $\{x_t; t = 0, \pm 1, \pm 2, ...\}$ be the time series of the RCA(1) model

$$x_t = (\phi + b_t)\, x_{t-1} + \epsilon_t, \tag{15}$$

where $\begin{pmatrix} b_t \\ \epsilon_t \end{pmatrix} \sim \text{iid} \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \omega^2 & 0 \\ 0 & \sigma^2 \end{pmatrix} \right)$.
A sufficient condition for the strict stationarity and ergodicity of x_t is $\phi^2 + \omega^2 < 1$ (Nicholls et al. (1982)).

Lee et al. (2003) considered the problem of testing for a change of the parameter vector $\theta = \left(\phi, \omega^2, \sigma^2\right)'$ based on a conditional LSE $\hat{\theta}$. Using the sample $x_1, ..., x_n$ with $x_0 = 0$ they intended to test the following hypotheses:

$$H_0 : \left(\phi, \omega^2, \sigma^2\right)' \quad \text{is constant over } x_1, ..., x_n \text{ versus } \quad H_1 : \text{not } H_0.$$

In order to perform the test, they construct the cusum statistic, with $\hat{\theta}_k = \left(\hat{\phi}_k, \hat{\omega}_k^2, \hat{\sigma}_k^2\right)'$, where $\hat{\phi}_k$ is the estimator of ϕ obtained by the minimization of $\sum_{t=1}^{k} \left(x_t - \phi x_{t-1}\right)^2$, and $\hat{\omega}_k^2$ and $\hat{\sigma}_k^2$ are the estimators of ω^2 and σ^2 defined as the minimizers of $\sum_{t=1}^{k} \left(\hat{u}_{k,t}^2 - \omega^2 x_{t-1}^2 - \sigma^2\right)^2$, with $\hat{u}_{k,t} = x_t - \hat{\phi}_k x_{t-1}$. Moreover, Γ is a matrix of dimension 3x3 composed by

$$
\begin{aligned}
\Gamma_{11} &= \frac{\omega_2 Ex_1^4 + \sigma^2 Ex_1^2}{\left(Ex_1^2\right)^2}, \\
\Gamma_{22} &= \left(Ex_1^4 - \left(Ex_1^2\right)^2\right)^{-2} \left(\left(Eb_1^4 - \omega^4\right)\left(Ex_1^8 - 2Ex_1^2 Ex_1^6 + \left(Ex_1^2\right)^2 Ex_1^4\right)\right. \\
&+ \left. 4\omega^2\sigma^2\left(Ex_1^6 - 2Ex_1^2 Ex_1^4 + \left(Ex_1^2\right)^3\right) + \left(E\epsilon_1^4 - \sigma^4\right)\left(Ex_1^4 - \left(Ex_1^2\right)^2\right)\right), \\
\Gamma_{33} &= \left(Eb_1^4 - \omega^4\right)\left(Ex_1^4 - \frac{2Ex_1^2\left(Ex_1^6 - Ex_1^2 Ex_1^4\right)}{Ex_1^4 - \left(Ex_1^2\right)^2}\right) \\
&- 4\omega^2\sigma^2 Ex_1^2 + E\epsilon_1^4 - \sigma^4 + \left(Ex_1^2\right)^2 \Gamma_{22}, \\
\Gamma_{12} &= \frac{Eb_1^3 Ex_1^6 - Eb_1^3 Ex_1^2 Ex_1^4 + E\epsilon_1^3 Ex_1^3}{Ex_1^2 Ex_1^4 - \left(Ex_1^3\right)^3} \\
\Gamma_{13} &= \frac{-Eb_1^3 Ex_1^2 Ex_1^6 + Eb_1^3\left(Ex_1^4\right)^2 - E\epsilon_1^3 Ex_1^2 Ex_1^3}{Ex_1^2 Ex_1^4 - \left(Ex_1^2\right)^3}, \\
\Gamma_{23} &= \frac{\left(Eb_1^4 - \omega^4\right)\left(Ex_1^6 - Ex_1^2 Ex_1^4\right)}{Ex_1^4 - \left(Ex_1^2\right)^2} + 4\omega^2\sigma^2 - Ex_1^2\Gamma_{22}.
\end{aligned}
$$

In order to obtain Γ they estimate $E\epsilon_t^3$, Eb_t^3, $E\epsilon_t^4$, and Eb_t^4 minimizing $\sum_{t=1}^n \left(\hat{u}_t^3 - x_{t-1}^3 Eb_t^3 + E\epsilon_t^3\right)$ and $\sum_{t=1}^n \left(\hat{u}_t^3 - x_{t-1}^3 Eb_t^3 + E\epsilon_t^3\right)$. Plug in those estimators and $n^{-1}\sum_{t=1}^n x_t^k$, $k = 2, 3, 4, 6, 8$ into Γ_{ij}, they obtain a consistent estimator of Γ.

3.2 Iterative cusum method to detect multiple parameters changes

In real time series and more in lengthy time series of very high frequency data the probability of changes afecting the structure of the data is high and therefor to consider the posibility of only one change is not realistic. In this section we start considering the hypothesis test presented in Lee et al. (2003) and following the idea of a sequential search of changes in Inclán and Tiao (1994) we propose an iterative cusum method (ICM). In general, ICM searches for changes in the parameters of the model.

To simplify the exposition, we consider a particular case in which the assumed model is a RCA(1), and we search for several changes in ϕ, σ^2 and ω^2. The results are easily extended to other models. The algorithm proposed consists of:

- Step 0: Center the time series in the sample mean and apply the following steps to it. Set $t_1 = 1$. Model the whole centered time series with an RCA(1) obtaining $\hat{\phi}_n$, $\hat{\sigma^2}_n$ and $\hat{\omega^2}_n$.

- Step 1: Compute $T_k(t_1 : T)$ for $k = J..., n$[1]. Let $k*$ the point where $\max_k |T_k(t_1 : T)|$ is obtained, called T_n and if $T_n > D*$ -being $D*$ the critical value with $1-p$ level of significance-, there is a shift in the time series at time $k*$ and the procedure continues with the step 2a. If $T_n(t_1 : T) \leq D*$ the algorithm stops.

- Step 2a: Let $t_2 = k*$. Estimate again a RCA(1) beginning in t_1 and finishing in t_2-1. We obtain new values for $\hat{\phi}_n$, $\hat{\sigma^2}_n$ and $\hat{\omega^2}_n$. Calculate $T_k(t_1 : t_2)$ and finally the new T_n. If $T_n(t_1 : t_2) > D*$, then we have a new point of change. Again, let $k*$ the point where $|T_k(t_1 : t_2)|$ is maximized. Repeat this step until $T_n(t_1 : t_2) < D*$. Then, the first point of change is $k_{first} = t_2$.

- Step 2b: Let $k*(t_1 : T)$ the point of change found in step (1), set $t_1 = k*(t_1 : T) + 1$, estimate the RCA(1) using the observations of the periods t_1 to T, calculate $T_k(t_1 : T)$ using observations and evaluate whether its maximum is greater than $D*$ or not. If the condition holds, the period $k*$ where we have the maximum is the period of the shift. Now, set $t_1 = k*$ and repeat this step until $T_n(t_1 : T) < D*$. The last period of change will be $k_{last} = t_1 - 1$ where $T_n(t_1 : T) < D*$.

- Step 2c: If $k_{first} = k_{last}$ there is only one shift in the time series. If $k_{first} < k_{last}$ repeat Step 1 and Step 2 with $t_1 = k_{first} + 1$ and $T = k_{last}$. Call N_T the number of shifts found.

- Step 3: Sort the breakpoint in increasing order. Let c_p be the vectors of breakpoints with $c_{p_0} = 0$ and $c_{p_{N_t+1}} = T$. Check all the breakpoint by calculating

$$T_k\left(c_{p_{j-1}} + 1 : c_{p_{j+1}}\right), \; j = 1, 2, ..., N_T \tag{16}$$

  If $T_k\left(c_{p_j}\right) > D*$ keep the point. Else eliminate it.

Since we work with lengthy time series, we compute the critical values for 0.05 and 0.01 significance levels and $J = 1, ..., 4$, but we investigate the sensitiveness of the statistic to the length of the time series. We perform 10000 replications of the statistic for $T = 2^k$, where $k = 9, ..., 15$. The results are presented in the table 1. We found that the critical values are not too sensitive to changes in the time series length, although they are to the number of parameters to which the test is applied.

[1] In parenthesis there is the interval used for the computation of the statistic. That is the same for both T_k and T_n

Table 1: Critical values for 0.05 and 0.01 significance levels and $J = 1, ..., 4$

	J			
T	1	2	3	4
512	1.76	2.41	2.97	3.37
	2.48	3.27	3.87	4.37
1024	1.78	2.42	2.97	3.48
	2.51	3.27	3.94	4.48
2048	1.79	2.44	2.97	3.51
	2.59	3.31	3.96	4.49
4096	1.81	2.44	3	3.49
	2.56	3.31	3.9	4.35
8192	1.85	2.49	3.04	3.49
	2.65	3.29	3.91	4.54
2048	1.82	2.5	2.98	3.5
	2.53	3.39	3.89	4.53
32768	1.86	2.5	3.03	3.51
	2.66	3.38	3.9	4.53

4 Monte Carlo simulations

In this section we evaluate the performance of the three methods presented above. First, we compute how many times the corresponding methodology segments a stationary process. The length of simulated series is set equal to 2^{12}. We generate 1000 of the following processes y_t:

- a white noise,

$$y_t = a_t \qquad \text{where} \quad a_t \sim iid\,(0,1)\,, \tag{17}$$

- autoregressive of order one (AR(1))processes

$$y_t = \phi y_{t-1} + a_t \qquad \text{where} \quad y_0 = 0 \quad \text{and} \quad a_t \sim iid\,(0,1)\,, \tag{18}$$

  where the parameter ϕ is set equal to 0.8, -0.8, 0.5, and -0.5, and,

- moving average of order one (MA(1)) processes

$$y_t = \theta a_{t-1} + a_t \qquad \text{where} \quad a_t \sim iid\,(0,1) \tag{19}$$

  where the parameter θ is set equal to 0.8, -0.8, 0.5 and -0.5.

The second evaluation of the methods consists of computing how many times the corresponding methodology correctly segments a piecewise stationary process. Since each process has two stationary segments or blocks the goodness of the results consists on the finding of these two stationary

segments or blocks. Thus, we observe if the methodology find these two segments or blocks and if any change occurs near the correct breakpoint.

The piecewise processes used in order to compute the type II error probability have length equal to 2^{12}. They are:

- AR(1) (and MA(1)) with parameter 0.8 changing to -0.8 in the observation $t = 2048$.
- AR(1) (and MA(1)) with parameter 0.5 changing to -0.5 in the observation $t = 2048$.
- AR(1) (and MA(1)) with parameter 0.9 changing to -0.2 in the observation $t = 2048$.

The observation 2048 is just in the middle of the sample. This location of the change is set in a arbitrary way and favors the dyadic structure used by AutoSLEX.

Tables 2 to 4 present the results for stationary processes. In those tables $\alpha*$ represents the proportion of wrong segmented stationary processes. The performances of AutoSLEX and AutoPARM methods are very satisfactory. Applying both methods to stationary process we obtain only one block or segment in the most of the cases and only a small percentage of processes are segmented in two blocks. It seems that AutoPARM has the best performance with a very small frequency of errors for stationary processes (only two cases in all the 9000 simulated stationary processes).

Table 2: Number of blocks or segments applying AutoSLEX to stationary processes

Processes	1 block	2 blocks	α^*
White Noise	1000	0	0
AR(1) ϕ =0.8	995	5	0.005
AR(1) ϕ =-0.8	995	5	0.005
AR(1) ϕ =0.5	990	10	0.01
AR(1) ϕ =-0.5	982	18	0.018
MA(1) θ =0.8	988	12	0.012
MA(1) θ =-0.8	994	6	0.006
MA(1) θ =0.5	984	16	0.016
MA(1) θ =-0.5	990	10	0.01

Table 3: Number of blocks or segments applying AutoPARM to stationary processes

Processes	1 block	2 blocks	α^*
White Noise	1000	0	0
AR(1) ϕ =0.8	999	1	0.001
AR(1) ϕ =-0.8	1000	0	0
AR(1) ϕ =0.5	1000	0	0
AR(1) ϕ =-0.5	1000	0	0
MA(1) θ =0.8	1000	0	0
MA(1) θ =-0.8	1000	0	0
MA(1) θ =0.5	1000	0	0
MA(1) θ =-0.5	999	1	0.001

Table 4: Number of blocks or segments applying ICM to stationary processes

Processes	1 block	2 or more blocks	α^*
White Noise	941	59	0.059
AR(1) ϕ =0.8	907	93	0.093
AR(1) ϕ =-0.8	906	94	0.094
AR(1) ϕ =0.5	902	98	0.098
AR(1) ϕ =-0.5	900	100	0.1
MA(1) θ =0.8	901	99	0.099
MA(1) θ =-0.8	908	92	0.092
MA(1) θ =0.5	906	94	0.094
MA(1) θ =-0.5	903	97	0.097

The proportion of wrong segmented stationary processes by ICM is always less than 0.1. We investigate the hypothesis that the correlation structure of the process could influence the segmentation performed by ICM method. We assumed a RCA(1) as the true model to clean out the process of its serial correlation. The results for MA(1) and AR(1) processes are similar leading to the conclusion that the serial correlation seems to be not important to perform a appropriate segmentation using ICM.

In tables 5 to 7 we show our results for the piecewise stationary processes, showing how many breakpoints found belong to the interval 2048 ± 100.

Table 5: Number of piecewise stationary processes with changes inside the interval 2048 ± 100 applying AutoSLEX

Processes	1 changes	2 changes	$\leq$ 3 changes
AR(1): 0.8 to -0.8	671	141	188
AR(1): 0.5 to -0.5	895	60	45
AR(1): 0.9 to -0.2	765	105	130
MA(1): 0.8 to -0.8	623	131	246
MA(1): 0.5 to -0.5	881	72	47
MA(1): 0.9 to -0.2	848	92	60

Table 6: Number of piecewise stationary processes with changes inside the interval 2048 ± 100 applying AutoPARM

Processes	0 changes	1 change	2 changes
AR(1): 0.8 to -0.8	4	986	10
AR(1): 0.5 to -0.5	0	998	2
AR(1): 0.9 to -0.2	5	990	5
MA(1): 0.8 to -0.8	1	994	5
MA(1): 0.5 to -0.5	0	998	2
MA(1): 0.9 to -0.2	1	991	8

Table 7: Number of piecewise stationary processes with changes inside the interval 2048 ± 100 applying ICM

Processes	0 changes	1 changes	2 changes	≤ 3 changes
AR(1): 0.8 to -0.8	27	797	155	21
AR(1): 0.5 to -0.5	17	839	129	15
AR(1): 0.9 to -0.2	9	856	122	13
MA(1): 0.8 to -0.8	46	891	60	3
MA(1): 0.5 to -0.5	65	932	3	0
MA(1): 0.9 to -0.2	69	931	0	0

AutoSLEX and ICM segment in more blocks than AutoPARM the piecewise stationary processes. However, for most of the simulated series AutoSLEX and ICM found only one breakpoint. In the case of AutoSLEX, given the dyadic segmentation, for all of the series which AutoSLEX found more than two blocks, it necesarily found the correct breakpoint located in the middle of the sample.

The resulting segmentations using AutoPARM are the most satisfactory. The proportion of no changes inside the interval 2048 ± 100 varies from 0.001 to 0.05 for the different simulated piecewise processes. Moreover, only in 0.006 to 0.06 proportion of cases AutoPARM segments in 3 blocks (2 breakpoints) when the process has only 2 blocks. Finally, the location of the breakpoints is well detected with a very high frequency ($\geq 98.6\%$) by AutoPARM.

Applying ICM method to piecewise stationary processes we noticed a good performance, i.e. only one change in the interval 2048 ± 100, in 79.7 to 93.2% of the simulated processes, meanwhile AutoSLEX achieves only a index of 67.1 to 89.5%.

Monte Carlo simulations show very good results for AutoSLEX and AutoPARM since the proportion of wrong segmented stationary processes are lower than 0.0018 and 0.001 respectively. For ICM this proportion is smaller than 0.10. When the process has two blocks or segments, it seems that AutoSLEX and ICM tend to segment it more than AutoPARM. However, since we put the change in he middle of the time series, in various of the processes which AutoSLEX segments more than it should,

it finds the correct location of the correct change. Moreover, ICM method had a better performance than AutoSLEX for piecewise stationary processes, obtaining a greater rate of correct unique changes.

5 Real datasets

The performance of the methods is illustrated with two datasets. We compare the results of applying ICM, AutoSLEX and AutoPARM to real datasets of different disciplines:

- a neurology dataset denoted by EEGT3, which represents the recordings from the left temporal lobe during an epileptic seizure of a patient with 32768 data observed at the sampling rate of 100 Hz.;
- a speech dataset consisting in the recording of the word GREASY with 8192 observations.

Both time series have been analyzed by Ombao et al. (2002) and Davis et al. (2006) and are presented in Figures 1 and 2 respectively. We apply the three methodologies to both datasets and present the resulting segmentations in Figures 3 and 4 respectively. Breakpoints are showed with dotted lines.

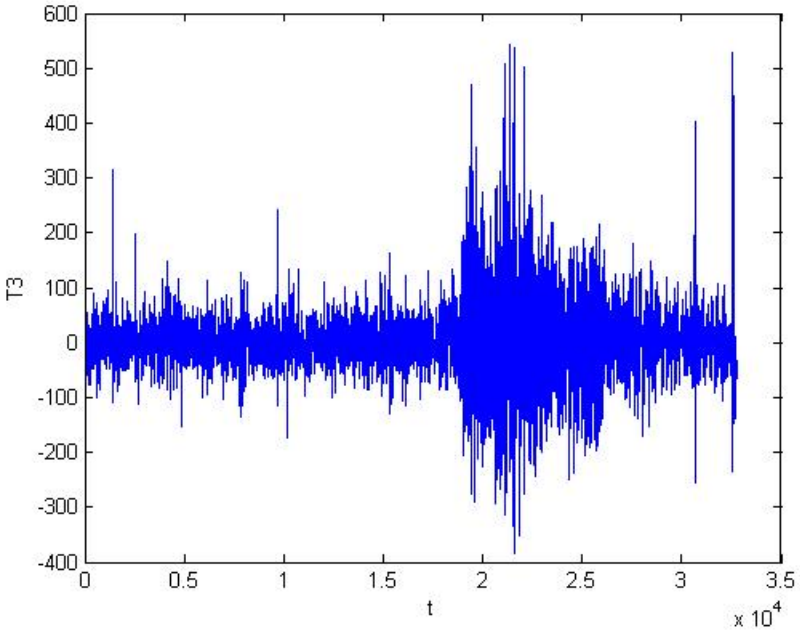

Figure 1: EEGT3: Recordings from the left temporal lobe during a epileptic seizure of a patient. $T = 32768$

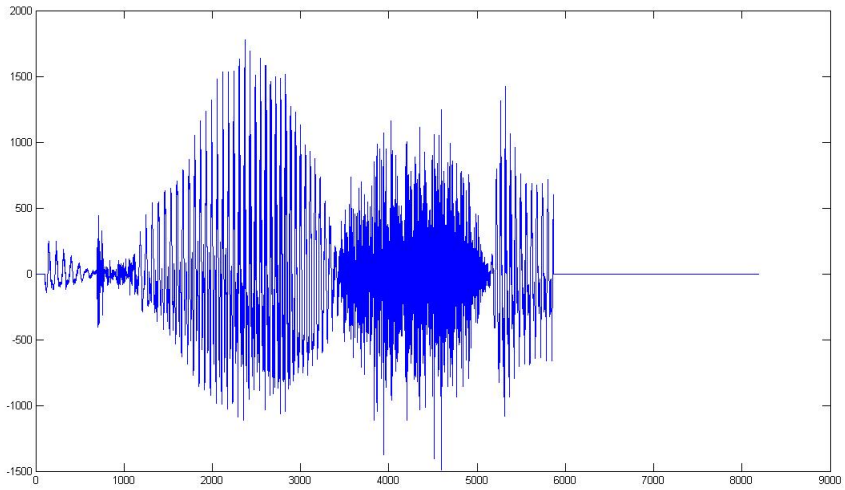

Figure 2: Speech signal representing the recording of the word GREASY. $T = 8192$.

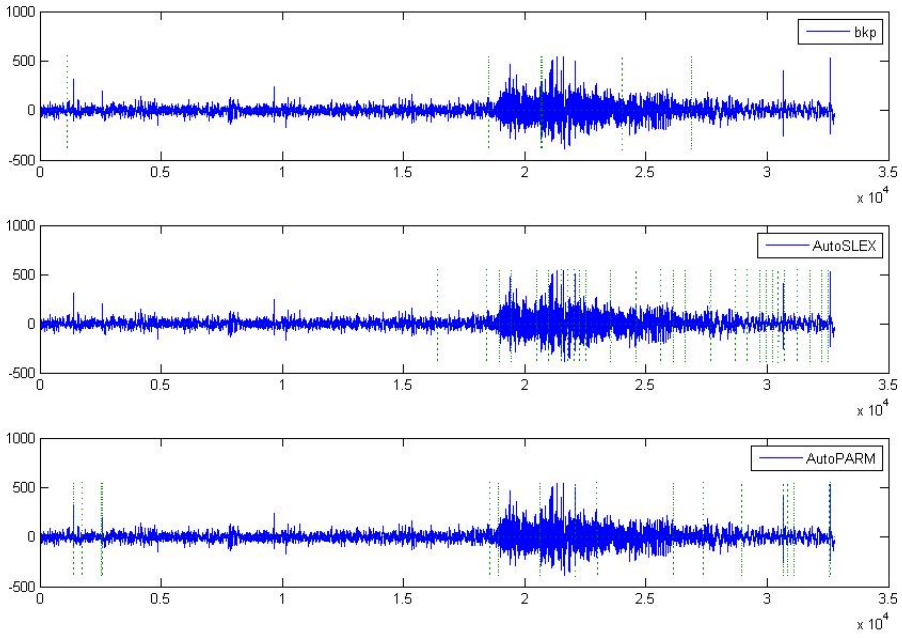

Figure 3: Breakpoints in EEGT3 estimated by ICM, AutoSLEX and AutoPARM

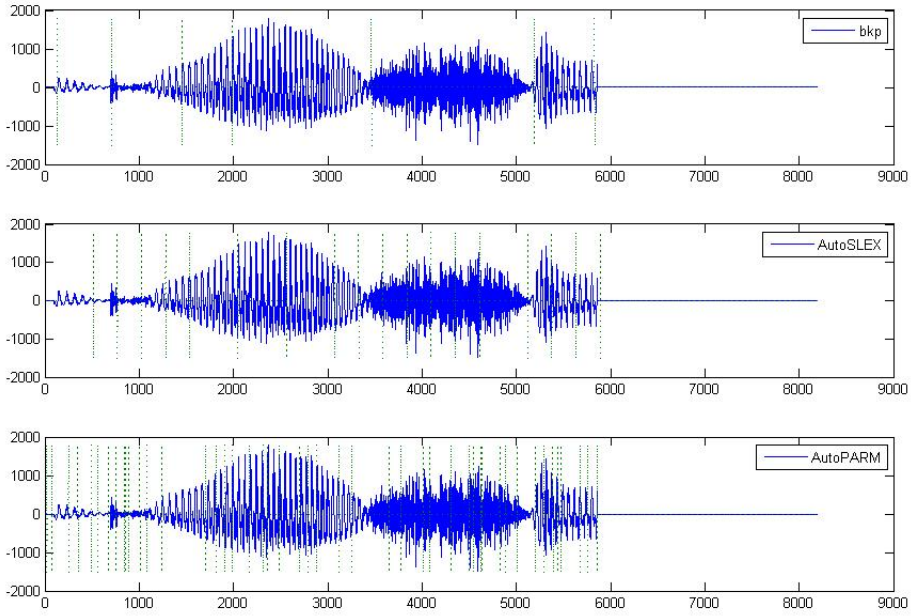

Figure 4: Breakpoints of GREASY estimated by ICM, AutoLSEX and AutoPARM

For EEGT3, AutoSLEX found 28 breakpoints, AutoPARM 18, ICM 5. The breakpoints estimated by ICM's are very similar to some of those found by AutoSLEX and AutoPARM. The segmentation performed by ICM is concentrated (4 of the 5 changes) in the interval of high volatility. AutoSLEX finds that the first half of the time series is stationary. The seizure, or at least a different behavior of the series, seems to begin in $t = 16384$. The other methodologies show a observation after $t = 18000$ as the breakpoint beginning the seizure.

GREASY appears in the figure as non stationary, but it could be segmented into approximately stationary blocks. Note that in the behavior of the time series we can identify blocks corresponding to the sounds G, R, EA, S, and Y (Ombao et al. (2002)). AutoSLEX and AutoPARM found a very high number of breakpoints. The performance of ICM seems to be better than the other methods: it found only 7 breakpoints, most of them limiting intervals corresponding to the sounds compounding the word GREASY.

The number of changes found in EEGT3 and GREASY by ICM, AutoSLEX and AutoPARM is presented in the Table 8.

Table 8: Number of breakpoints estimated for real datasets by each methodology

	ICM	AutoPARM	AutoSLEX
EEGT3	5	18	28
GREASY	7	47	19

6 Conclusions

We have presented a new segmentation methodology, ICM, based on an iterative cusum test which is designed to search and identify multiple moments of parameters change in time series when the underlying distribution is completely unknown. ICM has the advantage of been an intuitive parametric method which is computationally atractive because it does not need genetic algorithm as is the case of AutoPARM. Monte Carlo simulations show that although AutoPARM method usually performs better than ICM, the obtained proportion of wrong segmentations are still satisfactory. The applications of ICM to real datasets present very good performance and excellent results.

Still the method needs good benefits from other dynamic structure and more research, but we believe that the obtained results are very promising.

References

Brockwell, P. and R. Davis (1991). *Time Series: Theory and Methods*. Springer.

Davis, R., T. Lee, and G. Rodriguez-Yam (2006). Structural Break Estimation for Nonstationary Time Series Models. *JOURNAL-AMERICAN STATISTICAL ASSOCIATION 101*(473), 223.

Donoho, D., S. Mallat, and R. von Sachs (1998). Estimating Covariances of Locally Stationary Processes: Rates of Convergence of Best Basis Methods. *Dept. Statist., Stanford Univ., Stanford, CA 517*.

Huang, H., H. Ombao, and D. Stoffer (2004). Discrimination and Classification of Nonstationary Time Series Using the SLEX Model. *JOURNAL-AMERICAN STATISTICAL ASSOCIATION 99*, 763–774.

Inclán, C. and G. Tiao (1994). Use of cumulative sums of squares for retrospective detection of changes of variance. *Journal of the American Statistical Association 89*(427), 913–923.

Lee, S., J. Ha, O. Na, and S. Na (2003). The cusum test for parameter change in time series models. *Scandinavian Journal of Statistics 30*(4), 781–796.

Nicholls, D., B. Quinn, and D. Nicholls (1982). Random coefficient autoregressive models: an introduction.

Ombao, H., J. Raz, R. von Sachs, and W. Guo (2002). The SLEX Model of a Non-Stationary Random Process. *Annals of the Institute of Statistical Mathematics 54*(1), 171–200.

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

Tong, H. (1990). *Non-linear time series: a dynamical system approach.* Oxford University Press.

Wickerhauser, M. (1994). Adapted Wavelet Analysis from Theory to Software Algorithms. *AK Peters: Massachusetts.*